Context-based RNA-seq mapping

Thomas Bonfert

München 2016

Context-based RNA-seq mapping

Thomas Bonfert

Dissertation
an der Fakultät für Mathematik, Informatik und Statistik
der Ludwig–Maximilians–Universität
München

vorgelegt von
Thomas Bonfert
aus Nürtingen

München, den 28.01.2016

Erstgutachter: Prof. Dr. Caroline C. Friedel
Zweitgutachter: Prof. Dr. Dmitrij Frishman
Tag der mündlichen Prüfung: 22.04.2016

Eidesstattliche Versicherung

(Siehe Promotionsordnung vom 12.07.11, § 8, Abs. 2 Pkt. .5.)

Hiermit erkläre ich an Eidesstatt, dass die Dissertation von mir selbstständig, ohne unerlaubte Beihilfe angefertigt ist.

Bonfert, Thomas

Name, Vorname

München, 28.01.2016

Ort, Datum

Unterschrift Doktorand/in

Formular 3.2

Contents

List of Figures

List of Tables

Summary

In recent years, the sequencing of RNA (RNA-seq) using next generation sequencing (NGS) technology has become a powerful tool for analyzing the transcriptomic state of a cell. Modern NGS platforms allow for performing RNA-seq experiments in a few days, resulting in millions of short sequencing reads. A crucial step in analyzing RNA-seq data generally is determining the transcriptomic origin of the sequencing reads (= read mapping). In principal, read mapping is a sequence alignment problem, in which the short sequencing reads (30 - 500 nucleotides) are aligned to much larger reference sequences such as the human genome (∼3 billion nucleotides).

In this thesis, we present ContextMap, an RNA-seq mapping approach that evaluates the context of the sequencing reads for determining the most likely origin of every read. The context of a sequencing read is defined by all other reads aligned to the same genomic region. The ContextMap project started with a proof of concept study, in which we showed that our approach is able to improve already existing read mapping results provided by other mapping programs. Subsequently, we developed a standalone version of ContextMap. This implementation no longer relied on mapping results of other programs, but determined initial alignments itself using a modification of the Bowtie short read alignment program. However, the original ContextMap implementation had several drawbacks. In particular, it was not able to predict reads spanning over more than two exons and to detect insertions or deletions (indels). Furthermore, ContextMap depended on a modification of a specific Bowtie version. Thus, it could neither benefit of Bowtie updates nor of novel developments (e.g. improved running times) in the area of short read alignment software.

For addressing these problems, we developed ContextMap 2, an extension of the original ContextMap algorithm. The key features of ContextMap 2 are the context-based resolution of ambiguous read alignments and the accurate detection of reads crossing an arbitrary number of exon-exon junctions or containing indels. Furthermore, a plug-in interface is provided that allows for the easy integration of alternative short read alignment programs (e.g. Bowtie 2 or BWA) into the mapping workflow. The performance of ContextMap 2 was evaluated on real-life as well as synthetic data and compared to other state-of-the-art mapping programs. We found that ContextMap 2 had very low rates of misplaced reads and incorrectly predicted junctions or indels. Additionally, recall values were as high as for the top competing methods. Moreover, the runtime of ContextMap 2 was at least two fold lower than for the best competitors.

In addition to the mapping of sequencing reads to a single reference, the ContextMap

approach allows the investigation of several potential read sources (e.g. the human host and infecting pathogens) in parallel. Thus, ContextMap can be applied to mine for infections or contaminations or to map data from meta-transcriptomic studies. Furthermore, we developed methods based on mapping-derived statistics that allow to assess confidence of mappings to identified species and to detect false positive hits. ContextMap was evaluated on three real-life data sets and results were compared to metagenomics tools. Here, we showed that ContextMap can successfully identify the species contained in a sample. Moreover, in contrast to most other metagenomics approaches, ContextMap also provides read mapping results to individual species. As a consequence, read mapping results determined by ContextMap can be used to study the gene expression of all species contained in a sample at the same time. Thus, ContextMap might be applied in clinical studies, in which the influence of infecting agents on host organisms is investigated.

The methods presented in this thesis allow for an accurate and fast mapping of RNA-seq data. As the amount of available sequencing data increases constantly, these methods will likely become an important part of many RNA-seq data analyses and thus contribute valuably to research in the field of transcriptomics.

Zusammenfassung

In den letzten Jahren ist das Sequenzieren von RNA (RNA-Seq) mit Hilfe von Sequenzierungstechnologien der nächsten Generation (kurz als NGS-Technologien bezeichnet) zu einer leistungsfähigen Methode bei der Analyse des transkriptionellen Zustandes einer Zelle geworden. Moderne NGS-Technologien erlauben es, RNA-Seq-Experimente in wenigen Tagen durchzuführen. Hierbei werden die Sequenzen von Millionen von kurzen Fragmenten abgelesen (engl. sequencing reads). Ein entscheidender Schritt bei der Analyse von RNA-Seq-Daten ist im Allgemeinen die transkriptomische Herkunft dieser Reads zu bestimmen (= Read-Mapping). Prinzipiell ist das Read-Mapping ein Sequenzalignment-Problem, bei dem die kurzen Reads (30 - 500 Nukleotide) zu einer viel größeren Referenzsequenz, wie zum Beispiel das menschliche Genom ($\sim$ 3 Milliarden Nukleotide), aligniert werden.

In dieser Arbeit präsentieren wir ContextMap, ein RNA-Seq-Mapping-Ansatz, der den Kontext der Reads evaluiert, um so die wahrscheinlichste Herkunft eines jeden Reads zu bestimmen. Der Kontext eines Reads ist durch alle anderen Reads definiert, die in der gleichen genomischen Region aligniert werden konnten. Das ContextMap-Projekt begann mit einer Machbarkeitsstudie, in der wir gezeigt haben, dass unser Ansatz bereits existierende Read-Mapping-Ergebnisse von anderen Programmen verbessern kann. Anschließend haben wir eine eigenständige Version von ContextMap entwickelt. Diese Implementierung war nicht mehr von Mapping-Ergebnissen anderer Programme abhängig, sondern konnte initiale Alignments mit Hilfe einer Modifikation des Bowtie Read-Alignment-Programms berechnen. Dennoch hatte der ursprüngliche ContextMap-Algorithmus mehrere Nachteile. Im Wesentlichen konnte ContextMap keine Reads mappen, die über mehr als zwei Exons spannen oder Insertionen oder Deletionen (Indels) beinhalten. Zudem war ContextMap von einer spezifischen Bowtie-Version abhängig und konnte deshalb weder von Bowtie-Updates noch von neuen Entwicklungen (z.B.: bessere Laufzeiten) im Bereich von Read-Alignment-Software profitieren.

Um diese Probleme anzugehen, haben wir ContextMap 2 entwickelt, eine Erweiterung des ursprünglichen ContextMap-Algorithmus. Die Hauptmerkmale von ContextMap 2 sind das kontextbasierte Auflösen von mehrdeutigen Read-Alignments und eine akkurate Detektion von Reads, die eine beliebige Zahl von Exons überspannen oder Indels beinhalten. Darüberhinaus wird ein Plugin-Interface bereitgestellt, welches die Einbindung alternativer Read-Alignment-Programme (z.B.: Bowtie 2 oder BWA) in den Mapping-Workflow erlaubt. Die Performance von ContextMap 2 wurde auf echten und synthetischen Daten evaluiert und mit anderen state-of-the-art Mapping-Programmen verglichen. Wir haben

festgestellt, dass ContextMap 2 sehr geringe Raten an fehlplatzierten Reads und falsch vorhergesagten Junctions oder Indels aufweist. Zudem waren Recall-Werte genau so hoch und Laufzeiten mindestens zweifach geringer als bei den besten konkurrierenden Methoden.

Neben dem Mapping von Reads zu einer einzigen Referenz, ist es mit dem ContextMap-Ansatz möglich, mehrere potentielle Read-Herkünfte (z.B.: der menschliche Wirt und infizierende Krankheitserreger) gleichzeitig zu untersuchen. Deshalb kann ContextMap zur Suche nach Infektionen oder Kontaminationen oder zum Mappen von meta-transkriptomischen Daten verwendet werden. Zudem haben wir, basierend auf Read-Mapping Statistiken, Methoden entwickelt, die es ermöglichen die Konfidenz von Mapping-Ergebnissen zu identifizierten Spezies zu bewerten sowie falsch positive Hits zu detektieren. ContextMap wurde auf drei realen Datensätzen ausgewertet und die Ergebnisse mit metagenomischen Tools verglichen. Bei diesen Auswertungen konnten wir zeigen, dass ContextMap erfolgreich alle Spezies die in einem Sample enthalten sind identifiziert. Darüberhinaus liefert ContextMap Read-Mapping-Ergebnisse zu einzelnen Spezies, was für die meisten anderen metagenomischen Ansätze nicht gilt. Daraus folgt, dass die mit ContextMap berechneten Read-Mapping-Ergebnisse dazu verwendet werden können um zeitgleich Genexpressionswerte von allen Spezies eines Samples zu untersuchen. Deshalb könnte ContextMap in klinischen Studien Anwendung finden, bei denen Einflüsse infizierender Erreger auf den Wirtsorganismus erforscht werden.

Die in dieser Arbeit präsentierten Methoden ermöglichen es, RNA-Seq-Daten akkurat und schnell zu mappen. Durch den stetigen Zuwachs an verfügbaren RNA-Seq-Daten werden diese Methoden wahrscheinlich zu einem wichtigen Teil vieler RNA-Seq-Datenanalysen und so einen wertvollen Beitrag zur Forschung im Bereich der Transkriptomik leisten.

Chapter 1

Introduction

1.1 Overview and motivation

The *deoxyribonucleic acid* (DNA) is the molecule that carries the genetic information of all living organisms. The discovery of the DNA by Friedrich Miescher in 1869 (reviewed in Dahm [2008]) and the identification of the structure of DNA molecules by Watson and Crick (Watson and Crick [1953]) in 1953 were groundbreaking novel insights in the field of genetics. Based on this knowledge, Sanger and colleagues (Sanger et al. [1977]) as well as Maxam and Gilbert (Maxam and Gilbert [1977]) were able to develop methods for determining the exact order of nucleotides (= *sequencing*) of DNA molecules in 1977.

With the ability to sequence DNA, the dream of sequencing the entire human genome (∼3 billion nucleotides) emerged. However, for this purpose further technical improvements of sequencing procedures were required. Therefore, it took until 1990 before researchers of the publicly funded *Human Genome Project* (HGP) were able to start with the sequencing of the genome. Eight years later, the company Celera announced that it would also attempt to sequence the human genome. Eventually, both groups published a first rough draft of the genome in 2001 (Lander et al. [2001]; Venter et al. [2001]). However, it was the HGP consortium who continued to refine their draft and finally published a complete genome sequence in 2003, with about 20.500 identified genes (International Human Genome Sequencing Consortium [2004]). The HGP consumed around 3 billion US dollars, which was an immense cost factor, in particular when compared to the 300 million US dollars invested by Celera.

Encouraged by Celera and the HGP, other companies focused on commercializing sequencing by developing methods that were cheaper, faster and had higher throughput. In 2005, the company 454 Life Science introduced the first high-throughput method for sequencing DNA (Margulies et al. [2005]), which marked the beginning of the era of *next-generation-sequencing* (NGS) technologies (reviewed in van Dijk et al. [2014]). Competing companies such as Illumina and Applied Biosystems commercialized their own sequencing platforms one and two years later, respectively. Modern NGS platforms allow for sequencing whole genomes in a few days at a much lower cost in comparison to the HGP or Celera

(see Metzker [2010] and Liu et al. [2012] for a comparison of technologies). Due to technical limitations, all NGS methods have in common that only short fragments of huge DNA molecules can be sequenced. Therefore, the DNA is fragmented prior to sequencing and a typical NGS experiment results in millions of short so-called sequencing reads.

The application of NGS is not limited to the sequencing of genomes. For instance, methods like ChIP-seq (reviewed in Park [2009]) or PAR-Clip (Hafner et al. [2010]) can be applied for identifying and sequencing regions on the DNA that are bound by a particular protein. Another application of NGS technologies is the sequencing of RNA (*RNA-seq*) (reviewed in Ozsolak and Milos [2011]), which allows for the quantification of expression levels of almost all expressed genes or transcripts in a cell. In theory, it offers various advantages over hybridization based methods such as microarrays (reviewed in Wang et al. [2009]; Hurd and Nelson [2009]). In contrast to microarrays, RNA-seq does not require prior knowledge about the genome or genomic features of the species to which it is applied. Furthermore, it has a higher dynamic range than hybridization based methods, which means that RNA-seq is more suitable for measuring the expression of low and high abundant transcripts simultaneously. Finally, RNA-seq has very low background signal, in particular when compared to microarrays. Nevertheless, comparisons of the two techniques showed that both methods are useful tools for studying transcriptomes (Malone and Oliver [2011]; Sirbu et al. [2012]; Yu et al. [2015]). RNA-seq was already successfully applied to study alternative splicing (e.g. Sultan et al. [2008]; Tang et al. [2009]; Richard et al. [2010]), to detect gene fusions (e.g. Maher et al. [2009]; Berger et al. [2010]), antisense transcription (e.g. Yassour et al. [2010]; Lasa et al. [2011]; Bao et al. [2015]) and non-coding RNAs (e.g. Pauli et al. [2012]; Luo et al. [2013]; Jha et al. [2015]) and more.

The rapid development of NGS technologies and their broad field of applications comes along with two major bioinformatic challenges. First, bioinformaticians have to deal with the immense amount of data that is generated due to the dramatically decreasing prices for sequencing (Hayden [2014]) and large scale projects such as the 1000 genome project (Abecasis et al. [2010]) or ENCODE (ENCODE Project Consortium [2012]). Currently, most of the sequencing data is stored and organized in databases such as the Short Read Archive (Shumway et al. [2010]) or the European Nucleotide Archive (Leinonen et al. [2011]). However, there is a trend towards using cloud computing combined with discarding large parts of the raw data after analysis (reviewed in Stein [2010]; Stephens et al. [2015]). Second, biologists require algorithms and software solutions that are specifically designed for analyzing data originating from certain experimental setups. These programs must be fast and produce results that can be used for performing meaningful analyses.

When analyzing RNA-seq data, researchers basically have two options to start such an analysis. The first option is to assemble the sequencing reads to complete transcripts, for which a reference genome is not necessarily needed (reviewed in Martin and Wang [2011]). The second option is to align the sequencing reads to a given reference sequence (e.g. the genome) in order to determine the transcriptomic origin of every sequencing read (= *read mapping*) (reviewed in Trapnell and Salzberg [2009]; Garber et al. [2011]). The mapping of sequencing reads belongs to the category of sequence alignment problems and thus to a classical challenge for bioinformatics.

In the following, we provide the biological background knowledge that is relevant for understanding the aims of RNA-seq experiments. Furthermore, we give a short overview about procedures for sequencing RNA and algorithms for determining sequence alignments. An introduction to RNA-seq read mapping approaches is provided in chapter 2.

1.2 Biological background

1.2.1 Overview

The DNA stores the information needed for producing proteins, the basic elements for building a cell or a tissue of an organism. The two strands of the polymeric DNA molecule consist of only four different monomers, namely the *nucleotides*. The nucleotides are composed of a sugar molecule (*deoxyribose*), a phosphate group and an organic *base*, which is either *adenine* (A), *cytosine* (C), *guanine* (G) or *thymine* (T). For convenience, the four different nucleotides are often abbreviated with the single letter of the respective bases A, C, G or T. Each of the two DNA strands consists of a chain of nucleotides, which are linked via phosphodiester bonds. The orientation of a DNA strand is defined by a nucleotide with a free phosphate group at one end (*5' end*) and a nucleotide with a free hydroxyl group of the sugar molecule at the other end (*3' end*) of the strand. Furthermore, the nucleotides of the two different strands form hydrogen bonds between each other. Here, A bonds with T, and G with C, respectively. These pairs are called *base-pairs* (bps), which have great influence on the stability of the *double helix structure* of the DNA (see the upper box of Figure 1.1 for an illustration).

The information stored in the DNA is organized in *genes*, which are units of the DNA that encode for proteins or functional *ribonucleic acid* (RNA) molecules. The chemical structure of an RNA molecule is very similar to the structure of the DNA. RNA is a single stranded molecule that consists of a chain of four different nucleotides. The sugar molecule of these nucleotides is a *ribose* (instead of deoxyribose) and the four different bases are A, C, G and *uracil* (U) (instead of thymine). When the central dogma of molecular biology was formulated by Francis Crick in 1958 (Crick [1958]), it was well known that during the process of protein synthesis the information contained in a gene is initially transferred to an RNA molecule (see *Gene expression* (1.2.2)). At this time, it was assumed that RNA functions in two ways only, namely as messenger between DNA and protein and as molecules that are involved in the synthesis of proteins (e.g. rRNAs and tRNAs) (Hoagland et al. [1958]; Brenner et al. [1961]).

Since then, these findings were further extended and we distinguish today between non-coding and protein-coding genes. The former genes do not encode for proteins, but for non-coding RNAs (ncRNAs). Researchers proved the existence of thousands of ncRNAs encoded in the human genome (reviewed in Mattick and Makunin [2006]; Morris and Mattick [2014]). Recent studies suggest that ncRNAs are involved in the regulation of gene expression (e.g. miRNAs, long ncRNAs) (reviewed in Fatica and Bozzoni [2014]; Jonas and Izaurralde [2015]).

In general, the genes of eukaryotes are divided into exons and introns (see Figure 1.1). However, only the exons are the sequence parts that contain the information needed for synthesizing the final gene product, while the introns are removed during the so-called splicing process (see section 1.2.2).

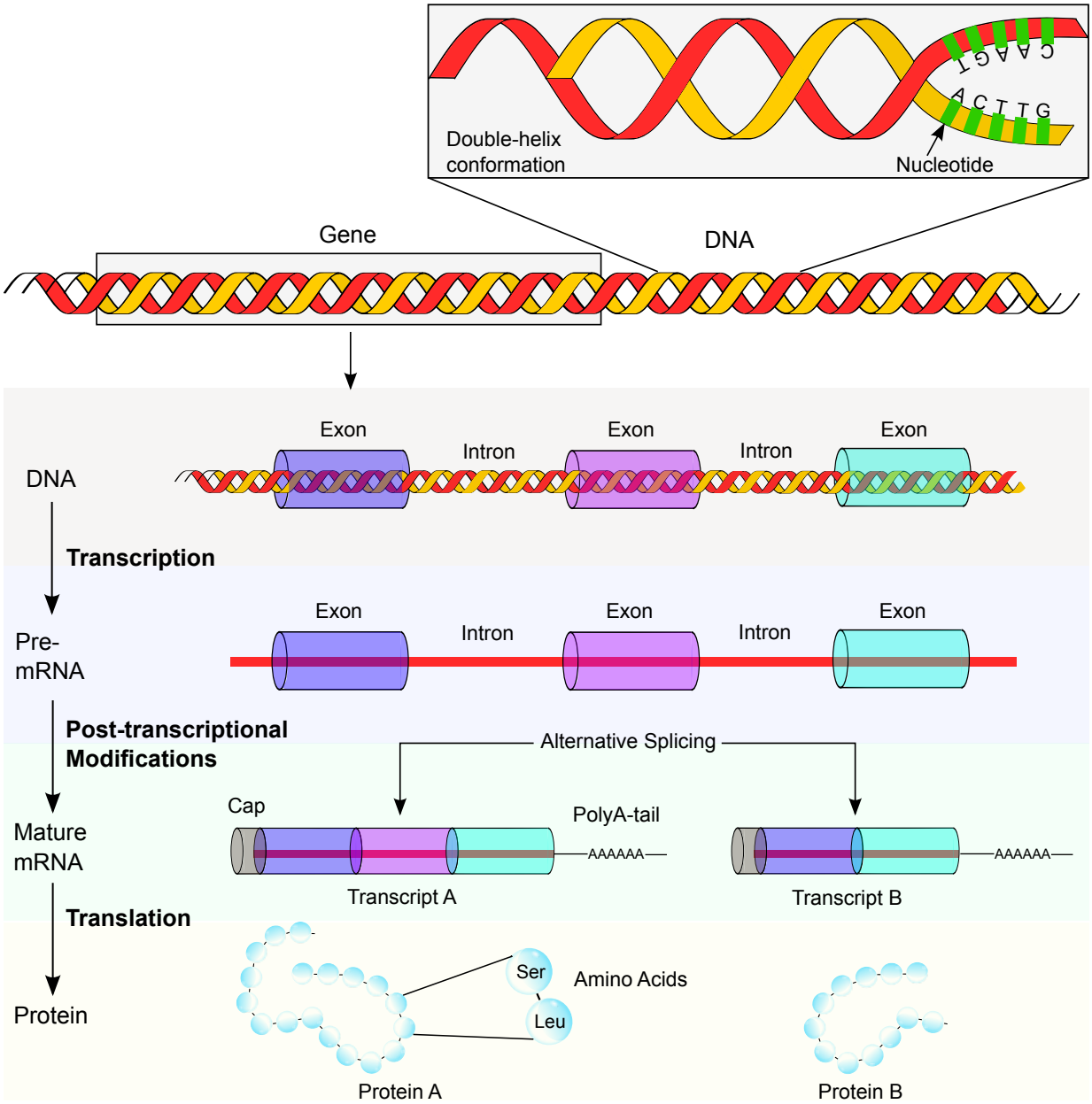

Figure 1.1: Overview of the gene structure and expression of a protein-coding gene. The upper box shows the structure of the DNA molecule, which is composed of two strands forming a double-helix. The coding parts of the depicted gene (i.e. the exons) are separated by introns. During transcription, one of the two DNA strands is copied to a single stranded pre-mRNA molecule. The pre-mRNA is subject to post-transcriptional modifications and translated to proteins.

1.2.2 Gene expression

For clarification purposes, we define the expression of a gene as the process of synthesizing a functional gene product from the information contained in the respective gene. To be more specific, the expression of protein-coding genes results in proteins and the expression of non-coding genes in ncRNAs, respectively. The gene expression processes of protein-coding and non-coding genes have the first two steps in common, while a last third step is unique to the protein-coding genes (see Figure 1.1).

First, *transcription* generates a copy of one of the DNA strands, the so-called coding strand. For this purpose, the enzyme RNA polymerase traverses along the template strand, which is the counterpart of the coding strand, and synthesizes a single stranded RNA copy of the coding strand. The resulting RNA molecule is denoted as pre-messenger RNA (pre-mRNA) for protein-coding genes.

In the second step, the RNA undergoes post-transcriptional modifications. At the 3' end of the RNA a poly(A)-tail is added, which consists of multiple adenine nucleotides. At the 5' end a modified guanine nucleotide is added, which is called the cap of an RNA. Both, the tail and the cap are involved in the protection of the RNA from degradation and are important for the export of the RNA from the nucleus. The maturation of the mRNA into a so-called transcript is completed by the splicing process, which removes the introns and joins the exons. Splicing can result in different exon compositions and thus in different transcripts for the same gene. This process is known as *alternative splicing*. At the end of this step, gene expression of a non-coding gene is completed.

For a protein-coding gene, ribosomes will eventually translate the nucleotides of each mRNA into chains of amino acids. Each amino-acid chain folds into a three dimensional structure, which results in functional proteins, the final product of a protein coding gene.

1.3 Sequencing of RNA

1.3.1 Overview

RNA-seq is a method for determining the exact order of the nucleotides in RNA molecules using NGS technologies. In general, NGS methods are designed for sequencing DNA molecules. Therefore, RNA molecules are reverse transcribed to DNA molecules prior to sequencing (described below (1.3.2)). A common sequencing approach of available NGS platforms is the *sequencing-by-synthesis* (SBS) method, in which the de-novo synthesis of double stranded DNA molecules is monitored with imaging technology (reviewed in Fuller et al. [2009]). For this purpose, fragments of the original DNA molecules are immobilized on a solid surface (e.g. a glass slide). Subsequently, the DNA fragments are denaturated (= conversion to single-stranded molecules) and an enzymatically catalyzed synthesis of double-stranded DNA is started with the addition of the four nucleotides A,C,G and T. Modern optical devices are used to detect the incorporation of single nucleotides to each fragment and a new cycle of nucleotide addition is performed. Eventually, after several

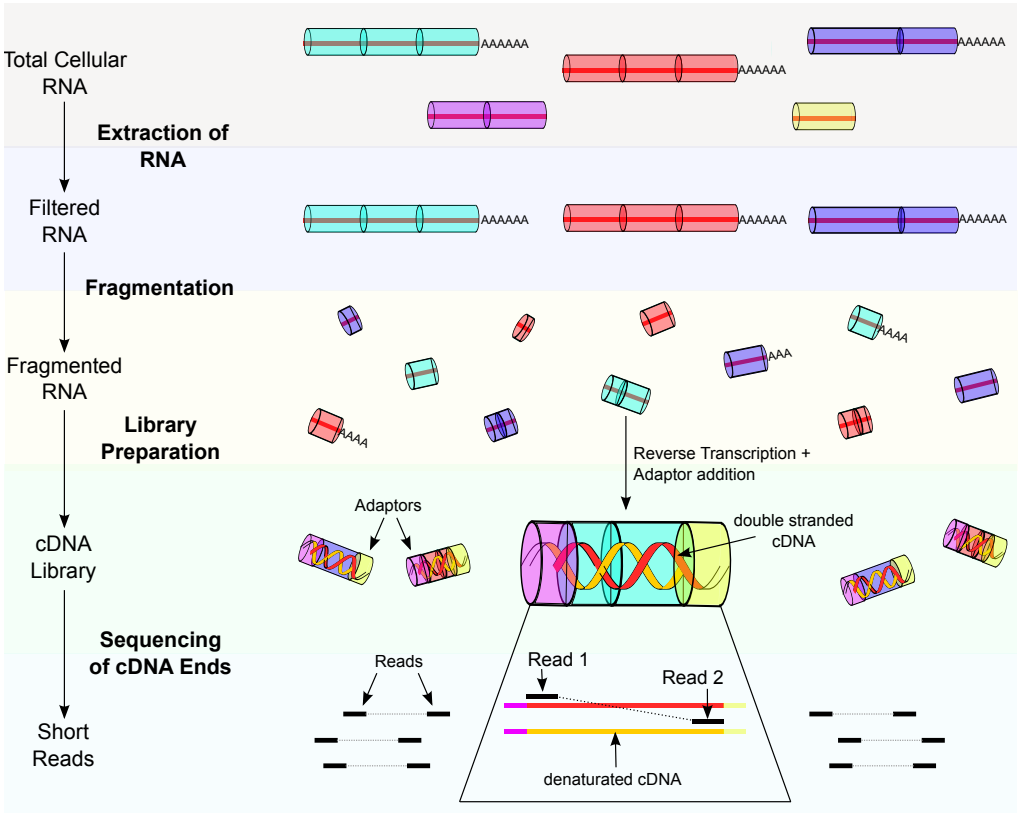

Figure 1.2: Overview of a general RNA-seq experimental workflow. In an intial step, the complete set of RNA is filtered for RNAs that are of interest for the experimental setup. Subsequently, the remaining RNA molecules are fragmented into smaller pieces. These fragments are reverse transcribed into cDNA. Usually, adaptor sequences are ligated to both ends of the cDNAs. The adaptors are used for attaching the cDNAs on a solid surface and for amplifying the molecules using PCR. The last step is the sequencing of the cDNA ends. In the depicted example, both ends of the denaturated cDNA fragments are sequenced (i.e. paired-end sequencing), which means that two sequencing reads are generated per cDNA.

cycles of nucleotide addition, the final nucleotide sequence is determined by evaluating the imaging data.

NGS platforms performing SBS differ in some points, such as the surface the DNA molecules are attached to, or the way how the nucleotides are made visible for the optical devices (e.g. fluorescently labeled nucleotides) and more. However, even though there are technical differences among individual sequencing technologies, all platforms have a common strategy for preparing the sequencing libraries, which will be explained below.

1.3.2 A general RNA-seq experimental workflow

In the following, we describe a general and simplified RNA-seq workflow. The various available NGS platforms have different protocols for sequencing RNA. However, almost all methods have the same basic preparation steps in common (van Dijk et al. [2014]) (see Figure 1.2 for an overview).

The workflow starts by extracting the type of RNA that is of interest for the underlying experimental setup. In the given example, only RNAs with polyA-tails are extracted. Subsequently, the extracted RNAs are fragmented at random positions into smaller segments by physical, enzymatic or chemical shearing. The fragmentation step is necessary because current NGS platforms are limited to sequence up to a few hundred nucleotides only, and thus are not able to sequence mRNAs in one piece. Following the fragmentation, a filtering step is performed in which only fragments of suitable sizes (between 50-500 bps) are selected.

The next step is the preparation of the library. For this purpose, the single stranded RNA molecules are reverse transcribed into double stranded complementary DNA (cDNA) molecules. Subsequently, adaptor sequences are added to the ends of the cDNA molecules. In general, these adapters consist of parts required for amplifying the molecules as well as for attaching them onto a solid surface. The latter is necessary for performing the SBS method, which allows for real-time monitoring of DNA synthesis. After immobilization of the cDNAs, the molecules are amplified with the polymerase chain reaction (PCR) in order to generate sufficient quantities needed for capturing the synthesis with optical devices.

Finally, the cDNAs are denaturated and the de-novo synthesis of the double stranded DNA molecules is started and monitored as described in the overview (section 1.3.1). Depending on the applied sequencing protocol, the synthesis can be performed from one end only (i.e. single-end sequencing) or from both ends (i.e. paired-end sequencing). After an evaluation of the images which are taken during the synthesis, the final outcome of the experiment are millions of short sequencing reads (30-400 bps). Note that the sequencing reads are generally shorter than the sequenced fragments and cover only the end of the fragments. Thus, sequencing reads originating from fragments of a single copy of the original mRNA do not necessarily cover the complete mRNA sequence. However, it is assumed that the respective mRNA is available in many copies and that the randomly chosen shearing positions during the fragmentation step result in equally distributed start positions of the fragments along the mRNA. Therefore, in theory the sequencing reads should cover the whole mRNA given enough sequencing depth.

1.4 Mapping of RNA-seq data

1.4.1 Overview

The mapping of RNA-seq data describes the process of assigning sequencing reads to positions on a given reference sequence (e.g. a genome). Ideally, every read is assigned to the position from where it was originally sampled during the experiment. The mapping of sequencing reads basically is an alignment problem, in which similarities between the sequencing reads and the given reference sequence are determined. Several methods exist for determining sequence alignments, including approaches that were explicitly designed for aligning millions of short sequencing reads (see section 1.4.2).

In Figure 1.3 we show four principle ways how RNA-seq reads can be aligned to a given

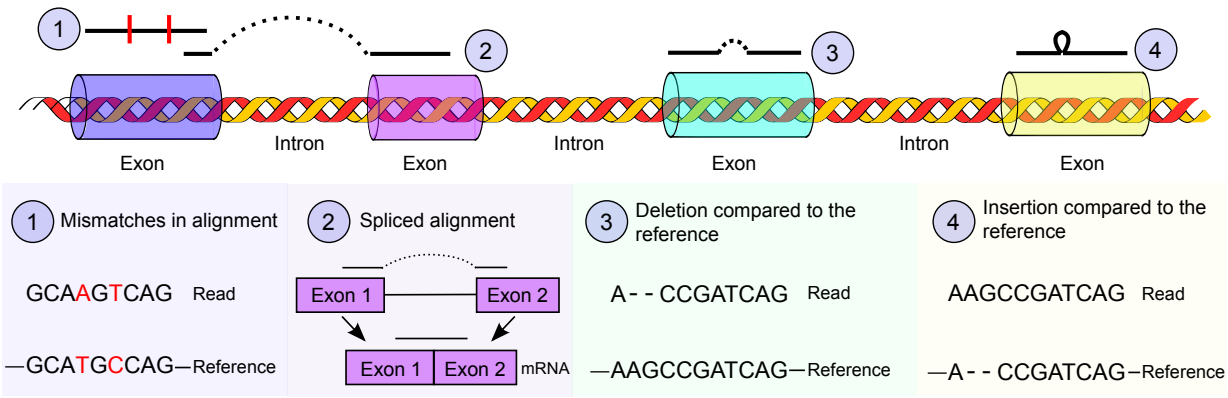

Figure 1.3: Four different ways of read alignments to the genome. Small gaps are represented by dashes, larger gaps by dotted lines. Nucleotides drawn in black are matches between read and reference, whereas nucleotides drawn in red indicate for mismatches in the alignment. Further explanations are given in the text.

reference genome. For the sake of simplicity, we depicted only a single event per alignment. However, in real-life data the different types of alignments can appear in arbitrary combinations for sequencing reads. Furthermore, a continuous alignment of an exact matching sequencing read is not depicted, as this is the simplest event that can occur and is not as challenging to determine as the other events are.

The first type of alignment occurs for sequencing reads that have mismatches against the reference, which can arise from sequencing errors or single nucleotide polymorphisms (SNPs). Sequencing errors are technical issues (see Dohm et al. [2008]), whereas SNPs arise from biological variance in DNA or RNA sequences. There are about 3 million SNPs per human individual, some of which are suggested to be responsible for phenotypic differences, drug responses and the susceptibility for disease (Ku et al. [2010]; Yang et al. [2010]; Buch et al. [2007]).

The second type of alignment arises when the sequencing read overlaps two or more exons of a spliced mRNA. Such a sequencing read cannot be aligned continuously to the genome as the connected exons of the mRNA are separated by introns on the genome. Therefore, a so-called spliced alignment has to be determined, which contains gaps for 'skipping' the introns. The sequencing read depicted in Figure 1.3 overlaps two exons, thus the respective spliced alignment contains a single gap only. However, with increasing read lengths, the probability of observing sequencing reads that overlap more than two exons increases.

Finally, the third and the fourth type of alignments occur for reads that either have an insertion or a deletion (indel) of nucleotides compared to the reference. Indels and SNPs are the two most frequent events that lead to genetic variation in humans (reviewed in Mullaney et al. [2010]). Recent studies showed that indels can also be responsible for disease or alterations in human traits (reviewed in Orr and Zoghbi [2007]).

1.4.2 Algorithms for determining short read alignments

The major requirements of algorithms for determining alignments of sequencing reads to a given reference sequence are the following. First, an alignment of a single sequencing read must be determined very quickly because NGS experiments produce millions of short reads. Second, the algorithms have to be tolerant against mismatches and indels, respectively. Otherwise, the sequencing reads that differ from the reference sequence due to sequencing errors or genetic variation cannot be aligned at all. Finally, if sequencing reads from an RNA-seq experiment are aligned, the algorithm must be able to handle the large gaps contained in spliced alignments.

In principal, alignments of sequencing reads can be determined with algorithms such as the Needleman-Wunsch (Needleman and Wunsch [1970]) or the Smith-Waterman (Smith and Waterman [1981]) algorithm. Both use dynamic programming and guarantee to find optimal alignments in terms of the alignment score. An alignment score is obtained by penalizing mismatches and gaps and awarding matches. However, determining optimal alignments with these algorithms is relatively slow and takes quadratic time with respect to the length of the input sequences.

To address this problem, heuristics were developed, such as FASTA (Lipman and Pearson [1985]), BLAST (Altschul et al. [1990]) or BLAT (Kent [2002]), to increase the alignment speed at the cost of sensitivity. BLAST was published in 1990 and was one of the most cited articles in bioinformatics in 2014, with more than 35,000 citations (Van Noorden et al. [2014]). Like FASTA and BLAT, BLAST is able to perform gapped alignments for protein as well as nucleotide sequences. However, due to the fast development in sequencing technology during the last decade, faster alignment algorithms for determining alignments of millions of short sequencing reads were developed. Currently, there are two different fundamental techniques that are widely used for determining such alignments (Flicek and Birney [2009]; Li and Homer [2010]).

The first group of methods consists of hash-based alignment programs. Here, either the reference is stored in a hash-table (e.g. BFAST (Homer et al. [2009]), MOSAIK (Lee et al. [2014]) and SOAP (Li et al. [2008b])) or the read sequences (e.g. ELAND, MAC, ZOOM (Lin et al. [2008])). Hashing the reference may have a large, but fixed memory footprint, depending on the size of the reference. But the reference has to be hashed only once and a query can be performed in constant time. If the read sequences are hashed, the memory usage may be smaller, but aligning a small number of read sequences can take a very long time because the whole reference has to be searched.

The second group of methods is based on suffix arrays or related data structures. The most prominent alignment programs of this category (e.g. Bowtie (Langmead et al. [2009]), Bowtie 2 (Langmead and Salzberg [2012]), BWA (Li and Durbin [2009])) rely on the *Burrows Wheeler Transform* (BWT) (Burrows and Wheeler [1994]) combined with an *Full-text index in Minute space* (FM-index) (Ferragina and Manzini [2000]). The BWT can be easily computed from a suffix array and thus is directly related to this data structure. In brief, the BWT is a reversible permutation of the input reference text, such that sequences of identical characters are generated. The FM-index consists of a compression of the

permutated text and of additional data structures that allow recovering and searching the original text. The memory footprint of the index and the time required for querying the index are both of sublinear complexity with respect to the size of the input reference. Due to the small memory footprint, tools based on this technique prefer to index the large reference instead of the sequencing reads. In general, the index of the reference is queried with small regions of each sequencing read (= *seeds*) only. Subsequently, seed hits are extended to completing alignments. Here, different techniques have been developed to increase sensitivity by allowing for inexact matches.

Both, the hash-based and suffix array-based techniques can be applied for determining continuous alignments of sequencing reads to a given reference sequence. However, the techniques are not capable of determining alignments with large gaps, which are contained in spliced alignments of RNA-seq reads. Therefore, the algorithms for aligning RNA-seq data most often apply sophisticated strategies for determining spliced alignments from continuous alignments of only parts of the read. A variety of such strategies will be introduced in the next chapter.

1.5 Outline of the thesis

The main focus of this thesis is the development of ContextMap, a novel approach for mapping RNA-seq reads. The main difference to existing approaches is that ContextMap determines the most likely origin of a sequencing read by evaluating the read context. The context of a sequencing read is defined by all other reads aligned to the same stretch on the genome.

In chapter 2, we provide an introduction to state-of-the-art mapping approaches, of which many have been previously developed. Furthermore, a transcriptome-based mapping workflow is introduced, which we developed before ContextMap. The application of this workflow is demonstrated by presenting an analysis of a time-course RNA-seq experiment (Windhager et al. [2012]). This analysis is based on read mapping results determined with our workflow. At the end of chapter 2, common drawbacks of our and other mapping approaches are discussed.

In chapter 3, we present ContextMap 2 (Bonfert et al. [2012, 2015]), a context-based RNA-seq mapping approach that was developed for addressing the problems of existing approaches. In the first part of chapter 3, the ContextMap 2 algorithm is described in detail. Subsequently, an evaluation of the method is performed on synthetic and real-life data. A comparison to other RNA-seq mapping programs shows that ContextMap 2 is a fast and accurate read mapping software.

In addition to the mapping of reads to a single species, ContextMap is also suitable for mapping reads of any species with a sequenced genome in parallel. In chapter 4, we show that this feature allows ContextMap to identify infections or contamination in RNA-seq data and to map reads from meta-transcriptomic studies (Bonfert et al. [2013]). Furthermore, methods based on mapping-derived statistics for assessing confidence of identified species and detecting false positive hits are introduced.

In the final chapter 5, we summarize the presented work and provide an outlook to future developments of the ContextMap project. This includes a short introduction of our latest ContextMap 2 extension: a novel method for the prediction of poly(A) cleavage sites by mapping poly(A)-tail RNA-seq reads (manuscript in preparation).

Chapter 2

Introduction to RNA-seq mapping approaches

Motivation: This chapter starts with a short introduction to state-of-the-art RNA-seq read mapping approaches. However, the presented overview is not exhaustive, as there are currently at least 15 RNA-seq mapping programs available. Therefore, it would be beyond the scope of this thesis to introduce all of them. For getting a complete overview, we suggest to read the articles by Fonseca et al. [2012] and Alamancos et al. [2014].

In the second part of the chapter we present a transcriptome-based mapping workflow we developed before ContextMap. This workflow aligns sequencing reads sequentially to different reference sequences, such as a given transcriptome and a genome. We demonstrate the application of the workflow by presenting parts of our recently published analysis of a time-course RNA-seq experiment (Windhager et al. [2012]).

Finally, the chapter closes with a description of drawbacks that our workflow and other mapping approaches have in common. These problems predominantly occur for sequencing reads that have more than one possible alignment to a given reference sequence. In such a scenario it is important that the underlying read mapping approach investigates all possible alignments of such a read and implements a strategy for deciding which of the alignments is the correct one.

Publication: The study on the time-course RNA-seq experiment in which we applied our workflow for mapping the sequencing data was published in Genome Research in 2012 (Windhager et al. [2012]). Here, we only included parts of the original manuscript to demonstrate the usability of our mapping workflow.

Author contributions: In the study by Windhager et al. (Windhager et al. [2012]), I performed the read mapping, generated count data and performed the quantification of gene, exon and intron expression levels. Lukas Windhager and Caroline C. Friedel (CCF) used these results for performing further downstream analysis of the data. Lars Dölken (LD) and co-workers did the laboratory work. CCF and LD wrote the article and all co-authors helped in revising the manuscript.

2.1 Existing mapping approaches

2.1.1 TopHat

TopHat (Trapnell et al. [2009]) was one of the first RNA-seq mapping approaches that was developed for the de novo discovery of canonical splice junctions. For this purpose, TopHat operates in three phases (see Figure 2.1 for an overview).

In the first phase, TopHat aligns the reads to a given reference genome using Bowtie (Langmead et al. [2009]). Here, TopHat uses all reads with at most n alignments ($n = 10$ per default) to the genome. Sequencing reads with more alignments are discarded, whereas reads without any alignment are collected for a later processing step.

The second phase assembles contigs from the mapped reads using the assembly tool Maq (Li et al. [2008a]). The resulting contigs are denoted as islands, which represent putative exons. In general, the sequence of an island consists of the consensus sequence defined by the reads covering the island. However, if the island is poorly covered, the reference sequence will be used to represent the island.

In the third phase, TopHat enumerates all canonical donor (GT) and acceptor (AG) splice sites within the island sequences. Subsequently, all pairs of canonical splice sites (GT-AG) between neighboring, but not necessarily adjacent, islands are determined. In addition, TopHat will generate pairs inside an island if the island is highly covered by reads. Finally, unassigned reads are aligned to sequences around the generated splice site pairs. Splice sites and associated spliced alignments will be reported if the coverage of the respective site and the corresponding exons do not differ too much.

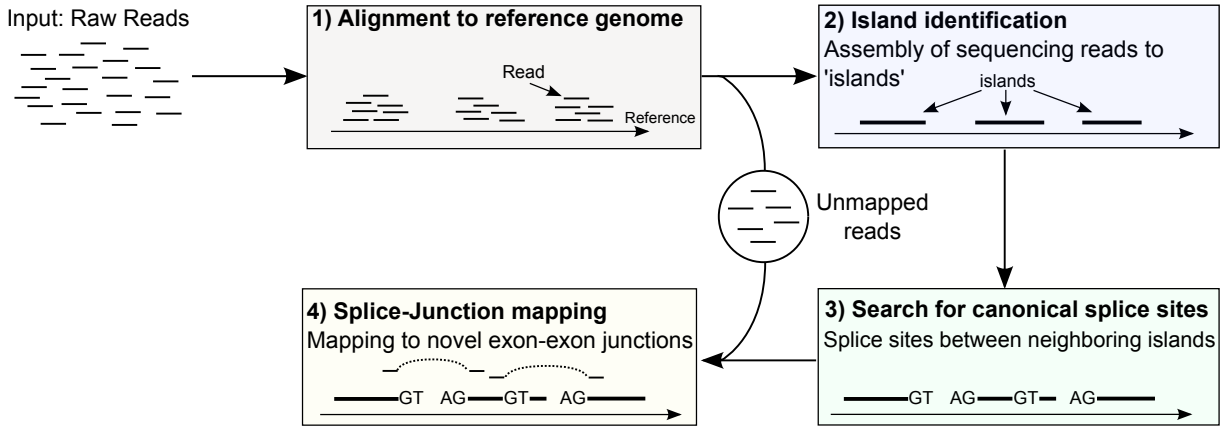

Figure 2.1: The mapping workflow of TopHat. TopHat starts by aligning the sequencing reads to a given reference genome using the Bowtie alignment program. Subsequently, aligned reads are assembled to contigs (denoted as islands) by applying the program Maq. Canonical splice sites between neighboring islands are annotated and, finally, spliced alignments are determined for unaligned reads.

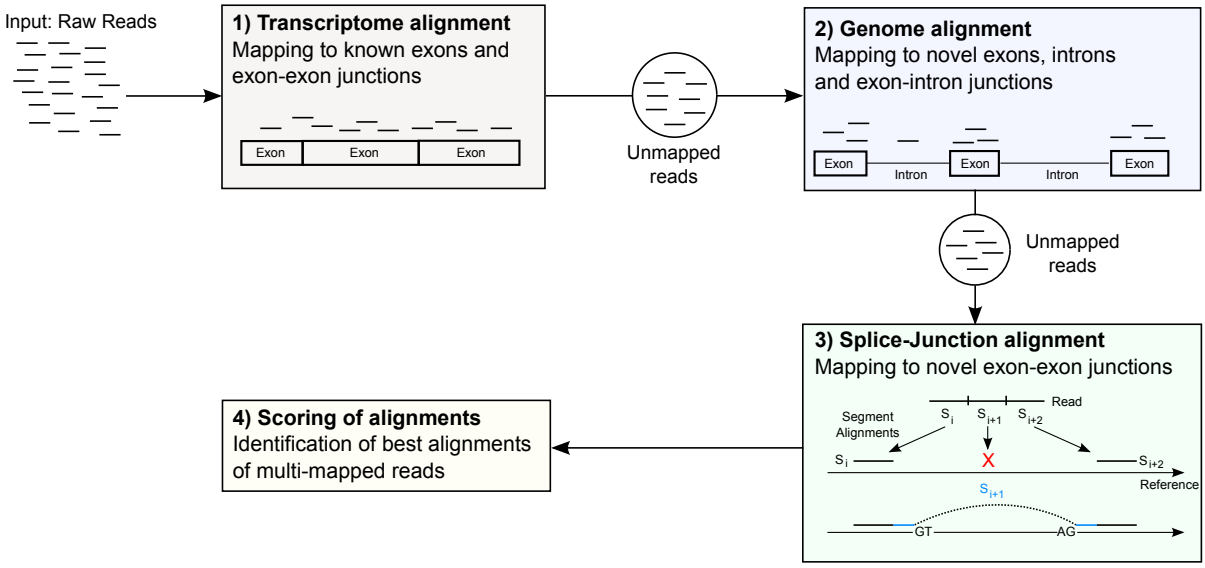

Figure 2.2: The mapping workflow of TopHat2. Sequencing reads are initially mapped to a transcriptome. Subsequently, only the unmapped reads are aligned to the reference genome. For determining spliced alignments, small segments of unmapped reads are aligned to the genome (details provided in the text). Finally, alignment scores of multi-mapped reads are calculated and the best scoring alignments are retained.

2.1.2 TopHat 2

TopHat2 (Kim et al. [2013]) is a complete re-design of the original TopHat algorithm and operates in four different phases (see Figure 2.2). In the first phase, reads are aligned to a transcriptome (if provided) using Bowtie 2 (Langmead and Salzberg [2012]). Reads with alignments that have an edit distance below a user-defined threshold are considered as mapped and will not be re-aligned in any of the following steps. In the second phase, unmapped reads are aligned to the reference genome. Again, all reads with alignments to the genome that have an edit distance below the mentioned threshold are considered as mapped.

In the third phase, TopHat2 detects reads that span over two or more exons. For this purpose, unmapped reads are split into smaller segments and then aligned to the genome. TopHat2 searches for segment alignments that are left and right neighbors of an unaligned segment of the same read. Subsequently, TopHat2 extracts parts of the reference sequence downstream of the left neighbor alignment and upstream of the right neighbor alignment. These parts are concatenated and unaligned segments are aligned against them. Finally, all aligned segments of a read are gathered to determine completing spliced alignments. At the end of this step, TopHat2 checks if there are alignments overlapping annotated exon-intron junctions by only a few nucleotides and re-aligns these reads to already detected splice junctions.

In the final phase, TopHat2 identifies the most likely mapping locations of reads with multiple alignments. For this purpose, alignment scores are calculated based on statistical information such as the number of supporting reads for a relevant splice junction.

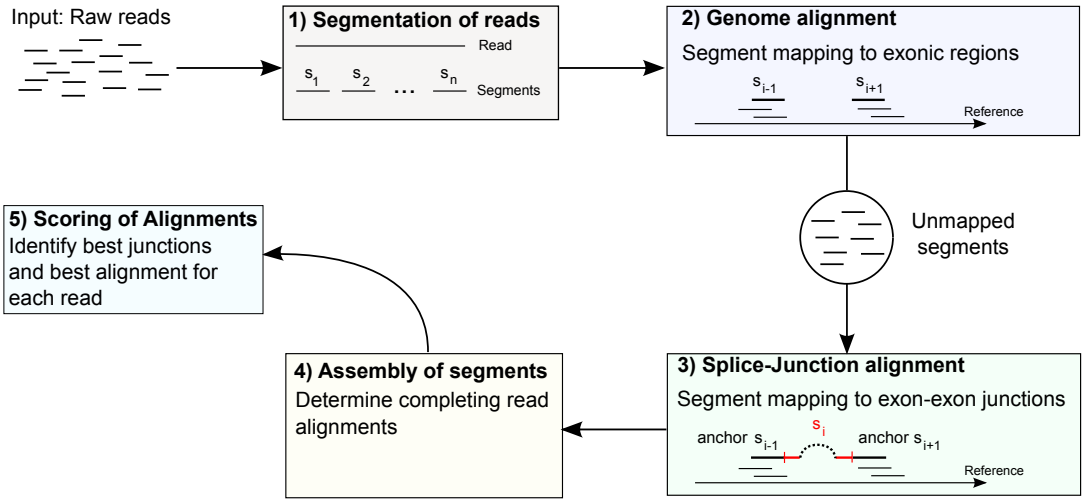

Figure 2.3: The mapping workflow of MapSplice. First, reads are partitioned into small segments (1) and subsequently aligned to the genome (2). Afterwards, aligned segments are used as anchor regions to determine spliced alignments for unmapped segments (3). Finally, completing alignments are assembled (4) and the best alignment is determined for each read (5).

2.1.3 MapSplice

MapSplice (Wang et al. [2010]) predicts canonical as well as non-canonical splice junctions de novo from RNA-seq data (see Figure 2.3 for an workflow overview). MapSplice starts by dividing the reads into small segments of equal length. Afterwards, the segments are aligned to the genome using Bowtie (Langmead et al. [2009]). Segments with at least one alignment are assumed to originate from exonic regions.

For reads with an unaligned segment s_i, MapSplice searches for spliced alignments. For this purpose, aligned segments of the same read are used as anchors. If segments s_{i-1} and s_{i+1} both have an alignment, then a spliced alignment for segment s_i is searched between s_{i-1} and s_{i+1} (see Figure 2.3). This strategy is denoted as the 'double-anchored' strategy. If only a single neighboring segment s_{i-1} has an alignment, then an alignment for a suffix of s_i is determined in a genomic window downstream of s_{i-1}. Similarly, if s_{i+1} is aligned, an alignment is determined for a prefix of s_i in a window located upstream of s_{i+1}. Subsequently, the same double-anchored strategy can be applied as described. Finally, all combinations of segment alignments are determined that result in the original read sequence within a certain genomic region.

In the last step, MapSplice assigns a score to every alignment, which is based on the number of mismatches and base call qualities. Furthermore, quality values are calculated for splice junctions in order to be able to distinguish between spurious and true junctions. Here, MapSplice assumes that the reads are uniformly distributed across the transcripts. Therefore, the quality value will be high if the respective junction is supported by reads with many different start positions on the genome. Finally, a combination of alignment score and junction quality is used to find the best mapping location for reads with multiple alignments.

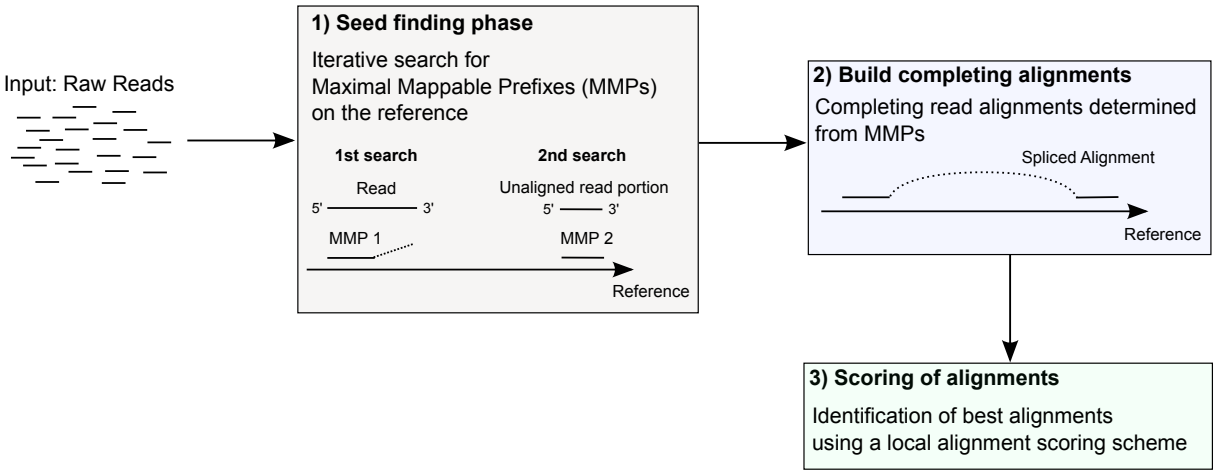

Figure 2.4: The mapping workflow of STAR. In contrast to MapSplice and TopHat, STAR does not rely on a short read alignment program (e.g. Bowtie) to determine alignments, but uses an uncompressed suffix array of the reference instead. First, STAR performs a seed finding step, which is a sequential search for Maximal Mappable Prefixes of each read (details provided in the text). Second, aligned seeds are gathered for each read and combined to produce completing read alignments. Finally, a local alignment scoring scheme is applied for identifying the best scoring alignments of each read.

2.1.4 STAR

STAR (Dobin et al. [2013]) was designed with the objective to map the vast amount of RNA-seq data of the ENCODE project (ENCODE Project Consortium [2012]) (> 80 billion reads) significantly faster than all existing approaches. For this purpose, the authors of STAR developed a novel RNA-seq mapping strategy that can be divided into three different steps (see Figure 2.4).

First, an uncompressed suffix array of a given reference genome is queried with the sequences of the reads. For every read sequence, the maximal mappable prefix (MMP) is determined. A MMP is the longest exact matching substring of a read, starting from the most $5'$ nucleotide. If a found MMP is shorter than the read sequence, this step will be repeated with the unaligned portion of the read until the unaligned portion is too small for another search.

In the second step, different MMPs of the same read are stitched together. For this purpose, MMPs are collected that are aligned in proximity to each other. Subsequently, STAR uses a dynamic programming algorithm to determine completing alignments that allow for mismatches and indels.

The last step of STAR starts by determining a score for every alignment based on a local alignment scoring scheme. Finally, every read is mapped to the position with the best scoring alignment. A read will be mapped to multiple locations, if there are several alignments of the same read with only small score differences from the best alignment.

2.1.5 GSNAP

GSNAP (Wu and Nacu [2010]) is a mapping approach that aligns reads by using a hash-based index of the reference. In contrast to other approaches, GSNAP does not align to a single reference, but to a reference space. This space is defined by all combinations of major and minor alleles of the reference derived from databases such as dbSNP (Sherry et al. [2001]). GSNAP has two modes to query the index, one for determining alignments with many mismatches and one for alignments with relatively few mismatches.

In the latter mode, GSNAP generates a minimal and non-overlapping set of q-mers for every read covering the complete read sequence. Subsequently, the hash-table of the reference is queried with these sets in order to obtain regions of candidate read alignments. In the next step, a lower bound on the mismatch count for completing read alignments is determined by using the pigeonhole principle of the non-overlapping set of q-mers of the reads. This means that the completing alignment of a read that has k missing q-mer hits, has at least k mismatches. Candidate alignments with a lower bound below a certain threshold are verified by determining the exact mismatch count via an alignment using the gmap algorithm (Wu and Watanabe [2005]).

In the second mode, the reference is queried with complete and overlapping sets of q-mers that cover each read sequence. The resulting hits are used to determine alignments that allow for many mismatches, which are used to detect indels and reads spanning splice-junctions. For this purpose, pairs of regions with q-mer candidate alignments are combined to completing read alignments such that a mismatch and gap penalty constrained is not exceeded.

Finally, a read is mapped to the position where it was aligned to with the fewest penalties (i.e. mismatches and gaps). As the original publication of GSNAP does not clearly state a workflow for the mapping procedure, a corresponding workflow diagram is not shown here.

2.1.6 RNASEQR

The mapping workflow of RNASEQR (Chen et al. [2012]) is very similar to the strategy of TopHat2 (see Figure 2.5 for an overview).

The workflow starts by aligning the reads to a reference transcriptome with Bowtie (Langmead et al. [2009]). If a read can be aligned uniquely, the respective alignment will be used as the final read mapping. RNASEQR defines a read as uniquely aligned if there is a single alignment that is best in terms of alignment score. For determining a score of an alignment, the Hamming distance is applied between the sequencing read and the reference at the location of the alignment. The Hamming distance calculates the minimum number of substitutions needed to transform one string into another. In the second step, unmapped reads are aligned to a reference genome using Bowtie. Again, only uniquely aligned reads to the genome are considered as mapped.

The final step has the aim to determine spliced alignments for reads that cross exon-exon junctions that are not part of the annotated transcriptome. For this purpose, RNASEQR

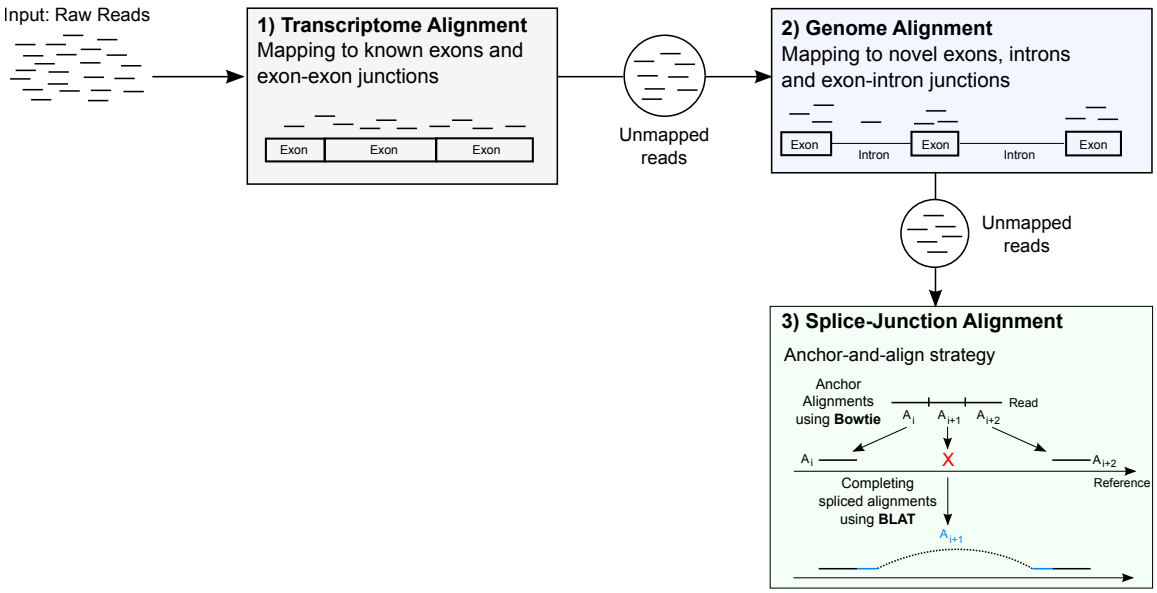

Figure 2.5: The mapping workflow of RNASEQR. First, reads are mapped to a reference transcriptome. Subsequently, the unmapped reads are aligned to the genome. Both alignment steps are performed with Bowtie. Finally, RNASEQR determines spliced alignments by aligning fragments of unmapped reads to the genome and determining completing alignments using BLAT.

uses a so called anchor-and-align strategy. For generating the anchors, unmapped reads are split up into fragments of 25 nucleotides length. These anchors are aligned in parallel to the reference transcriptome and genome using Bowtie. For reads with at least two anchors pointing to the same genomic region, BLAT (Kent [2002]) is used to determine completing spliced alignments.

2.2 A transcriptome-based mapping workflow

2.2.1 Overview

In the beginning of 2011, we started to develop our own workflow for mapping RNA-seq data. At that time, mapping programs for the de novo discovery of splice junctions (e.g. TopHat, MapSplice or GSNAP) were already available. However, the computationally expensive discovery of novel splice junctions was not our objective, but rather the fast and straightforward mapping of sequencing reads to known transcripts or genes. For this purpose, we developed a mapping approach that sequentially aligns the sequencing reads to ribosomal RNA (if provided), a known transcriptome and a reference genome using Bowtie (Langmead et al. [2009]). Finally, in an optional step, unmapped reads are mapped to known microbial and viral genomes for detecting potential contaminants contained in the data (see Figure 2.6 for an overview). Reads mapped in one of these steps will not be considered in any of the following steps, which guarantees a fast processing of the data.

Obviously, this mapping approach is very similar to the strategy of RNASEQR and

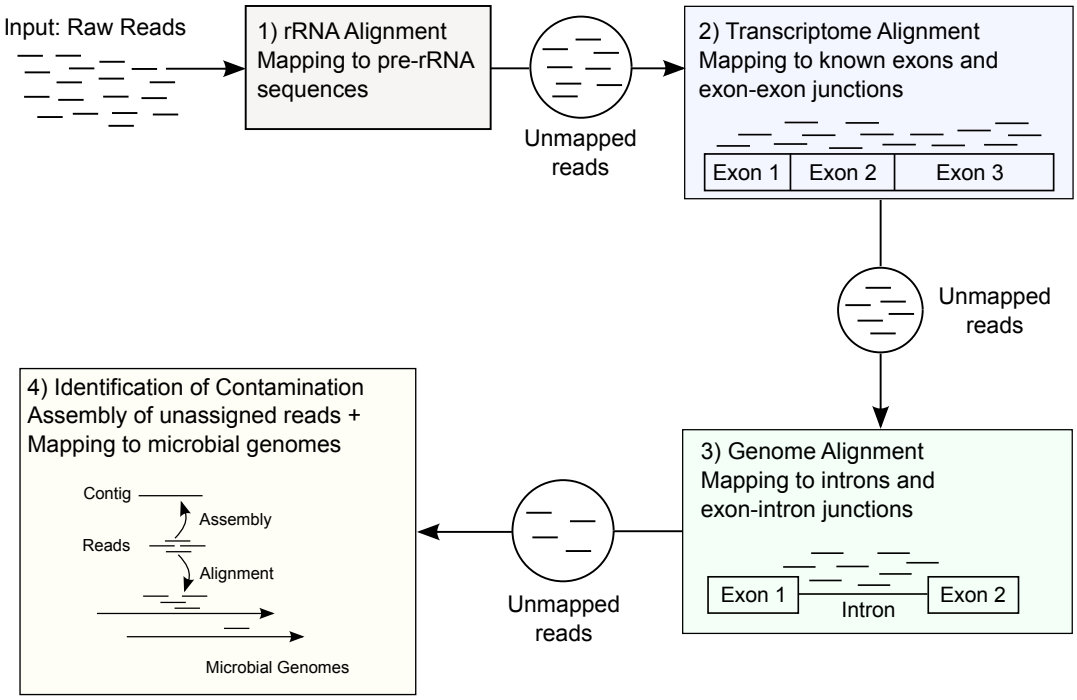

Figure 2.6: Overview of our transcriptome-based mapping workflow. All alignment steps are performed with Bowtie (Langmead et al. [2009]) and the assembly is determined with Abyss (Simpson et al. [2009]).

TopHat2, which were published at the end of 2011 and 2013, respectively. We did not try to publish our workflow because we realized that the strategy of a sequential mapping approach is not optimal, as it will be discussed later in section 2.5. In the following, we will describe the individual steps of our sequential mapping approach in more detail.

2.2.2 Filtering rRNA reads

The first step of the workflow is an optional filtering step, in which the read data is aligned to ribosomal RNA (rRNA) using Bowtie. In general, the amount of rRNA contained in a cell represents a large fraction of the total cellular RNA. For instance, in microbial and mammalian cells the rRNA fraction accounts for more than 95% (Peano et al. [2013]) and up to 90% (O'Neil et al. [2013]), respectively.

In most cases, researchers are more interested in mRNAs than rRNAs and therefore try to remove the rRNA prior to sequencing. However, depending on the sequencing protocol the amount of reads originating from rRNA may still contribute to a large portion of the whole data. Consequently, an alignment to the very short rRNA sequences can reduce the input for the following steps significantly and thus improve the overall running time of the workflow. We filter for reads that have at least one alignment to the rRNA sequences that does not exceed a predefined mismatch threshold.

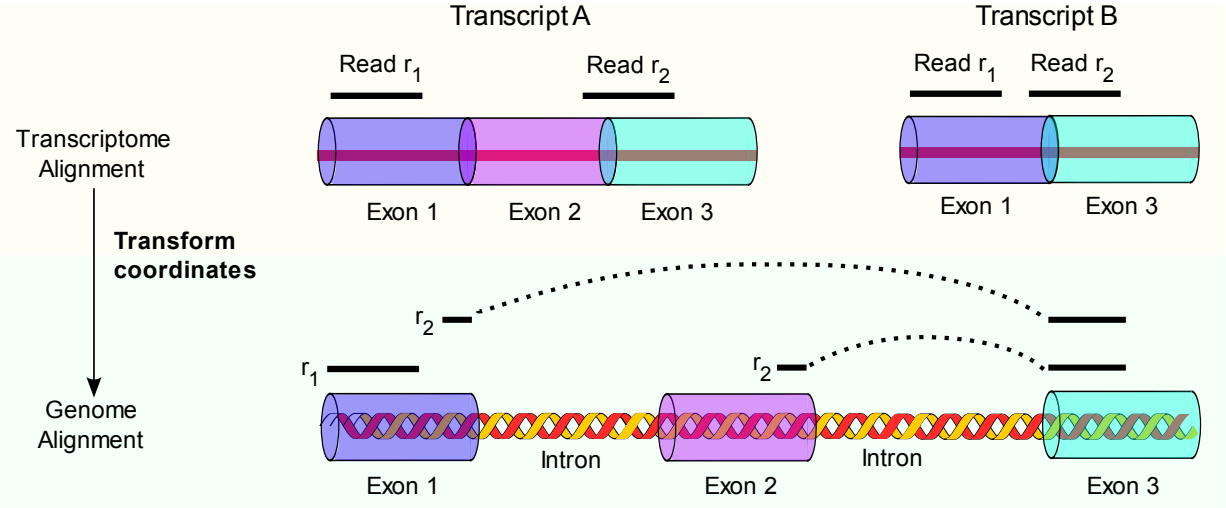

Figure 2.7: The two reads r_1 and r_2 can be aligned to transcript A and B, respectively. The alignments of the reads can be converted into genomic coordinates as described in the text. The transcriptomic alignments of read r_1 are transformed into a unique genome alignment on the genome. On the contrary, read r_2 has two different alignments after conversion to genomic coordinates. Therefore, read r_1 is mapped, whereas read r_2 is discarded.

2.2.3 Transcriptome and genome mapping

In the second step, the workflow aligns unmapped reads to known transcripts using Bowtie. Here, every alignment record contains information about the chromosome, transcript and the offset in the transcript a read was aligned to. Our workflow uses a hash-based data structure to store genomic coordinates of the set of exons composing each transcript. This allows to quickly convert transcriptomic alignment coordinates into the respective genomic coordinates. The coordinate conversion is necessary because we only map reads that are uniquely aligned to the genome. However, this can not be deduced directly from a transcriptome alignment as about 95% of human multi-exon genes undergo alternative splicing (Pan et al. [2008]). Reads that originate from exons shared by several alternative transcripts can be aligned to all of these transcripts. However, when transforming the transcriptomic coordinates of these alignments into genomic coordinates, the result can be a unique alignment (see Figure 2.7 for an example). Before alignment coordinates are converted, the workflow discards all alignments exceeding a predefined mismatch threshold. Furthermore, alignments of a read that have more mismatches than the best alignment, i.e. the alignment with the fewest mismatches, are also discarded. Eventually, alignment coordinates are converted as described and all uniquely aligned reads are considered as mapped.

In the third step of the workflow, unmapped reads are aligned to the reference genome. All alignments that exceed the mismatch threshold or have more mismatches than the best alignment of the respective read are discarded. Finally, all uniquely aligned reads are considered as mapped.

2.2.4 Identification of contamination

The last step aims at the identification of possible contaminants in the data. For this purpose, the workflow assembles the remaining unmapped reads to contigs using Abyss (Simpson et al. [2009]; Birol et al. [2009]). Subsequently, reads that are part of a contig are aligned to a collection of microbial genomes with Bowtie. If all reads that are contained in a contig can be aligned to a particular genome, there will be enough evidence for contamination in the data. Thus, the workflow maps all these reads to the respective microbe.

2.2.5 Setting up the workflow

Our mapping workflow can be configured via a single configuration file. The user is able to set the parameters of Bowtie (e.g. seed length, allowed mismatches) for every alignment step individually. Furthermore, the start and stop points of the pipeline can be specified. This may be useful if a repetition of an intermediate step is necessary or if the execution is not required for all workflow steps.

2.3 Application to 4sU-seq data

2.3.1 Background

Regulation of RNA levels of a cell may occur during the individual processes of RNA synthesis (transcription), RNA processing and RNA degradation. It is well known that changes in transcription (Wang et al. [2007]; Kim et al. [2009]) and degradation (Shalem et al. [2008]; Miller et al. [2011]) can have a significant influence on gene expression. However, it is little known about the contribution of RNA processing to changes in gene expression.

RNA processing can be monitored using 4sU-tagging, which is a method that uses a naturally occurring uridine derivative (4-thiouridine) to metabolically label newly transcribed RNA (Melvin et al. [1978]; Friedel et al. [2009]). The labeled RNA can then be separated from the pre-existing RNA using streptavidin-coated magnetic beads. Recent studies showed that 4sU-tagging is compatible with microarray analysis (Dolken et al. [2008]; Friedel and Dolken [2009]) and RNA-seq (Rabani et al. [2011]; Schwanhausser et al. [2011]).

The following analysis of a time-course experiment was taken from a publication by Windhager et al. [2012]. In this study, we showed that progressive 4sU-tagging combined with RNA-seq can be used to monitor the kinetics of RNA splicing and processing at the nucleotide resolution. Here, 4sU-tagging was combined with sequencing of newly and untagged RNA at five different time points after labeling. The first time point for sequencing was already 5 minutes after labeling, which is an ultrashort labeling time that has not been combined with RNA-seq before.

2.3.2 Data set

Our collaborators Lars Dölken and colleagues performed a time-course experiment of 4sU-tagging in DG75 human B-cells consisting of five samples with 60, 20, 15, 10, and 5 min of 4sU-tagging. Newly transcribed RNA from all five labeling conditions was purified and subjected to RNA-seq analysis using sequencing by ligation (SOLiD II, Applied Biosystems). In the following, we will refer to these samples as '5-min 4sU-RNA' to '60-min 4sU-RNA'. In addition to 4sU-RNA, total and untagged RNA following 60 min of 4sU-tagging were sequenced.

2.3.3 Methods

Read mapping

We mapped the reads with our transcriptome-based mapping workflow in the following way. First, reads were aligned to pre-rRNA sequences (18S, 5.8S, 28S, and spacer regions). The remaining unmapped reads were aligned to all Ensembl transcripts (Ensembl version 60) excluding pseudo-genes and haplotypes to identify exonic and exon-exon junction reads (aligned reads overlapping an exon-exon junction by >= 1 bp). Reads that remained unmapped after step two were aligned to the human reference genome (GRCh37/hg19) to identify intron and exon-intron junction reads (overlapping an exon-intron junction by >=1 bp). The following Bowtie settings were used for all three steps: seed region = first 20 bps, three mismatches allowed in the seed, five in the whole alignment.

Quantification of gene, exon, and intron expression levels

Expression levels of genes, exons, and introns were estimated using the standard RPKM measure (number of reads per kilobase of gene, exon, or intron per million mapped reads) (Mortazavi et al. [2008]). The number of reads mapping to a gene was determined as the total number of exon and exon-exon junction reads for this gene. To calculate RPKM values for exons and introns respectively, only reads mapping completely within this region were used. To account for problems in mapping reads to repetitive sequence regions, the effective length of exons and introns was used instead of the actual length. The effective length was calculated in the following way. First, in silico reads were simulated by sliding a window across gene regions with the size of the read length in the experiment (35 bps). Thus, the simulated read set contains exactly one read from each position in each gene. The simulated reads were then mapped using the same three-step procedure described above. The effective length was then defined as the number of positions within the respective region (exon, intron, or gene), which had exactly one correctly and uniquely mapped read starting at this position.

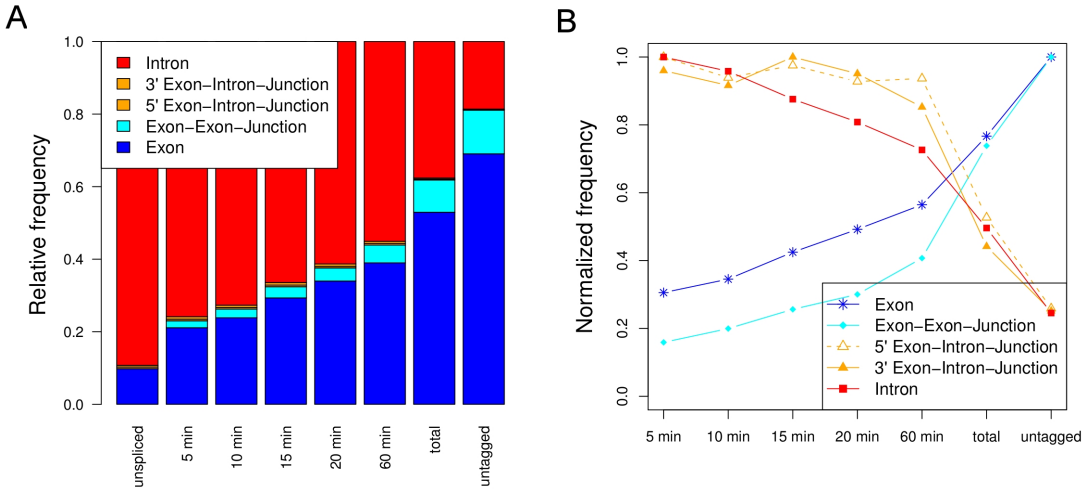

Figure 2.8: (A) Distribution of the number of reads mapped to exons, exon-exon junctions, exon-intron junctions, and intron regions for 5- to 60-min 4sU-RNA, total and untagged RNA. RNA in the untagged RNA samples is at least 60 min old. This visualizes the maturation of transcripts over time. The expected distribution of reads for completely unspliced RNA is shown in the left-most column (see Supplemental Methods of the original publication by Windhager et al. [2012]). (B) Normalized read frequencies were calculated by first dividing read numbers by the total number of reads on protein-coding genes in the corresponding sample. Subsequently, frequencies for a specific read type were divided by the maximum frequency observed for the corresponding read type in any sample.

2.3.4 Results

Gene feature mapping visualizes transcript maturation over time

We first assessed the contribution of intronic and exonic sequences in the seven RNA samples. As expected, the number of reads mapping to intronic sequences increased with reduced duration of 4sU-tagging from 18.9% in untagged RNA to 75.9% in 5-min 4sU-RNA (see Figure 2.8 A). As excised introns are generally believed to be rapidly degraded (Lamond et al. [1988]; Nam et al. [1997]; Clement et al. [1999]) this indicates the presence of large amounts of unspliced pre-mRNAs in the newly transcribed RNA samples. If none of the transcripts in 5-min 4sU-RNA had undergone any splicing events, the intronic reads would have been predicted to contribute ∼89% instead of 75.9% of all reads (see Figure 2.8 A). Thus, a substantial fraction of cellular transcripts in 5-min 4sU-RNA has already undergone splicing events with $> 65\%$ (conservative estimate) of all introns already decayed (see Supplemental Methods of the original publication on how this estimate was obtained).

Similar to the changes in the contribution of intronic reads over time, we observed a strong correlation between the number of reads crossing exon-intron or exon-exon junctions and the duration of 4sU-tagging (see Figure 2.8 B). Exon-intron junction reads result from unspliced or partially spliced transcripts. Accordingly, their contribution considerably decreased with the duration of 4sU-tagging (from 1.1% in 5-min 4sU-RNA to 0.29% in untagged RNA). Conversely, the frequency of exon-exon junction reads increased from 1.9% in 5-min 4sU-RNA to 12% in untagged RNA.

Distinct classes of introns are defined by their splicing kinetics

To investigate differences in the kinetics of intron processing, we first focused on the most highly expressed genes as intron expression levels in newly transcribed RNA, although substantially higher than in total RNA, are much lower than the expression levels of the surrounding exons. This is due to the large fraction of introns ($> 65\%$) already spliced and decayed in 5-min 4sU-RNA. Expression levels of genes were quantified in terms of reads per kilobase of gene per million mapped reads (RPKM) after normalizing for mappability (see Methods (2.3.3)), and the analysis was focused on genes with an RPKM ≥ 11 in all RNA samples (525 genes). For these genes, we distinguished between introns absent (RPKM < 0.5: 1014 introns) or present (5838 introns) in 5-min 4sU-RNA. Even after excluding 50 absent introns ($\sim 5\%$) that were shorter than the read length (35 bps) and, thus, could not contain any intronic reads, absent introns were significantly shorter than the present ones (Wilcoxon test, P-value $< 10^{-15}$). Furthermore, they were located closer to the $3'$ end of the gene than present introns (Wilcoxon test, P-value $= 0.0042$) with 12% of the absent introns being the last intron of the gene compared with 7% for present introns (Fishers exact test, P-value $< 10^{-6}$). This suggests that at least some of these introns were part of longer transcript versions that were not transcribed in this form in the DG75 cells.

For other introns, possible explanations for their absence in 5-min 4sU-RNA might be (1) very fast co-transcriptional splicing, (2) problems in sequencing, or (3) problems in mapping these parts of the pre-mRNA, e.g., due to repetitive sequences. Interestingly, in many absent introns, both neighboring exons were well expressed and precisely delimited. This indicates rapid co-transcriptional splicing and intron degradation rather than sequencing bias. In addition, there was no significant increase for the absent introns in the frequency of repetitive sequences identified by RepeatMasker (Smit et al. [1996]) or the frequency of non-unique read mappings. Notably, the fraction of absent intron positions contained within repetitive sequences was actually significantly smaller than for present ones (Wilcoxon test, P-value $< 10^{-9}$). These analyses confirm that numerous transcripts in 5-min 4sU-RNA ($> 65\%$) had already been spliced and their introns had been degraded.

2.4 Application to other data sets

In a recent study by Marcinowski et al. [2012], sequencing data was analyzed for another 4sU-seq experiment. Here, the investigated mouse cells were infected with the murine cytomegalovirus and RNA-seq was performed at different time points after infection. Our workflow was applied in the same way as described in section 2.3.3. However, we added an alignment to the reference genome of the virus as a fourth mapping step. In summary, viral and host gene expression was monitored in parallel over time. We found interesting responses (e.g. a fast inflammatory-response) of the host to the infection and revealed novel insights into gene regulation of the cytomegalovirus during infection.

Most recently, the workflow was applied for mapping sequencing reads originating from an RNA-seq as well as a ChIP-seq experiment (Hunten et al. [2015]). ChIP-seq is a

method by Johnson et al. [2007] that applies Chromatin Immunoprecipitation (ChIP) for identifying regions on the genome bound by specific proteins. Subsequently, these regions are sequenced using NGS technology. In the study by Hunten et al. [2015], ChIP-seq was applied for the genome-wide identification of binding sites of the tumor suppressor gene p53. Furthermore, RNA-seq was performed in cells with and without induced p53 gene expression, respectively. For mapping the RNA-seq data, we applied the same three-step procedure as described in section 2.3.3. For the ChIP-seq data, the transcriptome alignment step was skipped. By combining ChIP-seq and RNA-seq analyses, we were able to identify p53 target genes and study the effect of p53 activation for these genes at the same time.

2.5 Drawbacks of previous mapping approaches

One reason why our sequential approach is so fast is that the reads already mapped in an intermediate alignment step will not be considered in any of the following steps. However, this strategy, which in variation is also used by other RNA-seq mapping approaches, has several drawbacks. These drawbacks will be discussed in the following.

2.5.1 Exon-intron junction vs. splice junction alignments

The first problem occurs for sequencing reads that can be aligned to an exon-intron junction as well as to a splice junction. An example for such a scenario is given in Figure 2.9. Here, a read can be aligned continuously to the genome. This results in an alignment that overlaps a boundary between an exon and an intron. However, the same read can also be aligned with a spliced alignment to the genome or a continuous alignment to the transcriptome. Our study of the time course-experiment (see section 2.3) showed that both mapping types are relevant when analyzing a mixture of unspliced and spliced transcripts. The continuous genome alignment indicates an unspliced transcript. On the contrary, the spliced alignment provides evidence for a spliced transcript.

A mapping approach starting with a transcript alignment and subsequently aligning only the remaining unmapped reads to the genome (e.g. our workflow or RNASEQR) will always examine only one of the two mapping possibilities. Therefore, it possibly determines the wrong mapping for the read. TopHat2 also starts with an alignment to a known transcriptome. However, TopHat2 is able to identify both alignments by re-aligning aligned reads that have an alignment with an edit distance larger or equal to a user-defined threshold. If this threshold is set to 0, all reads will be aligned in every step of TopHat2. However, per default this threshold is set to infinity, because otherwise the running time of TopHat2 increases dramatically. Furthermore, TopHat2 assumes that the spliced alignment is the correct one, and therefore it would most likely discard the continuous genome alignment anyway.

There are genome-based mapping approaches that solely align to the reference genome and perform a de novo prediction of spliced alignments (e.g. TopHat or MapSplice). At

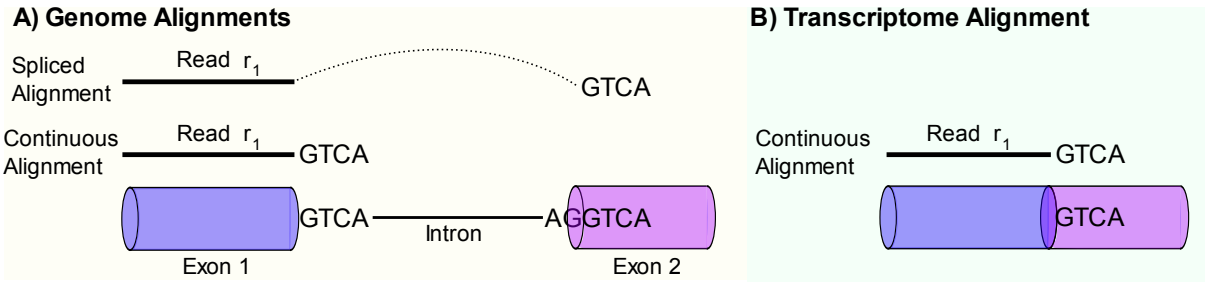

Figure 2.9: (A) This figure shows a read r_1 that has both, a spliced alignment and a continuous alignment to the genome. The latter overlaps an exon-intron junction and the spliced alignment overlaps an exon-exon junction. (B) The same read r_1 can be continuously aligned to the transcriptome. Here, only the alignment overlapping the exon-exon junction can be determined.

first glance, this strategy appears to circumvent the problem of approaches starting with a transcriptome alignment. However, when the mapping workflows of these programs are analyzed in more detail, it becomes apparent that they also have problems in determining both alternative alignments. For instance, TopHat starts by determining continuous read alignments to the genome and subsequently uses only unmapped reads for the prediction of spliced alignments. Thus, it will not determine the spliced alignment of the read. MapSplice starts by aligning small segments of the read to the genome. Following the segment alignment, MapSplice predicts spliced alignments for reads that have unaligned segments. However, in the example of Figure 2.9 there will be no unaligned segment of the given read and therefore MapSplice will not determine a spliced alignment.

2.5.2 Parent gene vs. pseudogene alignments

The next problem concerns the mapping of reads to genes that have an associated pseudogene in the genome. Pseudogenes are copies of gene sequences that are integrated in the genome at a new locus. In general, pseudogenes include some dysfunctional mutations, which lead to the loss of protein coding ability (Mighell et al. [2000]). A subtype of pseudogenes are so-called processed pseudogenes, which are reverse-transcribed copies of spliced mRNAs of the respective parent genes that were inserted into the genome. Processed pseudogenes also lost their ability to code for a protein, but not necessarily due to mutations. Here, additional reasons are incomplete copies of the original mRNAs or missing regulatory sequence elements of the original gene (Vanin [1985]). The human genome contains about 8000 processed pseudogenes, which have sequence similarities of up to 86% to their parent genes (Zhang et al. [2003]). Thus, it is likely that a read that can be aligned to a parent gene may also have an alignment to an associated processed pseudogene.

For instance, the read depicted in Figure 2.10 has a spliced alignment to a parent gene and a continuous alignment to the respective processed pseudogene. We already discussed that most of the presented genome-based mapping approaches do not determine a spliced alignment if a continuous genome alignment also exists. However, in most cases mapping reads to pseudogenes is wrong, as it has been suggested that only a very small fraction of

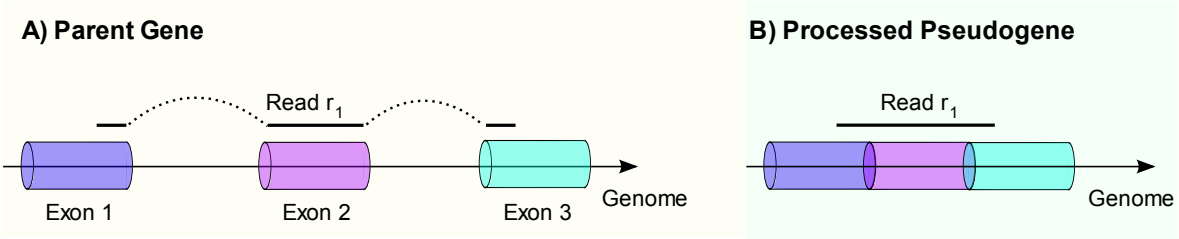

Figure 2.10: (A) This figure shows a gene that has a single transcript, which is composed of three exons. The read r_1 is a spliced read that spans all three exons. (B) A processed pseudogene of the gene depicted in (A) is shown. The spliced mRNA of the transcript has been reverse transcribed and inserted somewhere else into the genome. Thus, the read r_1 can be continuously aligned to the pseudogene variant of the gene.

processed pseudogenes are transcribed (Harrison et al. [2005]).

There may not be a problem in determining the alignment to the parent gene if the mapping approach starts by aligning the reads to the transcriptome (e.g. our workflow). However, if the transcriptome also contains the transcripts of (processed) pseudogenes, the mapping approach would end up in determining the alignment to both, the parent gene and to the pseudogene. In that case, the mapping approach can examine both alignments, and hence needs a strategy for resolving ambiguously aligned reads, which will be discussed in the next section.

2.5.3 Mapping of ambiguously aligned reads

The first two described drawbacks of existing mapping approaches concerned the missing ability of examining all possible alignments of a sequencing read. However, even if a mapping approach investigates all possible alignments of a read, it is not guaranteed that it eventually maps the read to the correct location. For instance, all presented approaches are able to determine multiple continuous read alignments of the same read to repetitive regions on the genome, resulting in alignments to different loci. In the following, we denote reads that can be aligned to multiple locations on the genome as ambiguously aligned reads.

Our workflow and RNASEQR discard ambiguously aligned reads in order to minimize the number of false positively mapped reads. However, this strategy may underestimate expression values for genes that contain repetitive regions (Robert and Watson [2015]; Finotello and Di Camillo [2015]). To address this problem, we performed a gene length normalization in the study presented in section 2.3. Nevertheless, the loss of information due to discarded reads can still influence downstream analyses.

Other mapping approaches also have problems in resolving ambiguous read alignments. STAR, TopHat 2, MapSplice and GSNAP implement scoring systems for determining the best mapping location for such reads. However, these programs generally assign the same score to continuous read alignments with the same number of mismatches. As consequence, all mentioned approaches would map reads that originate from a repetitive region on the genome to multiple locations, from which we know that only one is correct.

2.6 Conclusion

In this chapter, we introduced several state-of-the-art programs for mapping RNA-seq data. Most of them rely on short read alignment programs such as Bowtie or BWA and implement sophisticated strategies for determining spliced alignments. Here, STAR and GSNAP are an exception as both programs implement their own methods for determining alignments of sequencing reads. We also introduced a mapping workflow we developed, which maps the reads sequentially to one reference sequence after another. A similar strategy is pursued by RNASEQR, which was developed at the same time as our workflow. Both approaches first align sequencing reads to a given transcriptome using Bowtie and therefore determine alignments of reads overlapping annotated exon-exon junctions in a straightforward and very fast way.

We demonstrated the usability of our workflow by presenting a study in which we analyzed data from a time-course RNA-seq experiment. In this experiment, sequencing reads were obtained from fully spliced, partly spliced and completely unspliced transcripts. We showed that the mapping determined with our workflow can be used to visualize the transcript maturation over time. Furthermore, count data derived from the mapping was used to quantify gene, exon and intron expression levels. Based on the intron expression levels, evidence for fast co-transcriptional splicing was found.

Finally, in the last part of the chapter, problems were discussed that our and other RNA-seq mapping approaches have in common. These problems arise for sequencing reads with several possible alignments. From a recent study (Li et al. [2010]) we know that depending on the complexity of the transcriptome and read lengths a significantly large fraction (between 17% to 52%) of reads in a dataset can be aligned to several different genes. Therefore, analyses based on read mapping (e.g. gene expression quantification) are directly influenced by the strategy of the underlying mapping approach for mapping such reads.

We found that the different approaches either fail in determining all possible alignments of a sequencing read or have no general concept for resolving ambiguously aligned reads. The former can result in falsely mapped reads if the correct alignment is not part of the evaluated alignments. In the latter case, sequencing reads originating from repetitive regions cannot be confidently assigned to a particular region on the reference and are either discarded or all possible alignments are included in the output.

Chapter 3

ContextMap: Fast and accurate context-based RNA-seq mapping

Motivation: In this chapter we introduce ContextMap, an RNA-seq mapping approach that was designed for addressing the common drawbacks of other mapping approaches that were presented in the previous chapter (see section 2.5 of chapter 2 for details). The basic idea of ContextMap is not to consider each read individually to decide about its mapping location, but to take information provided by all other reads aligned to the same genomic region – the so-called *read context* – into account. This mapping strategy allows for an accurate resolution of ambiguously aligned reads, as we demonstrated in a proof of concept study in 2012 (Bonfert et al. [2012]). In this study, we developed a first prototype implementation of ContextMap that aimed at improving already existing mapping results determined by other programs such as TopHat (Trapnell et al. [2009]) or MapSplice (Wang et al. [2010]). However, by relying on other programs to provide an initial mapping result as input, ContextMap was not able to consider all possible alignments of each read.

Therefore, we extended our implementation to a standalone program that was able to determine all possible initial read alignments on its own using a modified implementation of the Bowtie alignment program (Langmead et al. [2009]). We were able to show that the standalone version of ContextMap allows parallel mapping against several reference genomes, e.g. the human host and infecting pathogens, in a straightforward way (see chapter 4 and Bonfert et al. [2013]). Nevertheless, this ContextMap version still had some substantial drawbacks. First, it was not able to map reads spanning over more than two exons or to detect reads that contain indels. Second, due to the dependency on a specific and modified version of the Bowtie alignment program, ContextMap could not benefit of novel developments (e.g. improved alignment sensitivity) in the area of short read alignment software.

In this chapter, we present ContextMap 2 (Bonfert et al. [2015]), an extension of the ContextMap algorithm. ContextMap 2 determines initial read alignments with unmodified short read alignment programs such as Bowtie, Bowtie 2 (Langmead and Salzberg [2012]) or BWA (Li and Durbin [2009]). Already existing mapping procedures of ContextMap were further improved and newly developed methods integrated into ContextMap 2. In addition

to the resolution of ambiguously aligned reads, ContextMap 2 is able to accurately map reads spanning an arbitrary number of exons and to perform a context-based prediction of indels. Furthermore, the ContextMap 2 implementation provides a plug-in structure for an easy integration of newly developed alignment programs. We show using synthetic and real-life data that ContextMap 2 can compete with the best state-of-the-art read mapping approaches in terms of running time and mapping accuracy.

Publication: This chapter was published in BMC Bioinformatics (Bonfert et al. [2015]). I moved parts of the Supplementary Material of the original publication into the methods section to provide a complete description of the methods implemented in ContextMap 2. Furthermore, I adapted the layout of the text, added section 3.2.3 and Figure 3.2 to the chapter and applied some minor changes to the text.

Author contributions: Gergely Csaba (GC) and I designed and GC implemented the first prototype of ContextMap (Bonfert et al. [2012]). I implemented the first standalone version of ContextMap with the exception of the modification of Bowtie, which was implemented by GC. I implemented ContextMap 2 (Bonfert et al. [2015]) independently and without outside assistance. This includes, in particular, the development and implementation of novel methods for the prediction of reads that span over an arbitrary number of exons. Furthermore, I implemented a context-based prediction of indels and performed a evaluation of ContextMap 2 and of other mapping approaches on synthetic and real-life data. Caroline C. Friedel (CCF) and I analyzed and discussed the results of this evaluation. CCF wrote the article on the proof of concept study (Bonfert et al. [2012]) and CCF and I co-wrote the article that is presented here (Bonfert et al. [2015]). CCF supervised the work and Ralf Zimmer and GC helped to revise the manuscript. Evelyn Kirner prepared the user manual and implemented scripts for example calls of ContextMap 2.

3.1 Background

Sequencing of RNA using next generation sequencing technology (RNA-seq) has become the standard approach for analyzing the transcriptomic landscape of a cell (Wang et al. [2009]; Ozsolak and Milos [2011]). The first step in RNA-seq data analysis generally consists in determining the transcriptomic origin of the sequenced reads (=read mapping) (Garber et al. [2011]), i.e. the best alignment of each read against a transcript. Here, the major challenge results from the fact that even for well-annotated species not all transcripts, in particular rare or non-coding transcripts (Djebali et al. [2012]), are known. Thus, alignment against known transcript sequences using short read alignment programs such as Bowtie (Langmead et al. [2009]) cannot identify reads from novel transcripts, in particular spliced reads crossing novel exon-exon junctions. Unspliced reads, in contrast, are easily mapped using genome alignments.

Currently, many different RNA-seq mapping algorithms are available, such as TopHat (Trapnell et al. [2009]), TopHat2 (Kim et al. [2013]), or MapSplice (Wang et al. [2010])

(see also Alamancos et al. [2014] for an overview). In most cases, these approaches combine alignment against reference sequences (i.e. a genome and/or transcriptome) using short read aligners, such as Bowtie (Langmead et al. [2009]) or Bowtie 2 (Langmead and Salzberg [2012]), with sophisticated strategies for identifying spliced reads crossing exon-exon junctions. A common strategy for this purpose involves splitting reads into smaller segments before aligning and is used e.g. by TopHat2 and MapSplice. Other mapping approaches, such as STAR (Dobin et al. [2013]) or GSNAP (Wu and Nacu [2010]), use their own alignment methods to identify spliced reads without fragmenting read sequences.

Independent of the strategy for identifying spliced reads, existing RNA-seq mapping approaches were implemented to use only specific short read alignment programs, in most cases Bowtie. Thus, they cannot be easily extended to make use of novel developments in short read alignment, e.g. Bowtie 2 (Langmead and Salzberg [2012]) or BWA (Li and Durbin [2009]), which improve alignment speed, recall or precision (Lindner and Friedel [2012]). Furthermore, they generally identify the best alignment for each read based only on the number of mismatches and do not take into account information provided by alignments of other reads. As a consequence thereof, the mapping problems that were described in the previous chapter (see section 2.5) will arise for those approaches. We recently proposed a different approach, ContextMap, to identify the most likely mapping for a read based on all reads aligned to the same general location, the so-called context (Bonfert et al. [2012]). This approach also has the advantage that it allows parallel mapping against several reference genomes in a straightforward way (Bonfert et al. [2013]).

In this chapter, we present ContextMap 2 (Bonfert et al. [2015]), an extension of the ContextMap strategy, which among other improvements addresses the problem of integrating different short read alignment programs. The key features of ContextMap 2 are:

(i) It accurately predicts spliced reads and resolves ambiguous read alignments by considering the context of each read.

(ii) It provides an easy-to-use plug-in interface for integrating different short read alignment programs into the mapping workflow. This flexibility guarantees that ContextMap can be quickly adapted to newly developed read alignment algorithms.

(iii) It extensively uses local read alignment options of novel short read alignment programs such as Bowtie 2 or BWA to detect spliced reads, which overlap an arbitrary number of exon-exon junctions.

(iv) It precisely predicts the exact position of deletions or insertions (indels) by using the information provided by all reads in the same context.

We evaluated the performance of ContextMap 2 using Bowtie, Bowtie 2 and BWA as integrated alignment programs on both simulated and real-life RNA-seq data used by the RGASP consortium in a recent evaluation of RNA-seq mapping programs (Engstrom et al. [2013]). The comparison of ContextMap 2 to the best performers of this study showed

that it combined high recall with high precision on read placement, splice junctions, multi-junction reads and indels. While individual competing RNA-seq mapping programs outperformed ContextMap 2 on some of these tasks, none was consistently better or performed comparably well in all of them. Furthermore, ContextMap 2 was generally at least twice as fast as the best competing methods.

3.2 Methods

3.2.1 Overview of ContextMap 2

ContextMap 2 is based on the ContextMap approach for RNA-seq read mapping (Bonfert et al. [2012]). Here, the central concept is the so-called read context, which is defined as a set of reads all originating from the same stretch of the genome and likely corresponding to transcripts of the same or overlapping genes. The first implementation of ContextMap was focused on improving initial mappings provided by other RNA-seq mapping programs, but has more recently been extended into a standalone version that also allows parallel mapping against several reference genomes (Bonfert et al. [2013] and chapter 4).

Similar to other mapping approaches, both of these implementations used a modified version of Bowtie for alignment. Thus, ContextMap suffered from the same problem as most state-of-the-art mapping approaches that newly developed short read alignment programs could not be easily integrated to replace the used Bowtie version. Furthermore, variable read lengths were not supported and reads crossing multiple exon-exon junctions or containing indels were not mapped. All of these problems are addressed by ContextMap 2 (Bonfert et al. [2015]).

In the following, an overview of the five steps of the ContextMap 2 workflow is presented (see Figure 3.1). The details of each step are described following this overview.

Step 1: Determination of initial alignments

This step includes both the determination of ungapped read alignments against one or several genomes using the integrated short read alignment program, e.g. BWA, as well as the extension of these alignments to alignments containing a splice junction (=split read alignments, see Figure 3.1 A). For this purpose, ContextMap 2 first performs a seeded alignment of all reads against the reference sequences with user defined seed values of 20-30 bps. Here, ContextMap 2 can use programs that determine only end-to-end alignments (e.g. Bowtie) as well as programs that also determine local alignments (e.g. Bowtie 2 and BWA). An end-to-end alignment starts at the read start and ends with the read end. In contrast, a local alignment allows unaligned prefixes or suffixes of the read if this improves the alignment score.

Parameters of the underlying alignment program are set such that all alignments for which the seed can be aligned are retained, allowing for multiple alignments of each read. The resulting alignments are then classified into four categories:

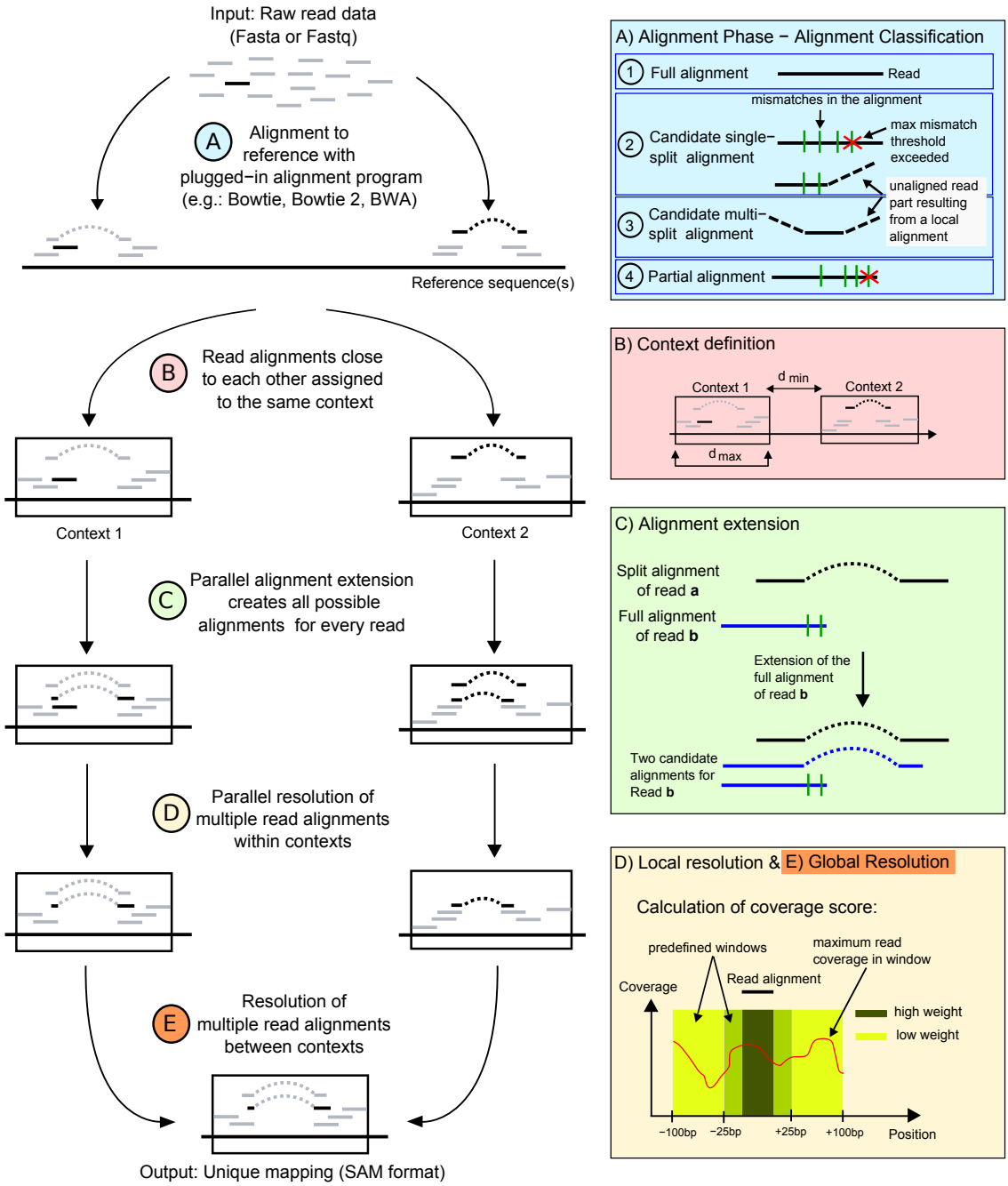

Figure 3.1: Workflow of ContextMap 2. (A) Reads are aligned to the reference sequence(s) using the integrated short read alignment program and the resulting alignments are classified into 4 different categories (top box, right side: full alignment, candidate single-split alignment, candidate multi-split alignment, and partial alignment). Dashed lines indicate unaligned sequence parts resulting from local alignments. Candidate single- and multi-split alignments are extended to split alignments using the sliding window approach (Figure 3.3). (B) Alignments less than d_{min} apart are assigned to the same context. The maximum context size d_{max} can be defined by the user (default is the average length of a mammalian mRNA). (C) Alignment extension of full (green box) and split alignments (see Methods section) to determine all valid alignments for a read. (D) + (E) Resolution of the best alignment for each read first within each context (D, local resolution) and then between all contexts (E, global resolution). For this purpose, a support score is calculated based on closely located alignments of other reads (bottom box, right side, and Methods section).

(a) *Full alignment:* if the read could be aligned end-to-end to the genome with a maximum number of mismatches (defined by the user).

(b) *Candidate single-split alignment:* if the seed could be aligned at the start or end of the read, the end-to-end alignment of the read contains more than the allowed number of mismatches and the last allowed mismatch is at least a predefined distance from the end of the alignment. If the integrated short read alignment program also produces local alignments, unaligned read positions are counted as mismatches for this classification.

(c) *Partial alignment:* if the same criteria apply as in (b) but the last allowed mismatch is less than the predefined distance from the alignment end.

(d) *Candidate multi-split alignment:* if only a local alignment could be determined with both a prefix and suffix of the read unaligned.

Following this classification, candidate single-split and multi-split alignments are extended to complete split alignments as described further below.

Step 2: Context definition

The alignments identified in the previous step are used to define contexts. For this purpose, read alignments are clustered into a context if their start or end positions on the genome are at most a maximum distance apart (Figure 3.1 B). Contexts are treated independently of each other until step 5. This allows both mapping read sequences against several reference genomes, e.g. of the human host and infecting pathogens (see chapter 4 or Bonfert et al. [2013]), as well as efficient parallelization of steps 3 and 4. Here, multiple alignments of each read to the same context or different contexts are allowed, which will be resolved in steps 4 and 5.

Step 3: Alignment extension

Once contexts have been defined, additional alignments are determined for each read based on the alignments found in the first step (Figure 3.1 C). This alignment extension is performed in parallel for different contexts. Its objective is to identify all valid alignments for each read with a maximum number of mismatches, such that the best supported alignment can be selected in the subsequent steps.

For this purpose, full and partial read alignments are checked for an overlap with split alignments of other reads. If overlaps are found, additional split alignments are created for the corresponding reads using the splice junctions indicated by the overlapping split alignments. Furthermore, all possible split alignments are generated for each read for which at least one split alignment was identified in step 1 (see section 3.2.7 for details). In all cases, only alignments are used that do not exceed the maximum mismatch criterion. At the end of this step, several different alignments have been created for each read, resulting in multiple alignments both within and between contexts.

Step 4: Local resolution of alignments within contexts

In this step, the best alignment for each read is determined within each context by taking other read alignments into account (Figure 3.1 D). For this purpose, the best supported splice sites among overlapping splice sites are determined first using a score based on the number of supporting reads, the number of mismatches and known splice signals (see section 3.6 for details). Split read alignments not using any of the three best supported splice sites are discarded.

Subsequently, a support score is calculated for the remaining read alignments based on the number of reads aligned within and around the read alignment. In principle, the support score is a weighted sum of maximum read coverages in predefined windows around the read alignment (see section 3.2.9 for details). Among several alternative alignments for the same read within each context, the one with the largest support score is then chosen.

Step 5: Global resolution of alignments between contexts

In this final step, multiple read alignments to several different contexts are resolved as in step 4 after recalculating support scores based on the read alignments chosen for each context (Figure 3.1 E). Thus, at the end of each step, each read is aligned to only one position in (at most) one context. If more than one reference sequence was provided, this will also automatically result in the choice of one reference sequence of origin for each read.

3.2.2 Plug-in structure of ContextMap 2

ContextMap 2 provides a plug-in interface which allows for integrating any short read alignment program without modification if it meets the following requirements:

(i) The alignment program has to support seeded alignments with adjustable seed lengths to allow use of different seed lengths in different steps of ContextMap 2.

(ii) The alignment program has to provide a tool to prepare an index of any reference sequence. Indexing reference sequences is a common strategy of all state-of-the-art short read alignment programs to speed up alignment.

(iii) If the read alignment program includes an option to identify indels, it must be possible to deactivate this option. ContextMap 2 uses its own context-based strategy for predicting the exact position of indels.

(iv) The output has to be in SAM format (Li et al. [2009]).

The interface for plugging in a short read alignment program into ContextMap 2 is composed of three methods, two for performing alignments at different steps of ContextMap 2 and one for indexing reference sequences. Implementing the interface requires implementing methods for managing the external program calls. In addition, the alignment methods have to collect the determined alignments. For this purpose, two classes can be reused that

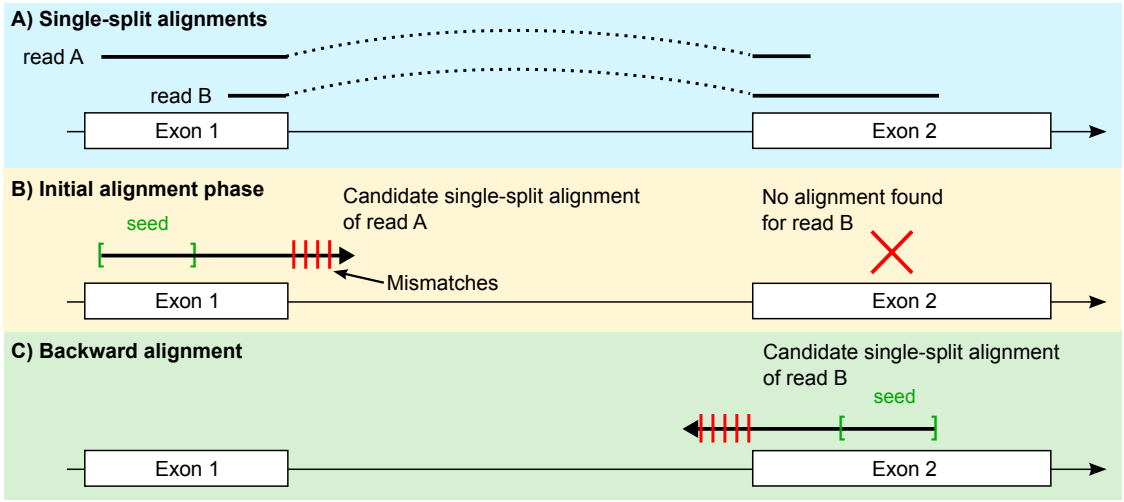

Figure 3.2: Candidate single-split alignment detection with alignment programs that only use a single seed region at the start of the read (e.g. Bowtie). (A) Two reads crossing the same exon-exon junction. More than half of the read sequence of read A can be aligned to exon 1 and only the remaining part to exon 2. On the contrary, only a small part of read B can be aligned to exon 1 and more than half of the sequence to exon 2. (B) During the initial alignment phase a candidate single-split alignment of read A is detected. However, an alignment of read B cannot be determined if only a single seed region at the start of the read is considered. (C) In the backward alignment step, ContextMap re-aligns the reverse complemented read sequence of read B to the genome. Due to the reversion of the read a candidate split alignment starting from the read end can be determined.

perform these tasks for Bowtie, Bowtie 2 and BWA, which have already been integrated in ContextMap 2.

3.2.3 Detection of candidate single-split alignments

The detection of reads crossing a single exon-exon junction is based on the fact that these reads overlap two exons only. Therefore, a complete single-split alignment can be divided into two parts. One part consists of at least half of the read sequence, while the other part consists of the remaining part of the read sequence (see Figure 3.2 A). In the initial alignment phase, the split detection starts by determining alignments for the larger part containing at least half of the read sequence using relatively large seed sizes. An alignment is classified as a candidate single-split alignment if the seed can be aligned at the start or end of the read, but the whole alignment exceeds the allowed number of mismatches. Furthermore, the last allowed mismatch has to be more than a predefined distance away from the alignment end. If ContextMap finds candidate single-split alignments of a read, then it will determine completing alignments of the remaining part of the read in a more sensitive alignment step (see section 3.2.4).

Novel alignment programs such as Bowtie 2 or BWA apply a so-called multi-seed heuristic for determining alignments. Here, several different regions of the read sequence are considered as seeds. This includes seed regions at the read start as well as at the read end.

Thus, ContextMap 2 using Bowtie 2 or BWA detects candidate single split alignments with the seed aligned at the start or at the end of the read during the initial alignment phase. However, some alignment programs, e.g. Bowtie, consider only a single seed region starting at the beginning of the read. These programs can miss candidate single-split alignments (see Figure 3.2 B for an example). For addressing this problem, we apply an additional backward alignment step for alignment programs that place the seed only at the start of the read (see Figure 3.2 C). The backward alignment is performed with reverse complemented read sequences. This allows to determine read alignments starting from the read end and thus to increase alignment sensitivity.

3.2.4 Detection of complete single-split alignments

At the end of the initial alignment phase, ContextMap 2 extends candidate split alignments to complete single-split alignments, i.e. alignments crossing one exon-exon junction only, using a so-called sliding window approach (Figure 3.3 A). This sliding window approach works in the following way: The sliding window is initiated at the left-most candidate split alignment on a chromosome and is extended to contain any overlapping alignment until a pre-defined maximum window length is exceeded. All candidate single-split alignments within this window are then extended to complete split alignments as described below. Afterwards, the current window is discarded and the next window is determined starting at the next candidate split alignment not completely contained in the previous window. This is repeated until all candidate split-alignments have been processed.

To determine the complete split alignments within each window, an index is built for the used short read alignment program covering the part of the reference sequence within the current window. This sequence is extended by x nucleotides (x = average intron size, can be defined by the user) downstream of the window if a candidate split alignment with the seed at the read start ends too close to the window end (i.e. the distance is less than the average intron size x). This allows finding split alignments that start within the window but end downstream of the window end. Similarly, an upstream sequence is added to the index if a candidate split alignment with the seed at the read end begins too close to the window start.

Using this dynamically built index and the corresponding short read alignment program, completing alignments of the unaligned read part are determined for each candidate split alignment within the sliding window (Figure 3.3 A). This restricts the search space to a region covering only one or very few genes, allowing the use of smaller seed lengths of 10-15 bps. Since the window is very small and only a relatively small number of reads is covered by the window, this step is very fast. The original candidate split alignment and the completing alignment for each read are then combined into one split alignment and included in the set of initial alignments in addition to the full and partial alignments.

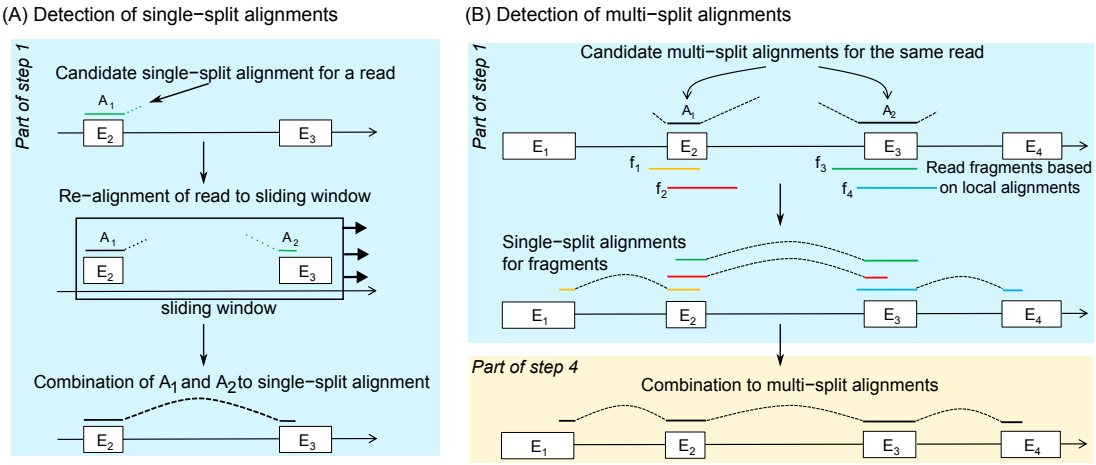

Figure 3.3: (A) Detection of single-split alignments as part of step 1 of ContextMap. First, reads are aligned to the genome and candidate split alignments (A_1) are identified. Second, reads with candidate split alignments are re-aligned within a window around the initial alignment to determine a completing alignment (A_2). The use of smaller seed lengths than in the initial alignment allows recovering completing alignments shorter than the seed length used for the initial alignment. Finally, the alignments are combined to a complete split alignment. (B) Detection of multi-split alignments. For every candidate multi-split alignment, ContextMap creates two fragments of the respective read sequence (i.e. f_1 and f_2 for A_1 and f_3 and f_4 for A_2). Subsequently, single-split alignments are detected for these fragments. Finally, overlaps of single-split alignments are combined to obtain a complete multi-split alignment after first identifying the best splice site for each split alignment as part of the resolution of overlapping splice sites in step 4.

3.2.5 Detection of complete multi-split alignments

The detection of multi-split alignments, i.e. alignments crossing more than one exon-exon junction, is a novel feature of ContextMap 2. It is based on local alignment options of recently developed alignment programs such as Bowtie 2 or BWA. Essentially, the local alignments are used to fragment the reads into smaller segments for which single-split alignments are then determined (see Figure 3.3 B). In contrast to other approaches that fragment all reads into smaller equal-sized segments, only reads for which a local alignment was determined, i.e. candidate multi-split alignments, are fragmented by ContextMap 2.

For this purpose, candidate multi-split alignments (=local alignments with suffix and prefix of the read not aligned) to the same genomic region are collected using the same sliding window approach used for the single-split alignment detection. In fact, ContextMap uses a single run of the sliding window approach to process single- as well as multi-split alignments.

For each candidate multi-split alignment in the current sliding window, two fragments of the read sequence are generated. If read $r = r_1 \dots r_l$ (l = read length) has been aligned at positions $r_i \dots r_j$, the first fragment consists of the subsequence $f_1 = r_{i-e} \dots r_{i-1} r_i \dots r_j$, where e is the predefined minimum exon size (default 20 bps). If the unaligned prefix ($r_1 \dots r_{i-1}$) of the read is smaller than the minimum exon size e, $f_1 = r_1 \dots r_j$. Similarly, the second fragment is defined as $f_2 = r_i \dots r_j r_{j+1} \dots r_{j+e}$. If the unaligned suffix of the read ($r_{j+1} \dots r_l$) is shorter than e, $f_2 = r_i \dots r_l$.

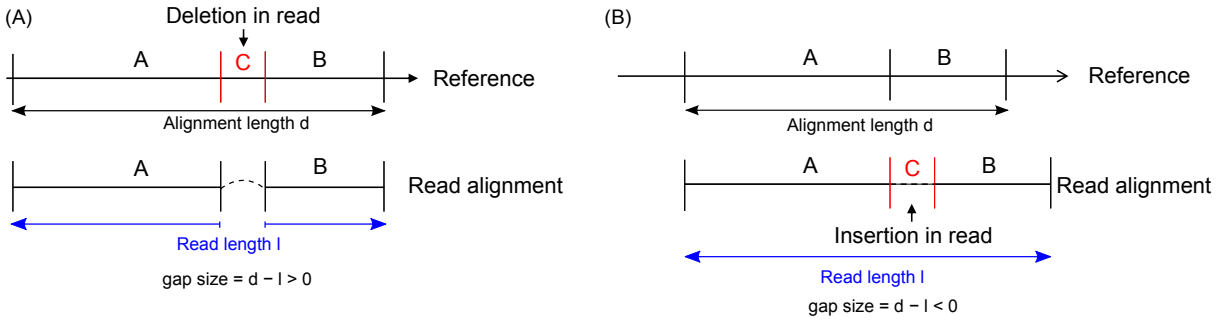

Figure 3.4: (A) Example of a read with a deletion compared to the reference sequence. In this case, the alignment length d is larger than the read length l and the gap size is positive. (B) Example of a read with an insertion compared to the reference sequence. Here, the alignment length d on the reference sequence is smaller than the read length l and the gap size is negative.

The original local alignment then provides candidate split alignments for f_1 and f_2. The completing alignments to these candidate split alignments are found within the sliding window as described in the previous section. This results in single-split alignments for the fragments, which are added to the list of initial alignments determined in step 1 of ContextMap 2 and extended to all valid single-split alignments of the fragments in step 3.

The complete multi-split alignment of the whole read is determined in step 4 by merging overlaps of the single-split alignments for fragments of the same read after the resolution of pairwise overlapping splice sites. Thus, the precise location of the splice sites is first determined for the single-split alignments of the fragments before combining them to the complete multi-split alignments.

3.2.6 Detection of indels

Essentially, the prediction of reads containing a deletion to the reference is the same as detecting spliced reads with a very small intron size (see Figure 3.4 A). Similarly, a read containing an insertion to the reference can be considered as a special case of a spliced read spanning an intron with negative length (see Figure 3.4 B). Thus, detection of deletions and insertions could be incorporated seamlessly into the single- and multi-split alignment detection procedure of ContextMap 2 by allowing both small and negative intron lengths, respectively. Conveniently, this also allows the mapping of reads containing both indels and splice sites by finding the corresponding multi-split alignment.

The distinction between indels and splice sites is only applied when preparing the output at the very end of the ContextMap 2 run. At this point, the gap size is determined for each split position in a single- or multi-split alignment (see Figure 3.4). The gap size is defined as $d - l$, where d is the alignment length on the reference genome and l is the read length. If the gap size is negative and its absolute value at most a user defined maximum insertion size (default = 10 bps), this split position is classified as an insertion. If the gap size is between 1 and a user defined maximum deletion size (default = 10 bps), it is classified as a deletion. If the gap size is between a user defined minimum intron size (default = 50

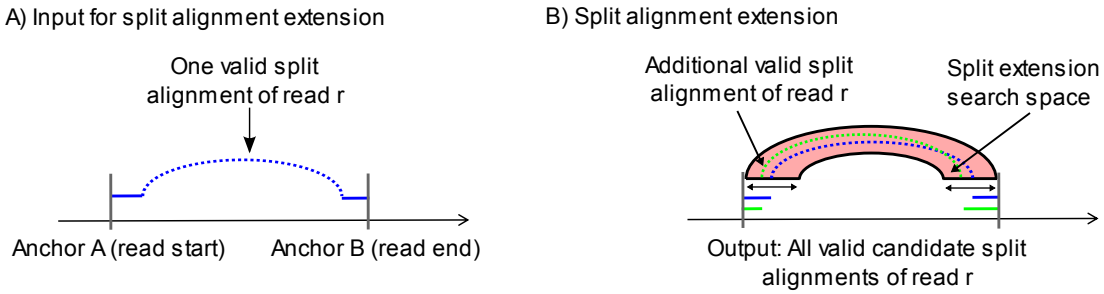

Figure 3.5: (A) The single-split detection method in step 1 provides only one valid split alignment with the minimum number of mismatches for each possible combination of alignment start and end. (B) By shifting the position of the splice site, alternative split alignments with the same number of mismatches or a few more (indicated by the maximum mismatch difference parameter) are determined. The range of the search space is defined by the read length.

bps) and a user defined maximum intron size (default = 300,000 bps), the split is classified as an intron. Split alignments with gap sizes that do not fall into these ranges are not determined when detecting single- and multi-split alignments.

3.2.7 Alignment extension for split alignments

Similarly to the extension of full and partial read alignments, additional alignments are determined for split alignments. The input for this extension is the set of single-split alignments (including single-split alignments of fragments of multi-split candidates) determined in step 1. Each single-split alignment consists of a combination of continuous alignments beginning at the read start and the read end separated by the predicted intron (Figure 3.5 A). The ends of this intron represent the predicted splice sites. Here, step 1 determines only one split alignment with the minimum number of mismatches for each combination of alignment start (anchor A in Figure 3.5 A) and alignment end (anchor B in Figure 3.5 A).

However, other split alignments may be possible for the same combination of alignment start and end with the same number of mismatches or only a few more (the difference in mismatches allowed is provided by the user using the maximum mismatch difference [*mmdiff*] parameter). These alignments are determined by shifting the position of the splice sites as shown in Figure 3.5 B.

Furthermore, single-split alignments (excluding single-split alignments of fragments of multi-split candidates) are checked for overlaps with split alignments of other reads indicating an additional split within the continuously aligned regions. If an overlap is found, the single-split alignment is extended to a multi-split alignment. Here, only single-split alignments of the whole read are extended to two-split alignments, but not single-split alignments of read fragments obtained for candidate multi-split alignments (see Detection of multi-split alignments).

In all cases, only alignments are used that do not exceed the maximum mismatch criterion. At the end of this step, several different alignments have been created for each

read, resulting in multiple alignments both within and between contexts. Resolution of these multiple alignments is then performed in step 4 and 5 of ContextMap 2.

3.2.8 Resolution of overlapping splice sites

This is part of step 4 of ContextMap 2, in which multiple alignments are resolved within contexts. Here, splice sites are eliminated which are very close to each other and suggested by alternative split alignments of the same read with the same alignment start and end but different position of the splice site. Elimination is based on the evidence for different splice sites provided by all reads. Although it might appear counterintuitive that alternative split alignments are first created in step 3 of ContextMap 2 and then some are deleted again, this guarantees that all reads with a valid split alignment using this splice site are included in calculating the evidence score.

Two splice sites $(s_{1,1}, s_{1,2})$ and $(s_{2,1}, s_{2,2})$ are considered overlapping if both $|s_{1,1} - s_{2,1}|$ and $|s_{1,2} - s_{2,2}|$ are smaller than the maximum read length. Here, $s_{i,1}$ denotes the genome position of the end of the first exon and $s_{i,2}$ the start of the second exon. While in the original ContextMap implementation only one splice site from a set of overlapping splice sites was used, ContextMap 2 retains the three splice sites from each set with the highest evidence score (see Figure 3.6). This allows the detection of alternative 3' or 5' splice sites.

ContextMap 2 uses a similar evidence score as in the original ContextMap version. The major difference involves the treatment of gene annotation (if provided) and known splice signals. If at least one splice site within the set of overlapping splice sites corresponds to an annotated exon-exon junction or shows a known splice signal, all other splice sites not corresponding to a known exon-exon junction or having no known splice signal are discarded. The evidence score used for evaluating the remaining splice sites is calculated as follows. Let n_i be the number of reads (full, split or partial) with i mismatches supporting the splice site pair and m the maximum number of mismatches allowed. Then the evidence score is defined as:

$$evidence = \sum_{i=0}^{m} (w^i \cdot n_i) \tag{3.1}$$

Here, w is a value < 1 (default $w = 0.3$). Thus, the score is the weighted sum of the number of reads with the weight decreasing exponentially with the number of mismatches.

For each set of pairwise overlapping splice sites, the three splice sites with the highest evidence scores are selected. Split read alignments containing the discarded splice sites are discarded. If more than one split alignment remains for a read to any of the remaining three splice sites, the split alignment to the splice site with highest evidence score is retained and all others discarded.

3.2.9 Resolution of multiple read alignments

ContextMap 2 resolves multiple read alignments first within each context in step 4 and subsequently between the contexts in step 5. For this purpose, a support score is calculated

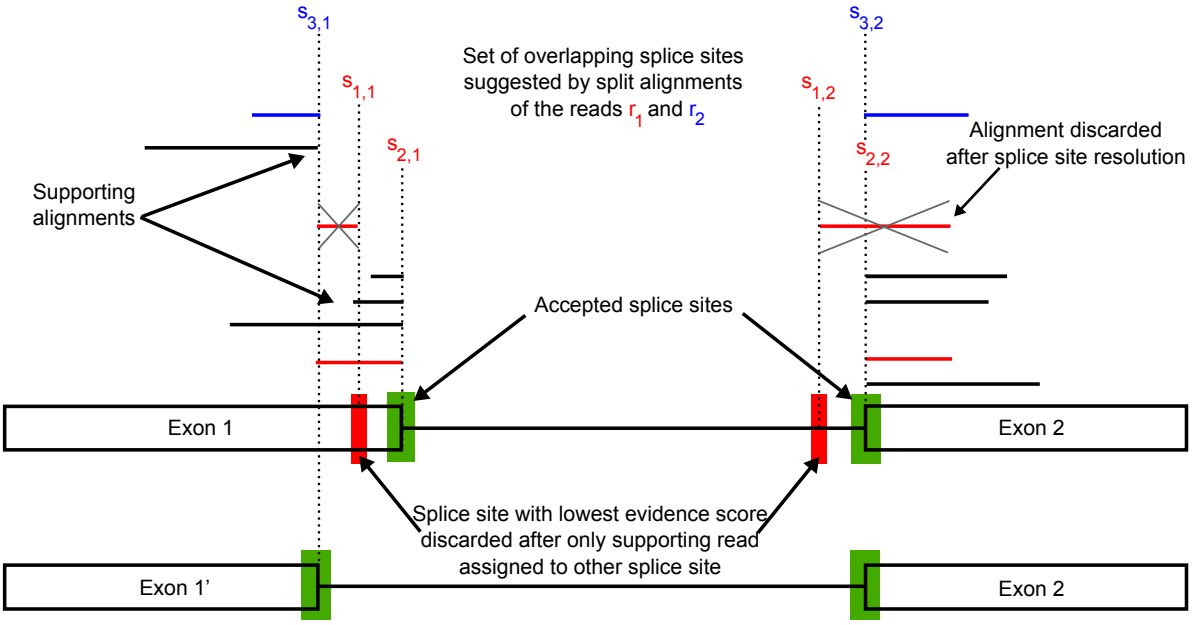

Figure 3.6: Resolution of overlapping splice sites. Split alignments of reads r_1 (red) and r_2 (blue) suggest a set of three overlapping splice sites $\{(s_{1,1}, s_{1,2}), (s_{2,1}, s_{2,2}), (s_{3,1}, s_{3,2})\}$. Here, $(s_{1,1}, s_{1,2})$ and $(s_{2,1}, s_{2,2})$ are indicated by alternative split alignments of the same read r_1. Assuming that all shown alignments have zero mismatches, this results in the following evidence scores for the three splice sites: 1, 4 and 2. Although all three splice sites would be retained at first, the only supporting read for the splice site $(s_{1,1}, s_{1,2})$, i.e. r_1, is assigned to the splice site $(s_{2,1}, s_{2,2})$ with higher evidence score. As a consequence, the splice site $(s_{1,1}, s_{1,2})$ is discarded as it is no longer supported by any reads.

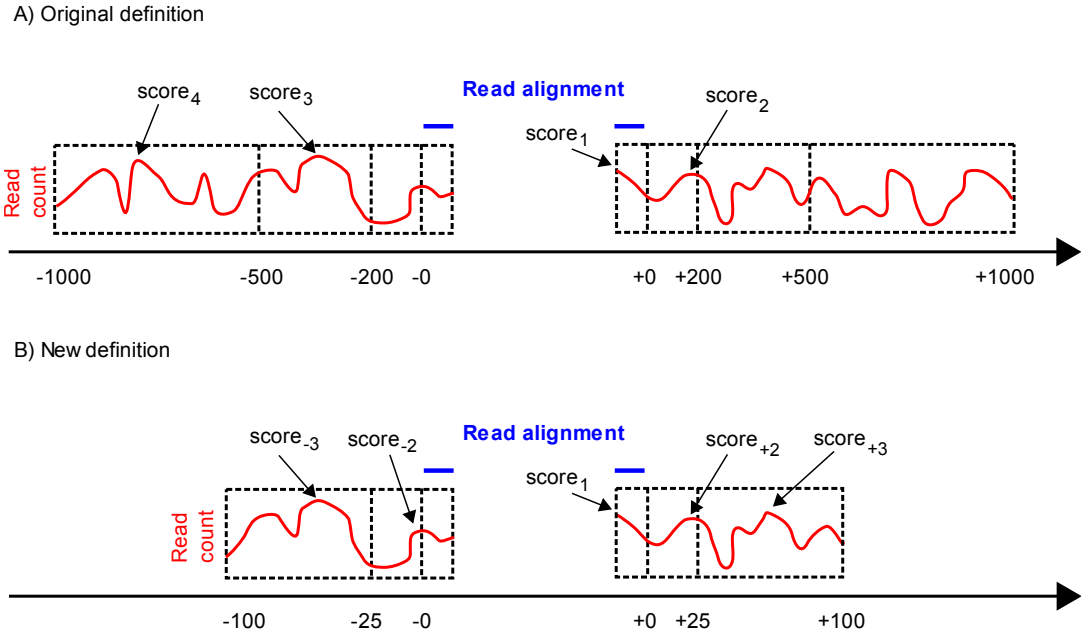

Figure 3.7: Illustration of the old (A) and new (B) support score definition. The definition for full alignments is the same, with the exception that the region to which the read is aligned is continuous.

for each alignment based on the number of reads aligned to the same genomic region. In the original ContextMap implementation, the score was defined as (see Figure 3.7 A)

$$support = \sum_{i=1}^{4} 2^{4-i} \cdot \lfloor \ln(score_i) \rfloor. \tag{3.2}$$

Here, $score_1$ was defined as the maximum number of reads mapping to any position within the region the read is aligned. $score_2$ was the maximum in a window of 200 nt either upstream of the read start or downstream of the read end. $score_3$ was the maximum > 200 but ≤ 500 nt from read start or end. Finally, $score_4$ was the maximum > 500 but ≤ 1000 nt from read start or end.

To better distinguish between different alignments for a read that are identical on one but not the other side of the read, the support score in ContextMap 2 was modified such that maximum read counts on both sides of the of read alignment are included separately in the score. Furthermore, we reduced the number of considered windows around the read alignment and their respective sizes since the considered region was much larger than an average exon (Figure 3.7 B). The new support score is then defined as

$$\begin{aligned} support = {} & 2^3 \cdot \lfloor \ln(score_1) \rfloor \\ & + \sum_{i=2}^{3} 2^{4-i} \cdot \left(\lfloor \ln(score_{-i}) \rfloor + \lfloor \ln(score_i) \rfloor \right). \end{aligned} \tag{3.3}$$

Here, $score_{-i}$ and $score_i$ were defined as the maximum read counts in the corresponding intervals upstream and downstream of the read alignment, respectively. For reads with multiple alignments, the count was 1/(#multiple alignments of read within context) for step 4 and 1/(#multiple alignments of read between contexts) for step 5 of ContextMap 2.

Thus, if many reads are aligned to the same region, indicating that this region is actually expressed, the score of the alignment is high. If only few other reads are aligned to the same region as the read, the score of the alignment is low. Among several alternative alignments for the same read within each context, the one with the largest support score is then chosen. Finally, reads aligned to several different contexts are resolved in the same way in step 5 after recalculating support scores based on the read alignments chosen for each context.

3.3 Results and Discussion

3.3.1 Data sets and methods for evaluation

Evaluation of ContextMap 2 was performed on simulated and real data previously used by the RGASP consortium for the systematic evaluation of RNA-seq mapping programs (Engstrom et al. [2013])(see Table 3.1 for a summary).

The simulated data was generated using the simulation program BEERS, which is provided with the RUM pipeline (Grant et al. [2011]). Two data sets were simulated, each containing 80 million 76-nucleotide paired-end reads (= 40 million read pairs). The second data set is more challenging than the first as higher rates of substitution errors, indel polymorphisms and reads from unannotated isoforms were simulated.

The real data consists of RNA-seq data of the human K562 cell line (whole cell, cytoplasmic and nuclear fraction) from the ENCODE project (ENCODE Project Consortium [2012]) (2 replicates each, resulting in 6 samples). Each sample consisted of ∼200 million 76-nucleotide paired-end reads (∼100 million read pairs).

We compared ContextMap 2 against the best performing RNA-seq mapping approaches identified in the RGASP study. These included MapSplice (Wang et al. [2010]), STAR (Dobin et al. [2013]), and GSNAP (Wu and Nacu [2010]). We also included TopHat (Trapnell et al. [2009]) (denoted as TopHat1 in the following) and Tophat2 (Kim et al. [2013]) as these are most commonly used RNA-seq mapping programs. Mapping results of these programs on the used data sets as well as evaluation scripts were provided by the authors of the RGASP study (`https://github.com/RGASP-consortium/`). For all programs, we evaluated the performance without and with an annotation (indicated by "ann"). For STAR, we evaluated both the 1- and 2-pass version. In the 2-pass version of STAR, splice junctions detected in the first run (1-pass) are taken as an input for a second run to improve mapping.

We applied the same evaluation scripts to evaluate ContextMap 2 mapping runs using Bowtie (version 0.12.7), Bowtie 2 (version 2.1.0), or BWA (version 0.7.8) as internal short read alignment programs. Additionally, we evaluated the performance of ContextMap 2

Dataset	ID	# Sequenced fragments	# Reads
Simulation 1	NA	40000000	80000000
Simulation 2	NA	40000000	80000000
K562 whole cell replicate 1	LID16627	113588758	227177516
K562 whole cell replicate 2	LID16628	119053315	238106630
K562 cytoplasmic fraction replicate 1	LID8465	124826068	249652136
K562 cytoplasmic fraction replicate 2	LID8466	88445339	176890678
K562 nuclear fraction replicate 1	LID8556	117113622	234227244
K562 nuclear fraction replicate 2	LID8557	105769104	211538208

Table 3.1: Data sets used for evaluation and number of sequenced fragments and reads for each data set.

using BWA and an annotation. Here, the annotation is only used for scoring splice junctions when resolving overlapping splice sites (see Methods). As for the RGASP evaluation, the annotation was taken from Ensembl version 62. Although we also performed evaluation of the original ContextMap implementation, we did not include it in the thesis as it performed worse in all evaluated metrics than ContextMap 2.

For runtime comparison, we applied all RNA-seq mapping programs with the same parameter settings as described in the RGASP study. The only exception was MapSplice. In this case, an internal version of MapSplice was used in the RGASP study, which is not available for download. Most likely it was an unfinished predecessor of MapSplice 2, which has since been made publicly available (`http://www.netlab.uky.edu/p/bioinfo/MapSplice2`). It was not the published MapSplice 1.x version as options were used (e.g. detection of indels with length > 3) that this version does not support. We thus included an evaluation of MapSplice 2 in this work by applying it to all data sets using default parameters. Since MapSplice 2 uses the annotation only to detect fusion junctions between different genes, which was not simulated in the RGASP data sets, MapSplice 2 was only applied without annotation.

3.3.2 Alignment yield

As a first metric, we evaluated the fraction of mapped reads for both simulated data sets (see Supplementary Table A.1). This showed significant differences between RNA-seq mapping programs with GSNAP having the highest mapping rates (∼99% and ∼98% of the reads for simulation 1 and 2) and TopHat1/2 and ContextMap 2 having lowest mapping rates (89-96% of reads mapped in simulation 1 and 78-88% in simulation 2).

When investigating the fraction of reads mapped either perfectly, part correctly or with no base correct (Figure 3.8 and Supplementary Table A.1), it became apparent that mapping rates alone are not meaningful for comparing the performance of algorithms. Despite GSNAP's high overall mapping rate, the fraction of perfectly mapped reads was

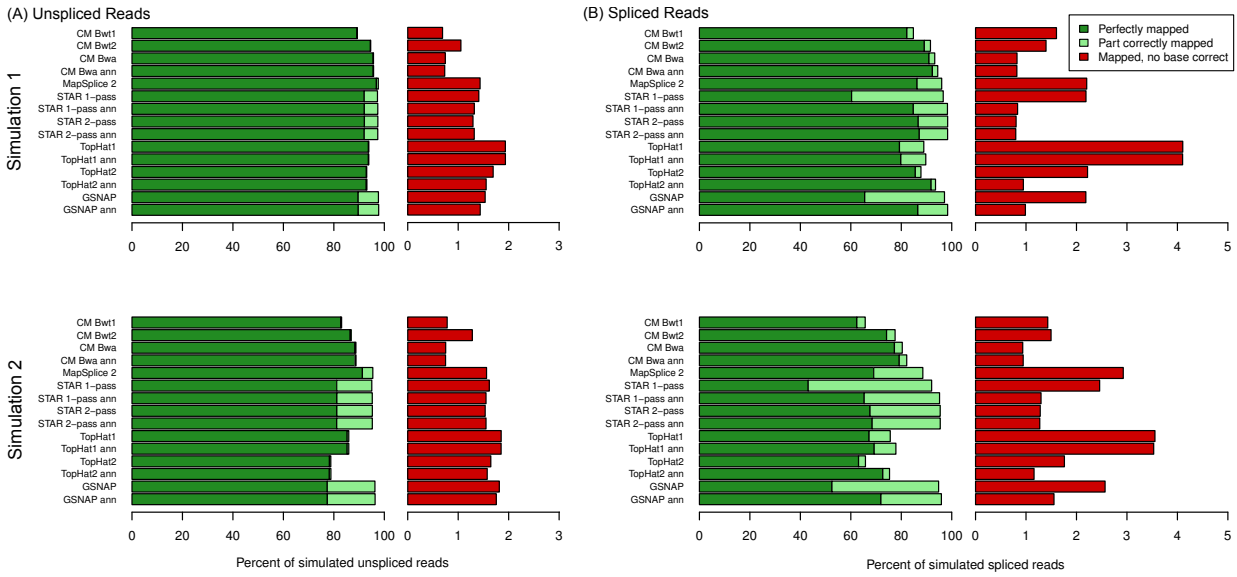

Figure 3.8: Fraction of perfectly mapped, part correctly mapped and incorrectly mapped reads for simulated unspliced and spliced reads of simulation 1 and 2, respectively. "CM Bwt1", "CM Bwt2" , "CM Bwa" denote ContextMap 2 used with Bowtie, Bowtie 2, and BWA as underlying alignment program, respectively. If a gene annotation was provided, "ann" was added to the name of the respective program.

only 89% and 76% of reads of simulation 1 and 2, respectively. In contrast, ContextMap 2 using BWA mapped almost 95% and 87% of reads perfectly, which was better than for all other evaluated methods except MapSplice 2. Consistently, both the fraction of part correctly mapped reads and reads with no base mapped correctly were lower than for all other methods (see also Figure 3.8). Thus, the higher mapping rates of other programs came at the cost of higher error rates.

To investigate whether performance differed between unspliced and spliced reads, mapping rates were also calculated separately for both types of reads (Figure 3.8 and Supplementary Tables A.2 and A.3). Indeed, the evaluated programs differed considerably in performance between spliced and unspliced reads but not in any consistent fashion. For ContextMap 2 using Bowtie, MapSplice, STAR 1-pass, TopHat1 and GSNAP (and TopHat2 on simulation 1), the fraction of reads mapped completely wrong increased by more than 0.5 percentage points for spliced reads compared to unspliced reads. In contrast, this fraction did increase less for ContextMap 2 using Bowtie 2 or BWA (and TopHat2 on simulation 2) and even decreased for the remaining tools. In all cases, however, the number of part correctly mapped reads increased for spliced reads, but least for ContextMap 2 and TopHat2. This was likely due to a part of the read on one side of the splice junction not being mapped correctly or not at all (e.g. in case of STAR, which can also output clipped alignments). In particular for STAR and GSNAP, this lead to 10-50% part correctly mapped reads.

In summary, these results show that ContextMap 2 using BWA had the lowest rate of incorrectly mapped reads among all evaluated programs. Furthermore, it mapped more

reads perfectly than any of the other programs except MapSplice 2. However, MapSplice 2 had ∼2-fold higher rates of incorrectly mapped reads.

Interestingly, we observed that the choice of the underlying alignment program had a significant influence on the performance in RNA-seq mapping. Both rates of perfectly and incorrectly mapped reads are improved significantly when using BWA within ContextMap 2 instead of either Bowtie or Bowtie 2. The reduced number of perfectly mapped reads for Bowtie is mostly due to its lower overall recall (Lindner and Friedel [2012]) and the fact that it does not determine local alignments and thus does not support the detection of multi-split read alignments and indels within ContextMap 2. The higher number of incorrectly mapped spliced reads results from spliced reads for which the seed at the read start cannot be aligned at the correct position, e.g. because the splice site in the read is closer to the read start than the seed length, but the seed can be aligned to a wrong position. In this case, no backward alignment is performed for the read in order to reduce runtime and only the incorrect alignments are further analyzed.

The lower mapping quality using Bowtie 2 compared to BWA resulted from the fact that – in contrast to Bowtie and BWA – Bowtie 2 has a dramatically increased runtime if the maximum number of valid alignments reported per read (`-k` option) is set to even moderately high values. Thus, per default we used a relatively low value of $k = 3$. Using a value of $k = 10$ resulted in comparable mapping quality to ContextMap 2 with BWA (see Supplementary Tables A.1-A.3) but runtime increased by at least 8 h compared to BWA or Bowtie 2 with $k = 3$ (see Table 3.3).

3.3.3 Alignment yield on real-life RNA-seq data

Consistent with evaluation results on simulated data, alignment yield of ContextMap 2 was lower on all samples for the K562 cell line than for MapSplice 2, STAR or GSNAP, but similar or slightly higher than for TopHat1/2 (Figure 3.9). This was only partly due to the relatively small number of mismatches (=4) allowed per default in ContextMap 2. Nevertheless, the ranking of algorithms with regard to the number of mapped reads is quite similar to the ranking on the simulated data. Thus, if we also extrapolate the results on perfectly and incorrectly mapped reads from the simulation to the real-life data, this would suggest that the difference in mapped reads between ContextMap 2 and most other mapping programs are to a large extent due to incorrect mappings identified by the other programs.

3.3.4 Spliced alignment

Since performance on spliced reads showed the largest differences among the mapping approaches, these were analyzed in more detail (Figure 3.10 A and Supplementary Figure A.2). For this purpose, splice recall and false discovery rate (FDR) were calculated as in

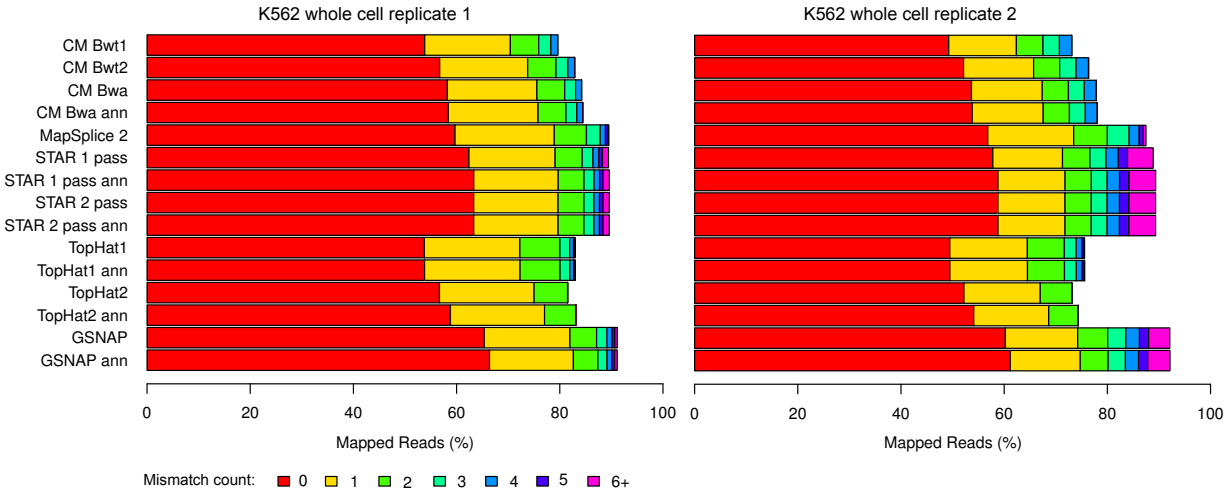

Figure 3.9: Percentage of mapped reads and mismatch distribution for the mapped reads for both replicates of the K562 whole cell RNA-seq samples. Results for all real-life samples are shown in Supplementary Figure A.1.

the original RGASP study. Here, splice recall is defined as

$$\begin{aligned} \text{recall} &= \frac{\#\text{true positive splices}}{\#\text{simulated splices}} \\ &= \frac{\#\text{true positive splices}}{\#\text{true pos. splices} + \#\text{false neg. splices}}. \end{aligned} \tag{3.4}$$

In this case, a splice is defined as one junction in one particular read. Thus, if a simulated junction within a read is recovered by the alignment for this read, it is considered a true positive splice. If it is not recovered, it is a false negative splice. If the alignment contains a junction that was not simulated for this read, it is considered a false positive splice. FDR is then defined as 1 - precision, with

$$\begin{aligned} \text{precision} &= \frac{\#\text{true positive splices}}{\#\text{predicted splices}} \\ &= \frac{\#\text{true positive splices}}{\#\text{true pos. splices} + \#\text{false pos. splices}}. \end{aligned} \tag{3.5}$$

For the real data, recall and FDR could not be calculated as the correct mapping was not known. Instead, the fraction of reads mapping to an annotated splice junction (=: frequency of annotated splices) was compared to the fraction of reads mapping to a novel splice junction (=: frequency of novel splices).

Consistent with the evaluation of alignment yield, this analysis showed that ContextMap 2 combined low FDR with high recall. Again the combination with BWA performed best. Although some of the other mapping programs showed higher recall, this was always accompanied by significantly higher FDR. Generally, the increase in recall compared to ContextMap 2 was only modest with the exception of annotation-based GSNAP on simulation 2.

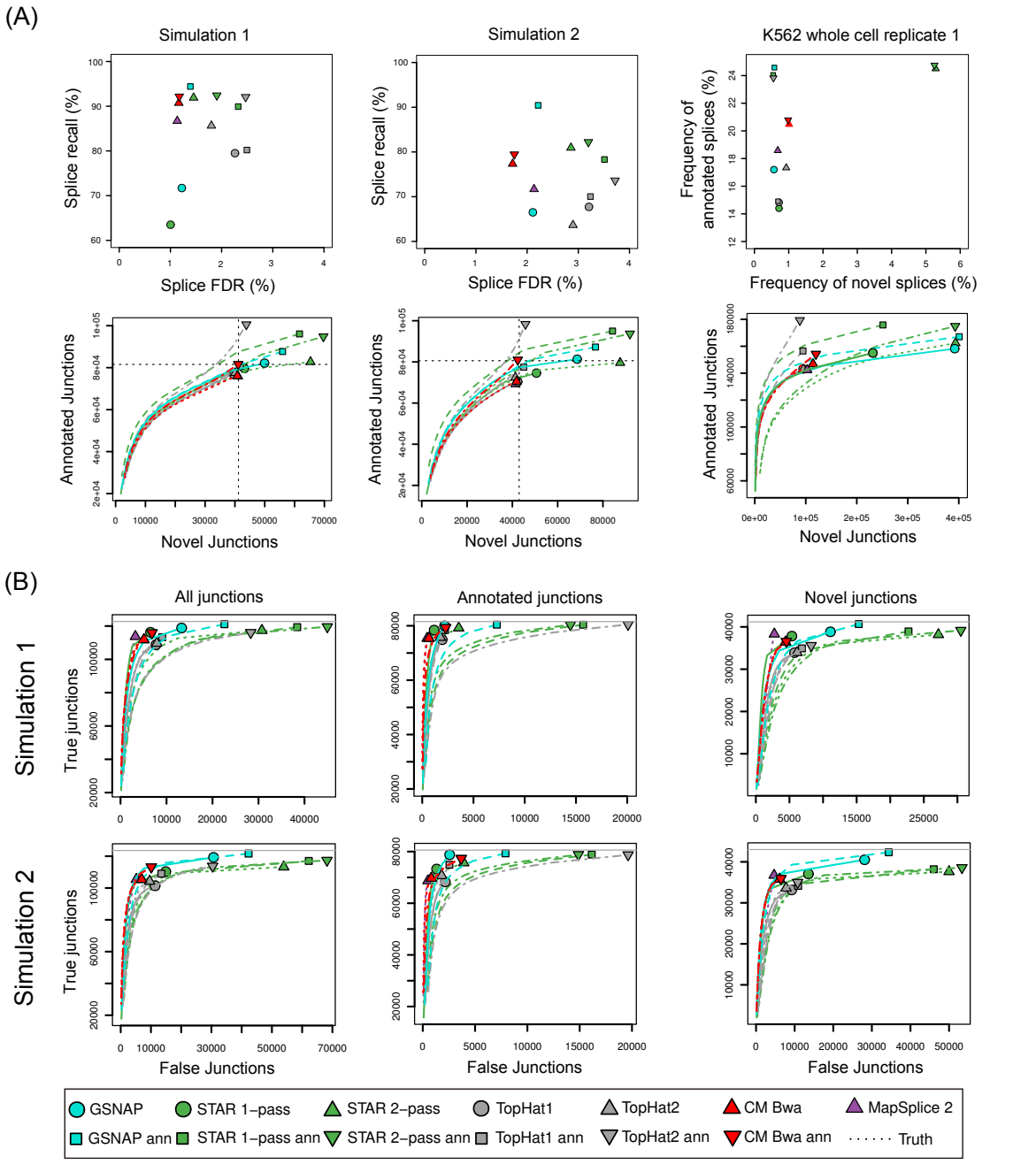

Figure 3.10: (A) Comparison of splice recall (y-axis) versus splice false discovery rate (FDR=1-precision, x-axis) on simulation 1 and 2 (see equations 3.4 and 3.5 for definitions). For the human data sets, the frequency of predicted novel splices was compared to the frequency of annotated splices for the Ensembl annotation (see text for definitions, Supplementary Figure 5 for results for all real-life data sets). Furthermore, the number of identified annotated and novel junctions was evaluated (see Supplementary Figure 6 for results for all data sets). To obtain receiver operation characteristic (ROC)-like curves, numbers were also calculated at increasing thresholds on the number of supporting reads for each junction. (B) Number of correctly predicted (true) and incorrectly (false) junctions were compared for all junctions and annotated and novel junctions separately. In contrast to the RGASP evaluation, we also included junctions covered by only 1 read. ROC-like curves were calculated as in A. (A-B) For ContextMap 2 only results using BWA are shown, results for Bowtie and Bowtie 2 can be found in Supplementary Figures A.2-A.3 (for A) and A.4 (for B).

The analysis of known and novel splices identified in the real data set showed that ContextMap 2 mapped reads to novel splices with similar frequency as most other programs except STAR 2-pass (Figure 3.10 A and Supplementary Figure A.2). In contrast, reads were mapped to known splice junctions less frequently compared to most programs using an annotation and more frequently than most programs without annotation. Unfortunately, these results are difficult to interpret as alignments to novel junctions are not necessarily wrong and alignments to annotated junctions not necessarily right.

To address this problem we also compared the number of novel and annotated junctions predicted by all methods between the simulations and the real data sets (Figure 3.10 A and Supplementary Figure A.3). Here, the same junction (in terms of the genomic coordinates) identified for several reads was counted only once. This consistently showed that ContextMap 2 predicted significantly fewer novel junctions than STAR and GSNAP (>50% less). Here, ContextMap 2 using BWA or Bowtie 2 and MapSplice showed quite similar performance, whereas annotation-based ContextMap 2 using BWA and, in particular, annotation-based TopHat2 predicted significantly more annotated junctions. Interestingly, annotation-based ContextMap 2 identified almost precisely the correct number of annotated and novel junctions for both simulations. The high similarity of the results between simulation and real data indicates that recall and FDR from the simulations can again be extrapolated to the real data sets. This would suggest that ContextMap 2 using BWA (both with and without an annotation) correctly identifies more reads with known junctions than programs not using an annotation but is less biased towards annotated junctions than other programs using an annotation.

This conclusion is also supported by the comparison of the number of correctly predicted junctions to false junctions (Figure 3.10 B and Supplementary Figure A.4). This again shows that ContextMap 2 (in particular when using BWA) predicts much fewer false junctions than approaches using an annotation, while missing relatively few of the true junctions. For novel junctions ContextMap 2 is only outperformed in terms of recall and FDR by MapSplice 2, but the difference in performance is relatively small. For annotated junctions, the ContextMap 2 version without annotation performs almost as good as MapSplice 2, which has the lowest FDR, whereas the version using the annotation has a significantly higher recall but also predicts more false junctions. Again, this highlights the problem in using an annotation, which might bias the results towards known junctions. Nevertheless, ContextMap 2 appears to be less biased by the annotation than STAR, GSNAP or Tophat2.

3.3.5 Detection of multi-junction reads

Since ContextMap 2 now also supports mapping of reads crossing multiple junctions, we calculated recall and precision separately for reads containing different number of junctions (Table 3.2 and Supplementary Table A.4). For this purpose, a read was considered a true positive if all junctions in this read were identified correctly and no additional junctions were predicted. If a different number of junctions were predicted than correct, it was considered a false negative for this junction number and a false positive for the junction

number predicted by the alignment. If the correct number of junctions were predicted for the read, but some of the junctions were wrong, it was considered a false positive for this junction number. To evaluate the trade-off between precision and recall, we calculated F-measure values defined as

$$F - measure = 2 \cdot \frac{precision \cdot recall}{precision + recall}. \tag{3.6}$$

This showed that ContextMap 2 using BWA (both with and without annotation) outperformed all other programs on reads containing only one junction except STAR 2-pass and annotation-based GSNAP. This includes the vast majority of all spliced reads. Here, only annotation-based GSNAP performed significantly better, at least on simulation 2. In general, F-measure decreased with increasing number of junctions for all programs, mostly due to lower recall values. Precision generally remained above 90%. For reads with two junctions, ContextMap 2 with BWA was still only outperformed by STAR 2-pass, annotation-based GSNAP and now also annotation-based Tophat2, but the difference in recall to these programs increased.

For three junctions, however, recall and thus F-measure of ContextMap 2 using BWA or Bowtie 2 dropped dramatically, such that only MapSplice 2, STAR 1-pass and GSNAP (both without annotation) performed worse. Since Bowtie does not perform local alignment, ContextMap 2 using Bowtie cannot identify multi-split alignments and therefore had zero recall on three-junction reads. A small number of two-junction reads were mapped as single-split alignments are extended to multi-split alignments in step 3 of ContextMap 2 if they overlap an additional splice site.

Since ContextMap 2 by default only determines multi-split alignments for which internal exons are at least 20 nt long (= minimum exon size e), we repeated the analysis only for multi-junction reads fulfilling this condition. The results of this analysis are shown in the last two columns of Table 3.2 and Supplementary Table A.4. Here, ContextMap 2 using BWA showed a significant improvement, resulting in similar or better performance for two-junction reads than all programs except TopHat2 on simulation 1 and annotation-based GSNAP. For three-junction reads, recall of ContextMap 2 was almost doubled, whereas for other programs improvements were less pronounced and recall of MapSplice 2 actually decreased to $< 2\%$. In addition, ContextMap 2 using BWA generally showed a significantly higher precision than the programs with particularly high recall.

3.3.6 Indel accuracy

Precision, recall and F-measure values were also calculated separately for reads containing insertions and deletions (see Figure 3.11, Supplementary Figure A.5, and Supplementary Tables A.5 and A.6). These results show that ContextMap 2 using BWA outperforms all other approaches on both insertions and deletions except for GSNAP (both with and without annotation) and annotation-based TopHat2. Furthermore, the latter programs only performed comparably well to ContextMap 2 on reads with small indel size (1-4, depending on the method). In almost all cases, precision of ContextMap 2 using BWA was

Program	Number of junctions spanned				
	1 (13808336)	2 (598297)	3 (11781)	2* (548382)	3* (6908)
CM Bwt1	91.47	14.24	-	15.16	-
CM Bwt2	94.03	78.47	50.21	82.37	72.66
CM Bwa	95.03	82.73	53.33	86.67	76.46
CM Bwa ann	95.74	84.65	53.9	88.47	76.79
MapSplice 2	92.42	79.18	27.27	80.65	3.44
STAR 1-pass	77.63	30.91	5.01	31.91	1.49
STAR 1-pass ann	93.55	81.65	75.71	82.57	82.17
STAR 2-pass	95.0	85.55	82.07	86.59	87.29
STAR 2-pass ann	95.07	86.28	82.55	87.02	86.49
TopHat1	87.83	77.51	63.57	80.42	75.56
TopHat1 ann	88.02	78.99	68.13	81.06	76.1
TopHat2	91.71	87.0	76.92	89.66	88.51
TopHat2 ann	94.84	90.79	85.92	92.05	90.35
GSNAP	83.13	43.45	18.52	42.35	12.29
GSNAP ann	96.47	88.59	79.51	89.86	84.67

Table 3.2: F-measure [in %] for spliced reads with different number of spanned junctions (simulation 1, recall and precision values for both simulations can be found in Supplementary Table A.4). Columns marked with an asterisk show results only for reads for which all exons except the first and last exon had length $\geq$ 20 nt. For this evaluation, read alignments were only considered a true positive if all simulated splice junctions in the read were recovered and no additional splice junctions were identified. Indels were ignored for this purpose.

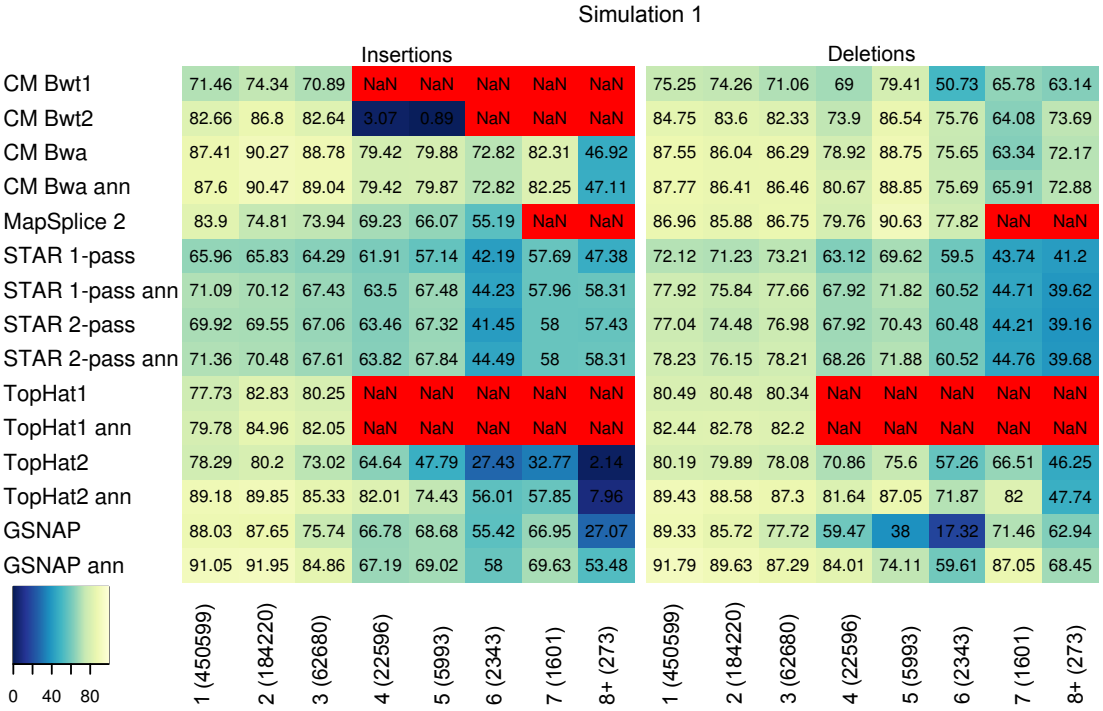

Figure 3.11: F-Measure [in %] for insertion and deletions identified by all programs on simulation 1. NaN indicates that no insertion or deletion of that size was identified. Insertion and deletion size are shown below the column of the heatmap. The numbers in parentheses indicate the number of simulated reads for each insertion or deletion size. Results for simulation 2 are shown in Supplementary Figure A.5. Recall and precision values are listed in Supplementary Tables A.5 and A.6.

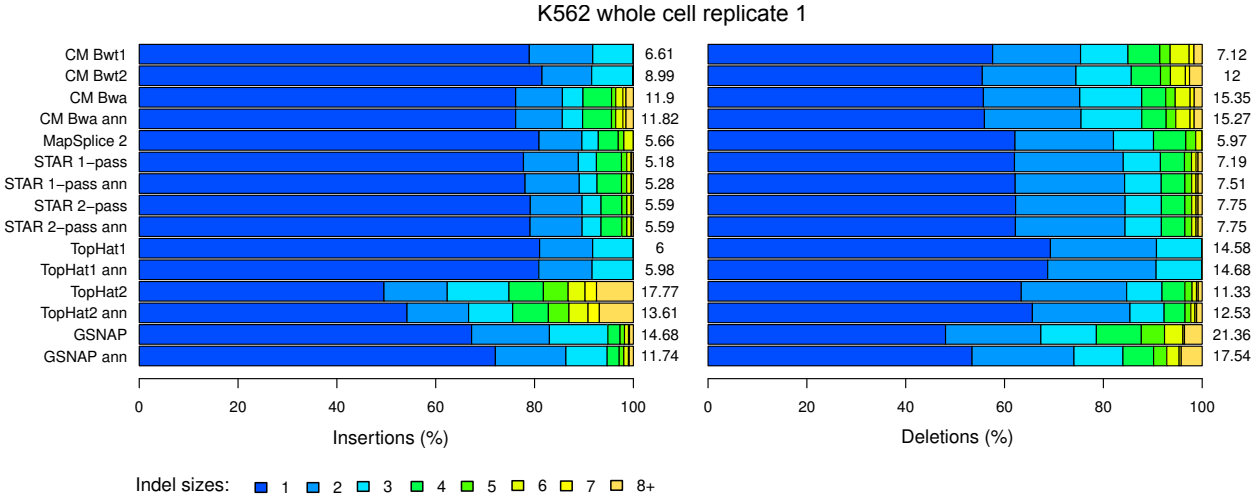

Figure 3.12: Fraction of mapped reads with different indel sizes among all reads with indels for the first replicate of the K562 whole cell sample. Numbers next to the barplots indicate the number of mapped reads with indels divided by 10^5 (i.e. number of reads per 100,000). Results for all samples are shown in Supplementary Figures 9 and 10.

above 90% and higher than for the best competing programs. Similar to multi-junction reads, the integration of Bowtie or Bowtie 2 in ContextMap 2 resulted in worse performance on indels than for BWA, in particular for longer insertions.

Numbers of detected indels and indel length were also evaluated on the real-life sequencing data (Figure 3.12 and Supplementary Figures A.6 and A.7). Consistent with their higher recall on the simulations, ContextMap 2 using BWA, TopHat2 and GSNAP mapped at least twice as many reads with insertions than the other programs. Interestingly, numbers of mapped insertions generally decreased significantly for TopHat2 and GSNAP when not using an annotation, while there were hardly changed for ContextMap 2 using BWA. Since simulation results showed higher precision for annotation-based GSNAP and TopHat2 compared to the runs without annotation but not lower recall, this indicates that the lost mappings were largely false positive results. Furthermore, even compared to annotation-based GSNAP and TopHat2, precision of ContextMap 2 was higher on the simulations (in particular for long insertions, which were enriched among TopHat2 results) indicating that many of the insertions additionally identified by these competing tools were not correct.

With regard to deletions, only GSNAP consistently recovered more reads with deletions than ContextMap 2 using BWA and again numbers decreased for annotation-based GSNAP. As the latter had both higher recall and precision on the simulations than GSNAP alone, this again suggests that the difference in mapped reads between GSNAP with and without annotation were false positives. Compared to ContextMap 2, annotation-based GSNAP identified a higher fraction of longer deletions. As the simulations showed a significantly lower precision, in particular on long deletions, for GSNAP, this again indicates that a significant fraction of the additional reads with deletions identified by annotation-based GSNAP are incorrectly mapped.

3.3.7 Runtime comparison

Finally, we compared runtime between all evaluated programs on the simulated data sets (Table 3.3). Here, ContextMap 2 was much faster than all evaluated programs except STAR 1- and 2-pass. Here, STAR 1-pass was extremely fast, whereas STAR 2-pass was only ∼20-24% faster than ContextMap 2. However, the evaluation on the RGASP data showed that this improved runtime came at the cost of both lower precision and recall for all STAR variants, in particular STAR 1-pass, compared to ContextMap 2.

Highest runtime of all evaluated approaches was observed for GSNAP with >128 CPU hours, i.e. more than 5 days. Thus, although it performed well on the detection of multi-junction reads and indels, runtime is too large for practical purposes. Among the remaining competing approaches, MapSplice 2 performed best in the evaluation of alignment quality, but not consistently better than ContextMap 2 using BWA. With regard to runtime, however, it performed significantly worse with ∼30 CPU hours on both simulations compared to 11-16 CPU hours used by ContextMap 2. Here, lowest runtime was observed when using Bowtie and highest using Bowtie 2, in particular when increasing the maximum number of reported alignments k to 10. Thus, BWA is the best choice as integral alignment algorithm

Program	Simulation 1	Simulation 2
ContextMap Bwt1	11.67	11.02
ContextMap Bwt2 ($k = 3$, default)	16.47	15.58
ContextMap Bwt2 ($k = 10$)	24.98	24.55
ContextMap Bwa	11.58	14.00
ContextMap Bwa ann	11.92	14.15
MapSplice 2	31.43	28.62
STAR 1-pass	0.82	1.28
STAR 1-pass ann	1.05	1.58
STAR 2-pass	9.60	10.28
STAR 2-pass ann	9.57	10.80
TopHat1	20.1	28.43
TopHat1 ann	20.53	29.03
TopHat2	25.17	27.23
TopHat2 ann	34.32	39.68
GSNAP	147.73	128.15
GSNAP ann	160.78	140.27

Table 3.3: Runtime in CPU hours for each program on simulation 1 and 2, respectively. All methods were run using 8 cores on the same machines and with the same parameter settings as in the RGASP evaluation (Engstrom et al. [2013]). ContextMap with Bowtie 2 was run with the maximum number of alignments reported per read (k) set to 3 (default setting used for evaluating mapping quality) and 10, respectively. Runtime of STAR 2-pass includes the time required for running STAR 1-pass, indexing the genome with splice sites found in the first STAR run and re-running STAR.

for ContextMap 2 taking into account mapping quality and runtime.

3.4 Conclusion

In this chapter, we presented ContextMap 2, a new and improved version of the context-based RNA-seq mapping program ContextMap. The key novel features of ContextMap 2 are the plug-in structure, which allows integrating new developments in short read alignment, as well as the detection of multi-split alignments, insertions and deletions. Performance of ContextMap 2 integrating either Bowtie, Bowtie 2 or BWA was evaluated on data sets from the recent RGASP evaluation of RNA-seq mapping programs and compared to the best performers of this study.

This showed that performance of RNA-seq mapping can be improved substantially by replacing the internal short read alignment program by more recent methods or versions.

In this case, the use of BWA as integral alignment program generally improved recall and precision of ContextMap 2 compared to Bowtie and Bowtie 2 at only slightly higher or even lower runtime, respectively. Here, the plug-in structure of ContextMap 2 allows the extension to future versions of these alignment programs or even newly developed short read alignment programs with improved accuracy or runtime. Furthermore, this extension can also be performed by developers of such programs or other users of ContextMap 2 by simply implementing the interface. In contrast, other existing RNA-seq alignment programs are limited to one or at most two short read alignment programs. For instance, MapSplice 2 still uses only Bowtie and TopHat2 only supports Bowtie and Bowtie 2.

ContextMap 2 with BWA performed similarly well or better than other state-of-the-art RNA-seq mapping programs with regard to perfectly mapped reads on simulated data, while having at least ∼2-fold lower rates of reads mapped only part correctly or at completely wrong positions. Thus, reduced mapping rates of ContexMap on both simulated and real data can be mostly explained by lower rates of incorrectly mapped reads. ContextMap 2 using BWA showed high precision and recall on all evaluated tasks, in particular on the detection of long insertions and deletions. Furthermore, runtime was generally at least 50% lower than for the best competing programs. Only STAR 1- and 2-pass were faster, but showed significantly lower precision, in particular on spliced reads and splice junctions, and low recall on reads containing indels.

Chapter 4

Mining RNA-seq data for infections and contaminations

Motivation: RNA-seq mapping approaches are usually designed for mapping sequencing reads derived from a single species only. Moreover, the possibility that underlying samples are infected by microbes or viruses is generally completely ignored. In such a scenario, the sequencing reads derived from an RNA-seq experiment originate from the host species as well as from unknown microbes or viruses. In this study, we show that our mapping software ContextMap can be applied for detecting infecting agents or contaminants in an RNA-seq experiment. Furthermore, we present methods to assess confidence of mappings to identified species and to detect false positive hits. Using several real-life data sets, we show that ContextMap identifies species contained in a given sample with high precision and compare our results to several state-of-the-art metagenomic programs.

Publication: This chapter was published in PLoS ONE (Bonfert et al. [2013]). I moved Figure 4.1 and Figure 4.3 of the Supplementary Material of the original article to the main text. The remaining parts of the Supplementary Material can be found in Appendix B. Furthermore, I adapted the layout of the text and applied some minor changes to the text. Please note that the ContextMap version used in this chapter and in the corresponding publication is an earlier release than the ContextMap 2 version described in chapter 3. Nevertheless, all presented methods are also implemented in ContextMap 2 and can be applied in the same way as described here.

Author contributions: Caroline C. Friedel (CCF) and I designed the study. I implemented the ContextMap standalone version used here with the exception of a modification of Bowtie, which was implemented by Gergely Csaba. CCF and I developed and I implemented the methods for assessing confidence of mappings to identified species and for detecting false positive hits. Furthermore, CCF and I analyzed the data and co-wrote the article. Ralf Zimmer helped in revising the manuscript.

4.1 Background

Next generation sequencing (NGS) technologies provide novel opportunities for transcriptomic analyses beyond simple quantification of gene expression. As one of the major challenges in analyzing RNA-seq data is the identification of the transcriptomic origin of each sequencing read (mapping), this has inspired the development of several novel RNA-seq mapping tools, e.g. TopHat (Trapnell et al. [2009]), TopHat2 (Kim et al. [2013]), MapSplice (Wang et al. [2010]), RUM (Grant et al. [2011]), and RNASEQR (Chen et al. [2012]). While all of these rely on fast alignment algorithms such as Bowtie (Langmead et al. [2009]), they use different strategies to identify reads from exon-exon junctions, a problem unique to RNA-seq data. In general, these approaches choose the alignment with the minimum number of mismatches for each read and cannot resolve multiple possible mappings for a read with the same alignment score.

This problem is addressed by our recently developed ContextMap method (see Bonfert et al. [2012, 2015] and chapter 3), which makes use of information provided by reads mapped to the same genomic region and likely originating from transcripts of the same gene. Thus, ContextMap does not aim at finding the mapping with the minimum number of mismatches, but the most likely mapping in the context of all other reads, in this way resolving non-unique mappings with high accuracy.

Independent of the mapping algorithm used, reads are usually only mapped against the reference genome (and sometimes transcriptome) of the species for which samples were collected. This completely ignores the possibility that reads may originate from other sources, e.g. unexpected contamination of samples, such as *Mycoplasma* species which are often found as contaminants in cell cultures, as well as viral or microbial infections of patients from which samples were derived. As RNA-seq protocols cannot distinguish between RNA from different species, mRNA from the infecting species will automatically also be sequenced. Indeed, dual RNA-seq of a pathogen and its host has recently been proposed for studying expression changes in both species simultaneously (Westermann et al. [2012]) and we performed it already for MCMV infection (Marcinowski et al. [2012]). While in this case the infecting species is known and an additional mapping against the corresponding genome is sufficient, for most applications contaminations or infections are not known beforehand.

Such an application would be the diagnostic screening of patient samples for unknown microbial or viral infections. Here, precise identification of the infecting agent is essential for medical treatment. Furthermore, it can provide novel insights into diseases, in particular tumorigenesis, by connecting them to otherwise undetected infections. One example that shows this nicely are the cervical cancer-derived HeLa cells. Human papillomaviruses (HPV), in particular HPV-16 and -18, have since been recognized as a predominant cause of cervical cancer (Walboomers et al. [1999]; zur Hausen [2002]) and HeLa cells have been shown to express transcripts of the integrated HPV-18 genome (Inagaki et al. [1988]).

As we show in this study, HPV-18 expression can be easily detected in RNA-seq data of HeLa cells. While in this case this only confirms previous knowledge, in other cases novel connections between viral infections and tumorigenesis can be detected. For instance,

Castellarin *et al.* (Castellarin et al. [2012]) used RNA-seq of tumor and normal tissue samples to link colorectal carcinoma to *Fusobacterium* infection.

With standard RNA-seq mapping tools, mapping both against the host reference genome and all available microbial and viral genomes is only possible using a sequential approach (Moore et al. [2011]) and requires additional steps for resolving non-unique read mappings that often occur due to local or global similarities between genomes. In contrast, ContextMap can be directly applied to automatically mine for reads from an arbitrary number of genomes since it already implements sophisticated strategies for resolving multiple read alignments. This makes it possible to also apply ContextMap for metatranscriptomics of species communities, e.g. the gut microbiome. While a number of such metatranscriptomics studies have already been performed (Lim et al. [2012]; Valles et al. [2012]; Xiong et al. [2012]; Yu and Zhang [2012]), these generally used BLAST to identify the involved species and did not even use existing metagenomics methods (e.g. MEGAN4 (Huson et al. [2011]), GRAMMy (Xia et al. [2011]), or GASiC (Lindner and Renard [2013])) for species identification.

In this study, we show how ContextMap can be easily used to identify reads from multiple sources in parallel such as viral and microbial genomes. Furthermore, we present methods based on mapping-derived statistics to assess confidence of mappings to the identified species/strains and identify false positive hits due to similarities between genomes and missing genome sequences. While some of these methods require information only provided by the ContextMap algorithm, they can in general also be extended to post-process output of other mapping approaches. We illustrate the performance of the proposed methods on three applications.

First, we use RNA-seq data of HeLa cells to characterize HPV-18 expression in these cells and correlate this to ongoing cell proliferation. Second, we illustrate the potential pitfalls of misidentifying species or strains in case of missing genome sequences based on a re-analysis of the Castellarin *et al.* data and show how these pitfalls can be avoided. Finally, for in-vitro sequencing data of a microbial community, we show how the involved species/strains can be identified despite the presence of several very closely related species/strains in the reference set and compare our results to MEGAN4, GRAMMy and GASiC as well as a number of other metagenomics tools.

4.2 Materials and Methods

4.2.1 Identifying sequencing reads from multiple sources using ContextMap

In the previous chapter, we introduced ContextMap, a novel mapping approach for RNA-seq data (Bonfert et al. [2012, 2015]). The central concept of ContextMap is the so-called read *context*. This is defined as a set of reads originating from the same stretch of the genome, indicating that these reads were derived from the same transcript or different transcripts of the same gene. These contexts are defined based on initial alignments de-

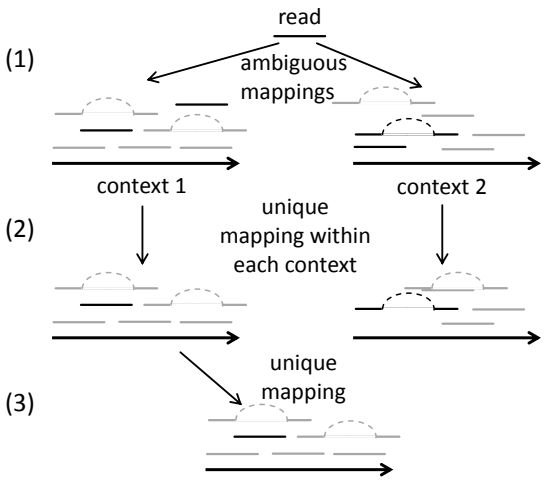

Figure 4.1: Central idea of ContextMap. (1) Within each context, ambiguous mappings are identified for each read with at most a maximum number of mismatches, including both full and spliced alignments. These ambiguous mappings may point to different contexts and may suggest different positions in the same context. (2) The best mapping within each context is identified for each read depending on the support by other reads. (3) Among the mappings to different contexts, the optimal one is chosen resulting in one unique mapping for the read.

termined with short read alignment programs such as Bowtie (Langmead et al. [2009]) or BWA (Li and Durbin [2009]). For each read not only the alignment with the minimum number of mismatches but any alignment to any context with at most a maximum number of mismatches is investigated. The unique mapping for the read to only one context is then determined by first finding the best mapping for the read in each context and subsequently finding the best context. For this purpose, a support score is used, taking into account the number of reads mapping within and around the region to which the read is aligned. Until the final step, contexts are treated independently of each other (see Figure 4.1).

As we show in this chapter, the advantage of this approach is that it allows investigating many alternative sources of reads in parallel, such as rRNA sequences, which are generally not included in reference genome assemblies of higher eukaryotes, as well as viral and microbial genomes. Contexts are then identified separately for each genome including the optimal context in each genome for each read. The final step is then used to decide for each read which of these contexts in any of the genomes considered results in the best mapping.

The parallel multi-species mapping is implemented by ContextMap in the following way (see Figure 4.2 A). First, the underlying alignment program is used to create independent indices for different potential read sources. Separate indices are necessary as, e.g. Bowtie is limited to 2^{32}-1 characters per index. This is relevant as the human genome alone needs 73% of the maximum index size and all microbial genomes from the NCBI database taken together require 134% of the maximum index size. We, thus, generally use one index for rRNA sequences, one for the host genome, e.g. the human reference genome, one for virus genomes and two for microbe genomes. This can be easily adjusted to more indices as soon

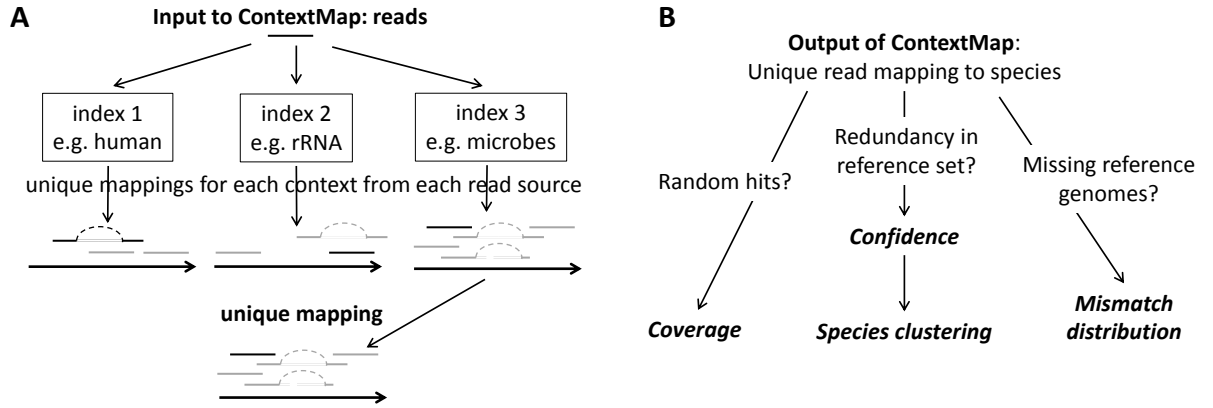

Figure 4.2: (A) Approach for mapping sequencing reads in parallel to multiple sources of reads using ContextMap. (B) After obtaining unique mappings to the species in the reference set, different questions can be addressed. Random hits to only a small region of the genome can be identified by investigating coverage. Strong similarities in terms of possible read mappings between different species in the reference set can be identified by analyzing confidence and species clusterings. Finally, by analyzing mismatch distributions in terms of the Jensen-Shannon divergence, it can be determined if reads have been mapped to the correct genome or only to a close relative due to missing genome sequences or local genome similarities.

as the increasing number of sequenced virus and microbe genomes makes this necessary.

After performing the initial alignment against all indices, ContextMap is then run without any further changes to define contexts, the optimal mapping for each read in each context it may belong to and finally the optimal and unique mapping for each read to any context.

In contrast to ContextMap, other RNA-seq mapping tools, which predominantly also use Bowtie, cannot be used for this application as they do not support the use of multiple indices required here due to the size and number of reference sequences and provide no way to distinguish between alternative alignments for a read to two different but related genomes with the same number of mismatches. Thus, they can only be applied sequentially by mapping first all reads e.g. against rRNA sequences, then the unmapped reads against the host reference genome, and then one microbe or virus genome one after the other. However, the latter approach also poses problems as it can lead to different results depending on the order in which genomes are mapped to in case of closely related species or strains.

4.2.2 Analysis of species hits

The mapping of reads to reference genomes using any algorithm directly implies a set of species potentially contained in the sample. Please note that in the following we use the term species loosely, in particular in the context of misidentification of species, and it may also refer to a particular strain of a species, represented by a specific genome sequence in the reference database. In particular for bacteria, the distinction between strains and

species is not clearly defined and species definition remains a difficult topic. The standard approach is now to use genome sequence differences and a cutoff of 95% average nucleotide identity is often used (Konstantinidis et al. [2006]). However, for species/strains for which no genome sequence is available, nucleotide identity to sequenced species/strains cannot be calculated. Thus, it cannot be determined whether they represent a different species or only a different strain of a species with known genome sequence.

Independent of which mapping algorithm was used to identify species potentially contained in a sample, a number of problems arise that need to be addressed. First, local similarities in the genome of one species not contained in the sample (species A) to a species contained in the sample (species B) may result in reads erroneously mapped to species A and the reporting of this species for the sample. Second, gaps in the reference database may lead to both missing and incorrect hits. If no genome from the species itself or closely related species is contained in the reference database, fast mapping algorithms, including ContextMap, which tolerate only a limited number of sequence differences, will fail to align the corresponding reads. This type of missing species hits is only a minor problem as a slower but more permissive BLAST run applied to unmapped reads may at least detect the infection by identifying more distant relatives of the infecting pathogen.

A more severe problem are misalignments in case that genome sequences are only available for closely related species. In this case, reads are incorrectly aligned to these related species, resulting in the identification of wrong species. For instance, in the recent study by Castellarin *et al.* (Castellarin et al. [2012]) several *Pseudomonas syringae* strains, which are plant pathogens, were likely misidentified in samples of colorectal carcinoma.

In the following, several statistics derived from read mappings are described that can be used to address the described problems and confidently identify the species contained in the sample (see Figure 4.2 B). Coverage and divergence of mismatch distributions can be calculated based on mappings provided by any algorithm. Calculation of species mapping confidence and distances between species relies on the support score calculated by ContextMap for each read mapping, but can be adapted to methods evaluating only the number of mismatches. All methods are available as part of the ContextMap software suite.

Read numbers

The standard approach for identifying the species contained in a sample based on the read mapping is to choose those species with the highest numbers of mapped reads. This is an important measure as small read numbers tend to indicate less likely matches. However, it can be misleading as local similarities to very small regions of the genome can lead to artificially high read numbers. As a consequence, we use read numbers only as one criterion for a hit and combine this with several other measures.

Coverage

To identify random matches, i.e. cases in which many reads are mapped to a small genome region only, we calculate the coverage of the genome by reads:

$$\text{coverage} = \frac{\#\text{ positions with mapped reads}}{\text{genome size}}. \tag{4.1}$$

Here, only the start positions of reads are counted. Mapping of reads to only a small fraction of the genome will result in very small coverage, suggesting a random hit. However, as coverage is influenced strongly by sequencing depth, low coverage for a correct hit may be observed in case of low sequencing depth. Thus, other measures have to be used in combination with coverage.

Mismatch distributions

Assuming that the average sequencing error is approximately the same for all species in the sample, an increase in mismatches in aligned reads for a species indicates that the identified species differs considerably from the actual species in the sample. To identify such cases, we compare the distribution of sequencing errors on mapped reads for each predicted species hit against a reference species for which we are certain that it is contained in the sample (e.g. the host species). The difference between the two mismatch distributions is calculated using the Kullback-Leibler divergence:

$$D_{KL}(P\|Q) = \sum_i \log\left(\frac{P(i)}{Q(i)}\right) P(i). \tag{4.2}$$

Here, $P(i)$ and $Q(i)$ are the fractions of mapped reads with i mismatches for the species under consideration and the reference species, respectively. Essentially, this quantifies the amount of information lost if Q is used to approximate P. As the Kullback-Leibler divergence is non-symmetric, i.e. $D_{\mathrm{KL}}(P\|Q) \neq D_{\mathrm{KL}}(Q\|P)$, we use a symmetric measure based on D_{KL}, the so-called Jensen-Shannon divergence:

$$D_{JS}(P \parallel Q) = \frac{1}{2}D_{KL}(P \parallel M) + \frac{1}{2}D_{KL}(Q \parallel M), \tag{4.3}$$

where $M = \frac{1}{2}(P + Q)$. The advantage of D_{JS} is that it is symmetric and has a clear-defined upper bound (= 1 if the base 2 logarithm is used for calculating D_{KL} (Lin [1991])). Furthermore, its square-root $\sqrt{D_{JS}}$ is a metric (Endres and Schindelin [2003]). Thus, in the following we will use $\sqrt{D_{JS}}$ to quantify differences of mismatch distributions between the identified species and the reference genome. Please note that for our examples $D_{KL}(P \parallel Q)$ and $\sqrt{D_{JS}}(P \parallel Q)$ were highly correlated.

The Jensen-Shannon divergence provides a quantification of the divergence between the actual species in the sample and the identified best hit but suggests no clear cutoff to discard potential hits. Instead, the choice of the cutoff depends strongly on the application

and the taxonomic level one is interested in. If the focus is on the genus level only, one may accept higher values of $\sqrt{D_{JS}}$ than for species identification. If one aims at identifying the actual strain even lower values of $\sqrt{D_{JS}}$ are acceptable.

As for the other measures proposed in this article, low $\sqrt{D_{JS}}$ should not be considered as the only criterion for a hit as it may result from random hits to a small region of a genome with few mismatches. Thus, other measures as coverage and the species mapping confidence as introduced below should also always be evaluated. In any case, high $\sqrt{D_{JS}}$ with a shift towards an increased number of mismatches indicates substantial divergence of the sequenced genome from the species in the sample, suggesting misidentification of the infecting species or strain.

Species mapping confidence

To identify mismappings due to similarities between genome sequences we calculate a score quantifying the confidence of read mappings to each species. Here, confidence for an individual read mapping is evaluated in terms of the support score difference between best and second-best mapping provided by ContextMap. Please note that the final output of ContextMap contains only the single best mapping for each read to any of the provided reference genome sequences. Only the score of the second-best mapping is recorded for calculation of mapping confidence. For each species A, we calculate the following mapping confidence score relative to a set of other species S ($A \notin S$):

$$\mathrm{conf}(A, S) = \frac{1}{|R_A|} \sum_{r \in R_A} \frac{s_1(r) - s_2(r)}{s_1(r)}. \tag{4.4}$$

where R_A is the set of reads mapped to A, $s_1(r)$ the support score for r in species A, and $s_2(r)$ the best support score of r to a species in S. If a read r cannot be mapped at all to any other species in S, $s_2(r) = 0$. As $s_1(r) \geq s_2(r)$, confidence is between 0 and 1 and low species confidence indicates that many of the assigned reads might alternatively be mapped to another species in S with only a little reduction in the score. The confidence score definition can be easily adapted to other mapping approaches by defining a support score measure for the corresponding mapping algorithm, e.g. based on the number of mismatches.

Clustering of genome hits

As ContextMap always assigns unique mappings to reads, a number of reads may still be mapped to related genomes for which they might be a better match due to sequencing errors. This is in particular the case if the genome for the microbe or virus contained in the sample is not known. In this case, reads from this microbe or virus may be dispersed over many relatives depending on local similarities. To identify such reads that likely originate from the same genome, we perform a clustering of genome hits using a dissimilarity function that is based on the relative mapping confidence of two genomes with regard to each other

as defined in equation 4.4. The mapping dissimilarity of genome X and Y is defined as

$$\mathrm{d(X,Y)} = \frac{\mathrm{conf(X,\{Y\})} + \mathrm{conf(Y,\{X\})}}{2} \tag{4.5}$$

Thus, if many reads mapped to genome X could alternatively be mapped almost as well to genome Y and vice versa, d is small. Like the confidence function, d is in the range of 0 and 1. Furthermore, it is symmetric and can be used with standard distance-based clustering methods.

4.2.3 Data sets

RNA-seq of HeLa cells

RNA-seq data of HeLa cells were taken from the study of Guo *et al.* (Guo et al. [2010]) who analyzed regulation of mammalian cells by miRNAs using both RNA-seq and ribosome profiling (Gene Expression Omnibus accession no. GSE22004). In this study, Illumina RNA-seq was performed for miRNA transfected HeLa cells at 12 and 32 h post-transfection. We used the RNA-seq data of mock and miR-155 transfected cells at 12 h post-transfection (28,735,355 and 29,595,334 36 bp reads, respectively).

RNA-seq of human colorectal carcinoma samples

For the second analysis, we used RNA-seq data for matched pairs of colorectal carcinoma and adjacent normal tissue samples from the study of Castellarin *et al.* (Castellarin et al. [2012]). Sequencing reads (75 bps) for 12 pairs of tumor and normal tissue were downloaded from the NCBI Sequence Read Archive (accession no. SRP007584). Although Castellarin *et al.* reported only the analysis of 11 sample pairs, 12 were available for download and no indication was given which of these were analyzed. Thus, we used all of them.

DNA-seq of *in-vitro* microbial communities

To compare our approach against standard metagenomics tools, we used pyrosequencing data of an *in-vitro* simulated microbial community (Morgan et al. [2010]). In this study, cultures for 10 species (yeast, *Halobacterium sp. NRC-1, Pediococcus pentosaceous, Lactobacillus brevis, Lactobacillus casei, Lactococcus lactis subsp. cremoris SK11, Lactococcus lactis subsp. cremoris IL1403, Myxococcus xanthus DK 1622, Shewanella amazonensis SB2B, Acidothermus cellulolyticus 11B*) were grown, cell pellets from a known number of cells for each species were mixed and DNA was extracted and sequenced. Thus, the exact species contained in this sample were known beforehand. Sequencing reads for pyrosequencing data were downloaded from the NCBI Short Read Archive (accession no. SRA010765.1). To simulate NGS data, which in contrast to pyrosequencing data is characterized by both a uniform read length as well as shorter reads, we trimmed reads to 100 bps and discarded reads shorter than 100 bps, resulting in 484,629 reads.

Reference genomes

Reference genomes for human (GRCh37) and yeast (sacCer3) were downloaded from the UCSC genome website (http://genome.ucsc.edu/). Completed microbe and virus genomes from RefSeq release 52 were downloaded from the NCBI ftp site (2919 microbial and 4092 viral genomes). For the analysis of the colorectal carcinoma data, we additionally used draft genome sequences from the Human Microbiome Project (NIH HMP Working Group et al. [2009]).

4.3 Results and Discussion

4.3.1 HPV-18 expression in HeLa cells

RNA-seq data of HeLa cells from the study of Guo *et al.* (Guo et al. [2010]) were mapped using ContextMap against indices for human, viral and microbe genomes and human rRNA. For the initial Bowtie runs a seed of 25 bps was used allowing up to 1 mismatch in the seed, the same settings used by Guo *et al.* In total, 5 mismatches were allowed, resulting in 11,040,798 (38.4%) and 10,162,289 (34.3%) mapped reads for the mock and miR-155 transfected cells, respectively. This is only 0.4 and 1.8 million reads less than mapped by Guo *et al.*, although they allowed an arbitrary number of mismatches outside the seed, i.e. up to 12 mismatches.

Although Guo *et al.* did not perform alignment against viral or microbial genomes (but also rRNA), only few ($\sim 35,000$) of the reads additionally mapped by ContextMap originated from viral or microbial genomes. Most reads additionally aligned by ContextMap were discarded by Guo *et al.* due to non-unique alignments. Interestingly, ~ 1.94 million reads originated from rRNA, which illustrates the importance of including rRNA sequences in the mapping process even though poly-A selection was performed.

Table 4.1 shows coverage, mapping confidence and $\sqrt{D_{JS}}$ compared to the human reference genome for all species with at least 1,000 mapped reads. Figure 4.3 illustrates coverage for all microbial or virus hits. Here, HPV-18 is the only virus or microbe with a coverage $\gg 0.01$ (0.34-0.37), high confidence (~ 1.0) and small $\sqrt{D_{JS}}$ (< 0.05) in both samples. This confirms previous reports of HPV-18 expression in HeLa cells (Inagaki et al. [1988]). In contrast, no reads were mapped to HPV-16, which is not expressed in HeLa cells.

Figure 4.4 A shows the distribution of reads across the HPV-18 genome both in the mock and miR-155 transfected cells. Here, results were highly reproducible between the two samples with peaks in read heights at the same genomic locations. The mapping to genes showed that only the E6, E7 and E1 genes were strongly expressed. In addition, weaker expression by an order of magnitude was observed for L1 as well as for a region covering the end of E1 and the start of E2. However, as no reads were observed for the rest of E2, it is likely not expressed. The same was true for genes E4, E5 and L2. These observations are in accordance with recent results showing that the oncogenes E6 and E7 are essential for continued proliferation in cervical carcinoma (Magaldi et al. [2012]). Both

species	# reads	coverage	confidence	$\sqrt{D_{JS}}$
A) mock transfected cells				
Human papillomavirus - 18	22105	3.7e-01	1.000	0.048
Hepatitis C virus genotype 6	1278	2.3e-03	0.558	0.351
Encephalomyocarditis virus	3105	2.4e-03	0.239	0.382
Thermoanaerobacter wiegelii Rt8.B1 chr.	28366	5.6e-05	0.087	0.460
B) miR-155 transfected cells				
Human papillomavirus - 18	18491	3.4e-01	1.000	0.045
Choristoneura occidentalis granulovirus	1144	2.4e-04	0.998	0.665
Encephalomyocarditis virus	4130	2.7e-03	0.258	0.418
Acinetobacter sp. ADP1 chromosome	1070	1.4e-04	0.070	0.077
Caviid herpesvirus 2	2215	2.1e-04	0.069	0.463
Thermoanaerobacter wiegelii Rt8.B1 chr.	24505	5.1e-05	0.063	0.508
Acinetobacter calcoaceticus PHEA-2 chr.	1480	1.9e-04	0.063	0.120
Acinetobacter baumannii ATCC 17978	1498	1.7e-04	0.020	0.121

Table 4.1: Microbial and virus species with at least 1000 mapped reads in the mock (A) and miR-155 (B) transfected HeLa cells.

genes are transcriptionally repressed by the E2 protein and loss of E2 expression leads to upregulation of E6 and E7 (Schweiger et al. [2007]). Thus, loss of E2 expression in HeLa cells as well as high E6 and E7 expression is consistent with their origin from cervical carcinoma cells and ongoing proliferation.

This shows that our approach is capable of identifying HPV-18 infection in HeLa cells and distinguishing this from spurious matches to other species. However, as less than 1% of reads in our samples originated from HPV-18 (22,105 and 18,491, respectively), the question remains which sequencing depth is necessary for confidently identifying such an infection. To investigate this question, we randomly sampled reads from the miR-155 data set with sample sizes between 10^4 and 10^7 (see Figure 4.4 B). For each sample size, 10 random repetitions were performed and reads were mapped using ContextMap as described. Here, a sequencing depth of as low as 500,000 reads (1.7% of all reads) was sufficient to clearly distinguish the HPV-18 infection from spurious hits to other species. Although only 303 HPV-18 reads were identified on average at this sample size, almost all of these reads (90%) were mapped to distinct genome positions, resulting in a coverage of ~ 0.034. Although this coverage is small, it is more than an order of magnitude larger than for any of the other species at this sequencing depth and increases much faster with increasing sequencing depth.

To compare the proposed method against alternative approaches, we performed megablast alignments for the miR-155 data set against all microbial and viral genomes as well as human rRNA sequences and the human mitochondrial genome. Alignments with an E-value

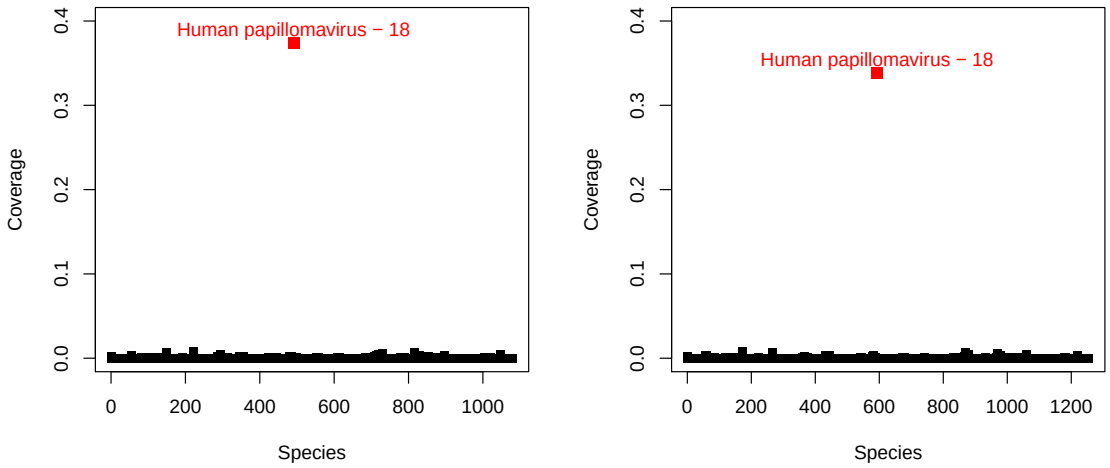

Figure 4.3: Coverage of all species identified for the mock (left) and miR-155 (right) transfected HeLa cells from the study of Guo *et al.*, respectively. The only species identified with a coverage > 0.01 is Human papillomavirus 18 (indicated in red) with coverages > 0.33.

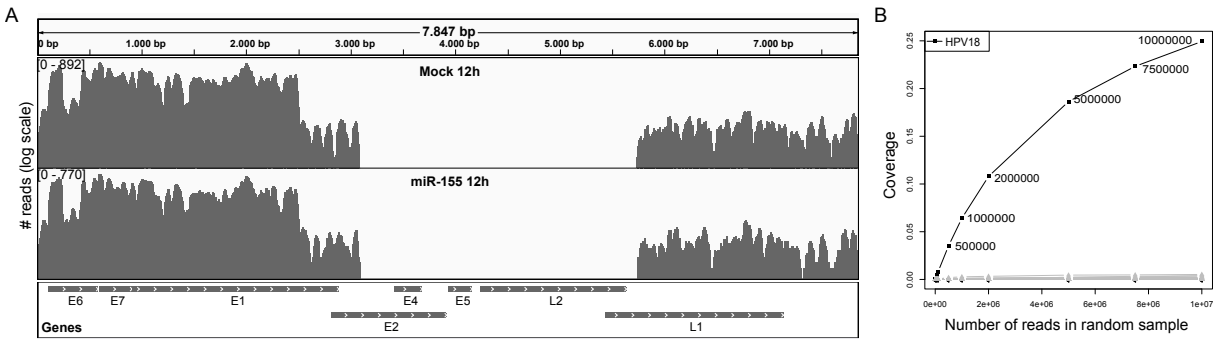

Figure 4.4: **Characterization of HPV-18 infection in HeLa cells.** (A) Distribution of reads across the HPV-18 genome for the mock and miR-155 transfected cells. Read numbers are shown in log scale. Expressed genes include E1 as well as E6 and E7, which are required for ongoing proliferation in cervical carcinoma (Magaldi et al. [2012]). L1 also appeared to be weakly expressed, however the expression pattern did not exactly correspond to the annotated gene coordinates. While the start of the gene was not expressed, L1 expression was extended to a region downstream of the gene. (B) Coverage as a function of increasing sequencing depth was evaluated by randomly sampling from the miR-155 data set. Coverage is shown as an average of ten repeated samplings for HPV-18 (black) and other species (gray). Sample size is annotated to the HPV-18 data points.

≤ 0.01 were then evaluated using MEGAN4. A megablast comparison against the complete human genome was aborted as output files already reached 10GB after mapping only 28% of reads against 30% of the genome, which would have resulted in an estimated 120GB of output (for an input of only 0.84GB). Since GASiC and GRAMMy could only be run in reasonable time on the > 60-fold smaller *in-vitro* microbial community data set by restricting them to the 10 species in question, we did not evaluate them here.

MEGAN4 results are shown in Supplementary Figure B.1 both with and without the additional alignment against human rRNA and mitochondrial genome sequences. In both cases, HPV-18 is clearly detected although 762 fewer reads (4.1%) are assigned than by ContextMap despite the fact that an arbitrary number of mismatches and gaps are allowed by BLAST. However, without additionally BLASTing against human rRNA and the mitochondrial genome, $> 11,000$ reads each are assigned to one bacterial (*Rickettsia rickettsii str. Hino*) and one viral (*Choristoneura occidentalis granulovirus*) species. When including human sequences for mapping, most of these are assigned to the inner nodes "cellular organisms" and "root", reflecting sequence similarities between human rRNA and the *Rickettsia* genome (lowest common ancestor (LCA) = "cellular organisms") and the human mitochondrial genome and the *Choristoneura* genome (LCA = "root"), respectively.

These results show the importance of also including the host species into mapping, as otherwise *Rickettsia* and *Choristoneura* would be reported erroneously for this sample. Here, MEGAN4 provides no direct way for identifying these hits as suspicious, e.g. by calculating coverage or mismatch distributions, or for resolving the non-uniquely mapped reads assigned to inner nodes. In contrast, ContextMap correctly assigns 90% of the *Choristoneura* BLAST hits to human rRNA and only 5% to *Choristoneura*. Furthermore, 88% of the *Rickettsia* BLAST hits are correctly identified as originating from human RNA (83% from mitochondrial RNA) by ContextMap and only 1% are assigned to *Rickettsia*. In addition, the few *Choristoneura* and *Rickettsia* reads assigned by ContextMap are clearly flagged as misalignments by very high values of $\sqrt{D_{JS}}$ (> 0.55).

4.3.2 The microbiome of colorectal carcinoma

In the second analysis, we focused on the RNA-seq data of colorectal carcinoma and adjacent normal tissue from the study of Castellarin *et al.* (Castellarin et al. [2012]). This data set was interesting as they identified a *Fusobacterium* to be enriched in colorectal carcinoma cancer. In addition, they reported a number of microbes that are unlikely to occur in colon tissue, e.g. *Pseudomonas fluorescens SBW25*, which was found at high levels in all samples, and two *Pseudomonas syringae* strains. *P. fluorescens* is mostly found in soil and water, whereas *P. syringae* are plant pathogens. While in the first case occurrence in colon samples might still be possible, e.g. due to contamination, in the latter case it is very unlikely. Although mapping with ContextMap also identified all three *Pseudomonas* species in all tumor and normal tissue samples, $\sqrt{D_{JS}}$ compared to the human reference genome was larger than 0.2 in all three cases (Supplementary Figure B.2), in particular for *Pseudomonas syringae pv. syringae* where more than half of the reads had at least 3 mismatches ($\sqrt{D_{JS}} = 0.458$). This indicates that the actual *Pseudomonas* species con-

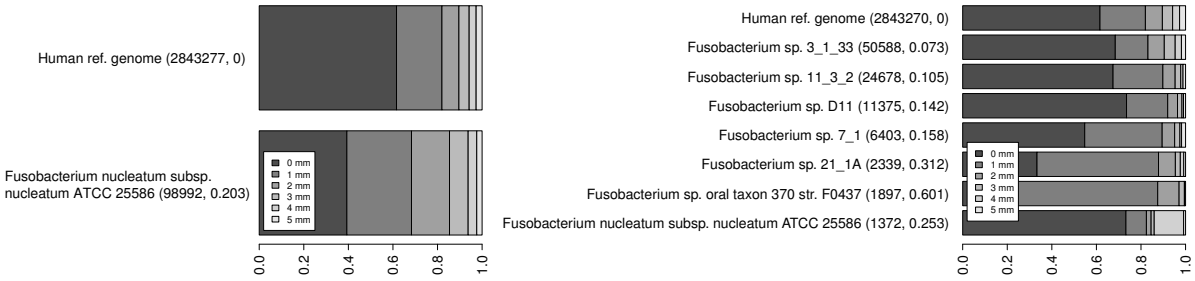

Figure 4.5: **Comparison of mismatch (mm) distributions for *Fusobacteria***. Results are shown for species identified by ContextMap with at least 20 reads on the colorectal carcinoma samples for patient 1 using either only the completed microbe genomes as reference set (left) or also the human microbiome draft genome sequences (right). Distributions are compared against the average mismatch distribution for the human genome. Number of reads mapped to each genome and $\sqrt{D_{JS}}$ are indicated in parentheses.

tained in the sample is not yet sequenced, resulting in reads from these species mapped to a number of related *Pseudomonas* species.

Based on these observations, we performed the same analysis for *Fusobacterium*. Previously, Castellarin *et al.* identified *Fusobacterium nucleatum subsp. nucleatum* as over-represented in the RNA-seq data of the tumor tissues. Subsequent DNA sequencing of a *Fusobacterium* culture isolated from the tumor samples and mapping of reads against additional *Fusobacterium* draft genomes from the Human Microbiome Project (HMP), however identified *Fusobacterium sp. 3_1_36A2* as a much better match than *F. nucleatum*. As *F. sp. 3_1_36A2* was extracted from the colon of a patient, this makes more sense than *F. nucleatum*, which was isolated from the human oral cavity and is most commonly found there.

To re-capitulate their analysis, we performed mapping of tumor samples using ContextMap both with and without the human microbiome in addition to the RefSeq genomes. Without the human microbiome index, *F. nucleatum* was identified in all tumor samples, in particular in samples from patient 1 ($\sim 100,000$ reads, Figure 4.5). However, the mismatch distribution differed considerably from the mismatch distribution for the human genome ($\sqrt{D_{JS}} = 0.203$ for patient 1), clearly indicating that *F. nucleatum subsp. nucleatum* is not contained in the sample but only a related species. Indeed, when performing mapping including the human microbiome, almost all of the reads originally mapped to *F. nucleatum* are mapped to contigs of other *Fusobacteria* species, such as *sp. 3_1_33, sp. 11_3_2, sp. D11, sp. 7_1*, and *sp. 21_1A*, which were isolated from biopsy tissues from the gastrointestinal tract. Furthermore, the primer sequences used by Castellarin *et al.* to confirm the presence of *Fusobacterium* are a better match to these species, with both primers matching with at most 2 mismatches, whereas for *Fusobacterium nucleatum* one primer has 3 mismatches.

Among the identified *Fusobacteria*, *F. sp. 3_1_33* has the highest number of reads for patient 1 ($> 50,000$) and smallest $\sqrt{D_{JS}}$ (0.073). It is also enriched in the tumor sample compared to the normal tissue, but not as strongly as some other species from the HMP

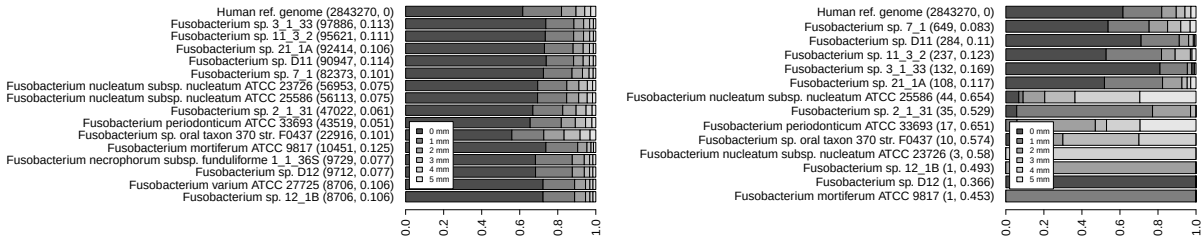

Figure 4.6: **Number of reads and mismatch distributions for the novoalign mapping on the *Fusobacteria*.** Results are shown for species identified by aligning with novoalign against viral and microbial genomes and the human microbiome for the patient 1 colorectal carcinoma sample. Only reads were used that were not mapped to human sequences by ContextMap. Mismatch distributions are compared against the average mismatch distribution for the human genome derived from the ContextMap mapping. Number of reads mapped to each genome and $\sqrt{D_{JS}}$ are indicated in parentheses. The left-hand side shows results if multiple read alignments with the same maximum score to different species are allowed. The right-hand side shows the results for unique alignments only.

with fewer mapped reads, in particular some *E. coli* strains (Table B.1). Although a comparatively small number of reads (1,372) are still assigned to *F. nucleatum* even with the inclusion of the human microbiome, the mismatch distribution still diverges strongly from the human reference ($\sqrt{D_{JS}} = 0.253$) and is unusual in that it has a higher number of reads both with zero and with four mismatches. Together with the observations that $\sqrt{D_{JS}}$ for *F. sp. 3_1_33* is still higher than in the HPV-18 example, a number of other *Fusobacteria* are also found with substantial read numbers, and most of the *F. sp. 3_1_33* reads can be aligned almost equally well to the other gastrointestinal *Fusobacteria*, this suggests that *F. sp. 3_1_33* is also not the actual strain in the sample. However, it appears to be a much closer relative than *F. nucleatum*. This also shows that $\sqrt{D_{JS}}$ should always be analyzed in combination with read numbers as local similarities may allow the mapping of some reads to a wrong species with few mismatches.

To compare our results against other approaches, we extracted all reads for the tumor sample of patient 1 that were not mapped to human sequences (including rRNA) by ContextMap (404,234 reads) and performed both megablast and novoalign alignments for these reads against virus and microbe genomes and the human microbiome. Novoalign (http://www.novocraft.com) was used by Castellarin *et al.* to align reads to the bacterial and viral genomes after filtering out all reads that could be aligned to human rRNA, cDNA or the reference genome using BWA (Li and Durbin [2010]), a fast short read aligner applying a similar strategy as Bowtie. Thus, we effectively recapitulated their analysis in our study, this time also including the human microbiome. Again MEGAN4 was applied to the BLAST output as shown in Supplementary Figure B.3. Almost all ($> 99\%$) of the *Fusobacteria* reads could be aligned to more than one *Fusobacterium*, thus, resulting in an assignment of these reads to their LCA by MEGAN4. In addition, MEGAN4 allows no further analysis as to which of the identified *Fusobacteria* is the most likely candidate or closest relative of the species or strain contained in the sample.

Novoalign was applied in two modes: one outputting all alignments for a read with the

same maximum score and one outputting only unique alignments (Figure 4.6). To compare against the ContextMap results and calculate $\sqrt{D_{JS}}$ compared to the reads mapped to human by ContextMap, we then extracted only those alignments with at most 5 mismatches and no gaps. Please note that read numbers were hardly increased for the *Fusobacteria* if gaps were allowed. For evaluation of the novoalign mode allowing multiple alignments, we used only one of the best alignments for each read for each genome, but allowed multiple alignments with equal score to different genomes. Here, almost all reads could be aligned equally well to more than one genome with $< 1\%$ unique read alignments per genome. Although $\sim 57,000$ reads were still aligned to *F. nucleatum*, only 44 of these were unique and $\sim 98\%$ were aligned equally well to *F. sp. 3_1_33*. Again, this illustrates the problems similarities between sequenced genomes present for mapping algorithms that are based only on the individual read alignments. Without taking into account alignments of other reads, they may only either completely exclude or include non-unique alignments. In this application, a restriction to unique alignments would vastly underestimate *Fusobacterium* expression in the sample, whereas the inclusion of non-unique mappings would result in the reporting of essentially all of the identified *Fusobacterium* species. In this case, evaluation of $\sqrt{D_{JS}}$ is not meaningful as due to the multiple mappings the sets of reads assigned to each species and, consequently, the calculated mismatch distributions are very similar.

4.3.3 Meta-transcriptomics for an in-vitro simulated microbial community

For the final analysis, we analyzed DNA sequencing data for an *in-vitro* simulated microbial community by Morgan *et al.* (Morgan et al. [2010]) and compared our results against several state-of-the-art metagenomics tools, in particular MEGAN4 (Huson et al. [2011]), GRAMMy (Xia et al. [2011]), and GASiC (Lindner and Renard [2013]). This data set was selected as the species contained in the samples were known. Furthermore, it constituted a challenging application due to strong similarities of the genomes of the microbial strains contained in the sample to other sequenced genomes. One example for this is *Halobacterium sp. NRC-1*, whose genome is almost identical to the *Halobacterium salinarum R1* genome (Pfeiffer et al. [2008]). They differ only by 4 base changes, 5 single-nucleotide indels and 3 longer indels between 133 and 10,007 bps long.

We investigated the performance of ContextMap on this data set using a reference containing the yeast genome and all microbial and viral genomes downloaded from NCBI (see methods) and allowing 5 mismatches. To compare our results against BLAST as well as MEGAN4 and GRAMMy, which use BLAST alignments as input, we performed megablast searches against the same genomes and extracted all alignments with the maximum score for each read, using only alignments without gaps and at most 5 mismatches. Here, 12% of reads could be aligned equally well to at least two different RefSeq entries using BLAST. In addition, we applied GASiC to all genomes from the same genus as any of the species contained in the sample (122 RefSeq entries, 92 taxa). The same restriction was applied to GRAMMy as both methods already took more than 7 CPU hours on this smaller set

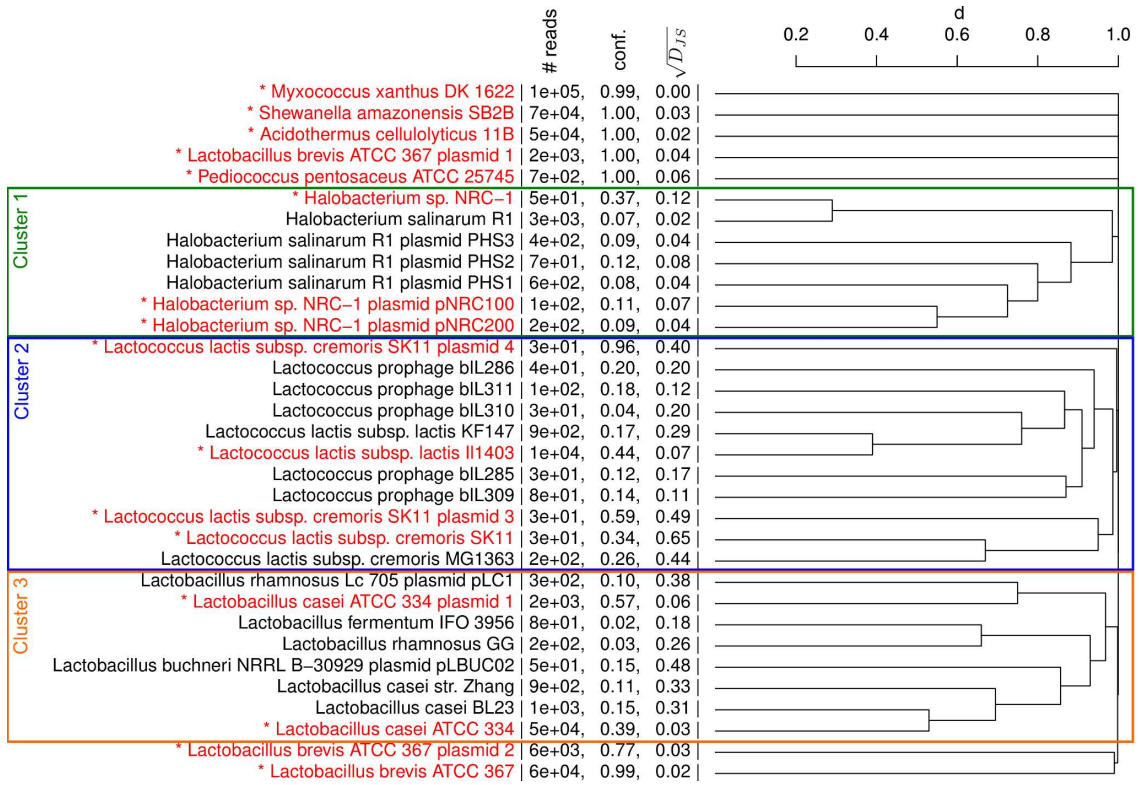

Figure 4.7: **Hierarchical clustering (average linkage) of microbes and viruses**. Results are shown for hits with a coverage $> 10^{-5}$ and at least 20 mapped reads as determined by ContextMap. Microbes actually contained in the sample are indicated in red and by an asterisk and the three clusters discussed in the text are marked by rectangles. In addition, number of reads, confidence and $\sqrt{D_{JS}}$ are indicated next to the microbe names.

compared to ~ 30 min for ContextMap on all microbes and viruses (Table B.2).

Table B.3 lists the microbe and virus hits identified by ContextMap with a coverage $> 10^{-5}$ and at least 20 reads. Here, ContextMap identified all of the microbial species contained in the sample, but also several related strains and prophages. As *Myxococcus xanthus* had the highest number of mapped reads, we used it as reference for calculating $\sqrt{D_{JS}}$. Interestingly, all microbes that are contained in the sample had a higher mapping confidence than all other hits despite low numbers of reads for some of them.

For five species, identification is straightforward based on this list. *A. cellulolyticus*, *S. amazonensis*, *L. brevis*, and *M. xanthus* are characterized by high mapping confidence (> 0.98), low $\sqrt{D_{JS}}$ (< 0.05) and high number of reads and coverage. For *P. pentosaceus* confidence is also high and $\sqrt{D_{JS}}$ still relatively low (0.064), but coverage is quite small ($< 10^{-3}$). However, as 90% of the reads map to distinct positions, it is clearly a correct hit and the low coverage is likely due to low abundance of *P. pentosaceus* in the simulated community. In the clustering of species hits, these five species also form distinct clusters with no similarities to any of the other species hits (Figure 4.7).

For the remaining hits the situation is less clear-cut. Here, clustering identified three

large groups among these: (1) a *Halobacterium* cluster, (2) a *Lactococcus* cluster, and (3) a *Lactobacillus* cluster. In the first case, *H. sp. NRC-1* clusters tightly with *H. salinarum R1* and the plasmids also cluster together, reflecting the small number of sequence differences between these. In all of these cases, $\sqrt{D_{JS}}$ is relatively small (≤ 0.08), with the only exception being *H. sp. NRC-1* for which all 50 reads have zero mismatches, i.e. fewer mismatches on average (Figure B.4). Additionally, confidence for *NRC-1* (0.37) is much higher than for *R1* (0.07). Thus, although significantly fewer reads are mapped to *NRC-1*, all other mapping statistics support the presence of the *NRC-1* strain rather than the *R1* strain. However, since both strains are almost identical, $\sqrt{D_{JS}}$ is still very low for the R1 strain (0.02).

In the second cluster, read numbers, confidence and $\sqrt{D_{JS}}$ clearly indicate the presence of *L. lactis subsp. lactis Il1403*. Although $\sqrt{D_{JS}}$ (0.07) is somewhat increased compared to the microbes identified unambiguously, it is not yet large enough to question this hit. For all other *Lactococcus lactis* strains, in particular *cremoris SK11*, the number of mismatches is significantly increased, indicating that these are not contained in the sample. This is surprising as *SK11* was part of the community. The reason for this is that 99% of the reads potentially mapping to *SK11* can be aligned equally well to other species, in particular to *L. lactis subsp. lactis Il1403*. As the latter is more abundant, it ends up with most of the reads, apart from those with too many mismatches. Finally, analysis of the last cluster confirms the presence of *L. casei ATCC 334* as it is characterized by high coverage and confidence and sufficiently low $\sqrt{D_{JS}}$ (0.06). The other strains in the cluster, in particular the *BL23* and *Zhang L. casei* strain, can be clearly excluded due to high $\sqrt{D_{JS}}$ (> 0.31) and low coverage ($< 5 \cdot 10^{-4}$) and confidence (≤ 0.15).

In summary, these results show that ContextMap can be used to correctly identify all species in the community including the strain, with the exception of *cremoris SK11*. However, analysis of results for MEGAN4 (Figure B.5), GASiC (Table B.4) and GRAMMy (Table B.5) shows that none of these identify *cremoris SK11*, at least not with more confidence than for other species/strains not contained in the community. MEGAN4 assigns almost all of the *cremoris SK11* reads to the LCA of the *cremoris* and *lactis* subspecies. GASiC assigns a p-value of 1, i.e. considers it an insignificant hit. Finally, GRAMMy, which only estimates relative abundances but performs no read mapping, assigns an abundance of $< 0.04\%$, less than assigned to *L. lactis subsp. lactis KF147* (0.17%), which is not part of the community.

Apart from *cremoris SK11*, GASiC fails to identify *H. sp. NRC-1* and *P. pentosaceus* but otherwise predicts only microbes contained in the community. Thus, GASiC is the most restrictive of the analyzed approaches. MEGAN4 does not really resolve multiple mappings but assigns reads with multiple mappings to the LCA of these microbes. Based on the number of reads assigned uniquely to any of the children of such an LCA, the correct microbes can then be predicted. In this example, the predictions would be correct with the exception of *SK11* and *H. salinarum*, where the *NRC-1* strain cannot be properly distinguished. Nevertheless, even when combining read numbers for the microbes and the LCA, ContextMap generally identifies 2.5-6.5% more reads per microbe (including plasmids). Furthermore, assignments to inner nodes of the phylogenetic tree by MEGAN4

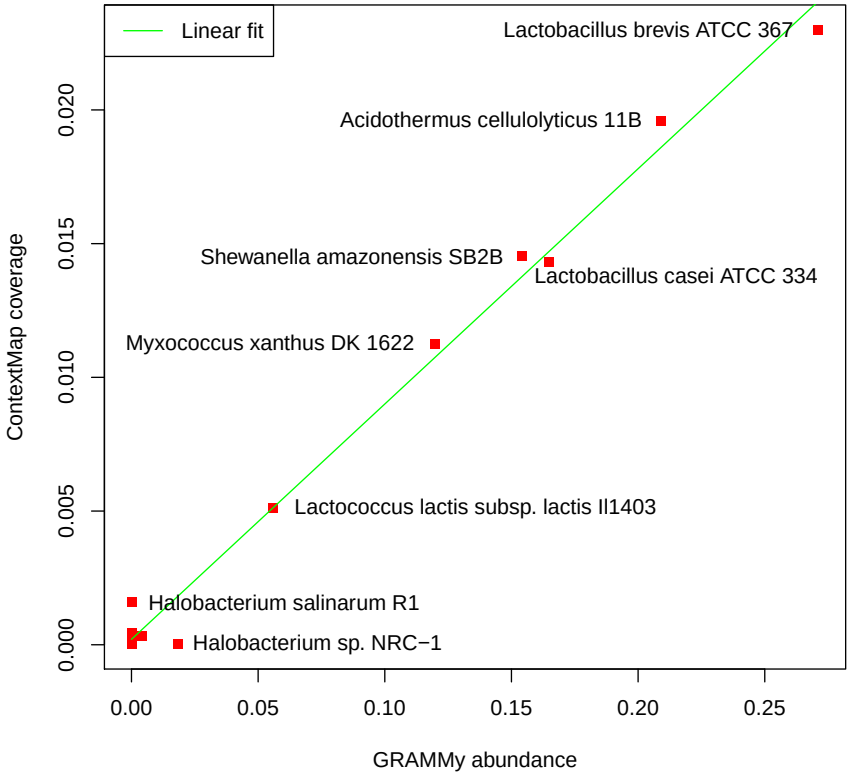

Figure 4.8: **Comparison of abundance calculated by GRAMMy and coverage determined by ContextMap on the microbial community data set**. Results are shown for all taxa identified by GRAMMy with a relative abundance of at least 0.1%. The green line indicates a linear fit to the data.

do not allow calculation of mismatch distributions as corresponding genome sequences are not known. Even for the leaves of the taxonomic tree, additional statistics of alignment quality or coverage are not directly accessible and can only be obtained by extracting the assigned reads and performing this analysis using additional scripts.

GRAMMy correctly identifies 7 of the 9 microbes with an estimated abundance of $> 1\%$, but also assigns a very low abundance to *P. pentosaceus* (0.4%). Remarkably, the relative frequency estimated by GRAMMy and the coverage calculated by ContextMap are highly correlated (correlation coefficient 0.995), in particular for microbes with high coverage (Figure 4.8). This indicates that coverage as determined by ContextMap provides a reliable estimation of the relative frequencies identified by GRAMMy. As ContextMap is much faster than GRAMMy, it can thus be used to replace GRAMMy for applications where GRAMMy is too inefficient.

We also evaluated a number of other metagenomics tools for binning/classifying sequencing reads or identifying relative abundance of species. This includes alignment-based

approaches (MG-RAST (Meyer et al. [2008]), MetaPhyler (Liu et al. [2011]), SOrt-ITEMS (Monzoorul Haque et al. [2009]), MARTA (Horton et al. [2010]), MLTreeMap (Stark et al. [2010])), composition-based approaches (PhyloPhytiaS (McHardy et al. [2007]), ClaMS (Pati et al. [2011]), Phymm (Brady and Salzberg [2009])) and a hybrid approach (PhymmBL (Brady and Salzberg [2009])). Here, comparison of the results was difficult as several approaches only perform classification at the genus- (MetaPhyler) or species-level (MG-RAST, PhyloPythiaS, MARTA), but do not identify individual strains. Thus, we could not evaluate their performance in distinguishing the *Halobacterium* and *Lactococcus lactis* strains. Furthermore, only MG-RAST, MetaPhyler, Phymm and PhymmBL were developed for NGS reads as short as 100 bps, while the other tools require longer reads. Thus, the meaningfulness of the comparison against these other approaches is limited. Results for all tools are shown and discussed in Tables B.6 (MG-RAST), B.7 (MetaPhyler), B.8 (SOrt-ITEMS), B.9 (MARTA), B.10 (MLTreeMap), B.11 (PhyloPhytiaS), B.12 (ClaMS) and B.13 (Phymm and PhymmBL). In summary, although the correct species or at least genera were usually identified, performance at the level of the individual strains was usually poor as often wrong strains were ranked higher than strains contained in the community. A particular poor performance was observed for the composition-based approaches PhyloPhytiaS and ClaMS, which likely suffered from the short sequencing read length.

The analysis of runtime and memory requirements on this data set (Table B.2) showed that ContextMap was both faster than almost all other approaches (apart from MetaPhyler) and required less or a comparable amount of memory (with the exception of MLTreeMap and GRAMMy if memory requirements of the BLAST run to provide the input for GRAMMy are not counted). The comparison for the other two data sets was less informative as ContextMap was either applied on more reads as in the case of the colorectal carcinoma data set or more reference sequences as in the case of the HeLa cell RNA-seq data. In the first case, ContextMap took $\sim$ 0.02 sec per read on the complete 5,343,842 read set for patient 1, whereas BLAST took $\sim$ 0.19 sec per read on the smaller 404,234 read set without human reads and novoalign only $\sim$ 0.006 sec per read. Thus, ContextMap was much faster than BLAST but slower than novoalign. In contrast, ContextMap required much less memory with $\sim$ 10G for the complete 5 million read set compared to the $\sim$ 15G required by novoalign for only 400k reads and $>$ 27G required by BLAST. This large memory requirement (and also runtime) of BLAST for this relatively small data set was rather remarkable, in particular in comparison to the similar-sized microbe data set and the much larger HeLa data set, which both only required $\sim$ 2.5G. The reason for this is the much larger number of possible hits per read found by BLAST in the colorectal carcinoma data set ($\sim$ 400 hits per read) as compared to the microbe ($\sim$ 8 hits per read) and HeLa data set ($\sim$ 0.2 hits per read). It should be noted that in the latter case we did not perform simultaneous mapping with BLAST against the complete human genome as the estimated output size was too large. Thus, these results suggest that a complete run of BLAST on the same references as used for the ContextMap run would have resulted in substantially increased runtime and memory requirements compared to ContextMap.

4.4 Conclusion

In contrast to microarray experiments, RNA-seq is not limited to previously defined probes but allows quantification of all transcripts in the cell, including also transcripts expressed by viral or microbial pathogens. However, current mapping approaches generally ignore the possibility of multiple origins of reads and are not designed to resolve resulting non-unique mappings. Thus, RNA-seq experiments are not routinely mined for the presence of contaminations or infections. Previous studies explicitly focusing on metatranscriptomics generally used only BLAST despite the availability of a number of metagenomics tools for identifying the species in the sample. As some of these tools were published only very recently, this might explain why they have not yet permeated the metatranscriptomics/-genomics community.

In this study, we showed how different sources of reads can be easily investigated in parallel using ContextMap without limitations to the number of potential sources investigated. This allows unbiased screening of RNA-seq data for transcripts of any species with a sequenced genome. ContextMap is particularly suited to this task as it tolerates a large degree of ambiguous mappings at intermediate steps, allowing multiple mappings to different species during these steps. These multiple mappings are then resolved in the final step using a support score calculated based on other reads aligned to the same region. From this support score, a confidence value can be calculated for each individual read mapping and the confidence of mappings for identified species and similarity of two species in terms of possible read mappings can be evaluated. This is of particular importance when mining RNA-seq data for the presence of related species. As previously published mapping methods generally cannot resolve multiple mappings and no scoring of alignments apart from mismatch counting is performed, the number of reads they cannot uniquely assign to a species is substantial. For instance, in the case of the microbial community, $> 54\%$ of *L. lactis subsp. lactis Il1403* reads can be aligned equally well to other *L. lactis* subspecies and thus cannot be resolved by mapping tools relying only on alignment quality.

Our approach was evaluated first on previously published RNA-seq data sets for HeLa cells, where it allowed the identification and characterization of HPV-18 expression leading to ongoing proliferation in this cervical carcinoma-derived cell line. Here, we showed that relatively small sequencing depth can already be sufficient for reliable detection of pathogen infections, e.g. for diagnostic purposes. A comparison against BLAST combined with MEGAN4 showed the importance of aligning reads against both host and pathogen species, as local sequence similarities of microbial or viral genome sequences to human sequences, in particular rRNA and mitochondrial DNA, would otherwise lead to wrong microbial or viral hits. While ContextMap correctly resolved most of the resulting non-unique hits, MEGAN4 effectively only flagged them as non-unique hits by assigning them to internal nodes close or equal to the root of the phylogenetic tree.

A second problem arising in the context of both metatranscriptomics and metagenomics are missing genome sequences for the species/strains in the sample, which may result in misalignments of reads to related species or strains. To identify such cases, we proposed to analyze differences of mismatch distributions compared to a reference species known to be

in the sample, e.g. the host species. This can be automatically evaluated using the Jensen-Shannon divergence and the usefulness of this approach was illustrated on the colorectal carcinoma data from the Castellarin *et al.* study. Here we showed that divergence of the mismatch distributions on the RNA-seq data suggested that *F. nucleatum* was not the *Fusobacterium* species in the tumor sample. Instead, a different *Fusobacterium* sequenced for the human microbiome project was identified as a more likely candidate. Again, application of MEGAN4 to BLAST results only indicated the presence of *Fusobacteria* in the sample, but could provide no further resolution as to which of the sequenced *Fusobacteria* is most likely present in the sample or most closely related to the species in the sample. We also compared our approach against the strategy used by Castellarin *et al.* by applying novoalign both to complete virus and microbe genomes and the human microbiome. Again, this approach suffered from the high similarity between *Fusobacteria*, resulting almost exclusively in non-unique hits.

Finally, we applied ContextMap to metagenomics of the *in-vitro* simulated microbial community to compare it against state-of-the-art metagenomics tools. Here, ContextMap vastly outperformed both GASiC and GRAMMy in terms of runtime, while also providing more helpful results. GASiC missed 3 of 9 microbial species in the community and furthermore allowed multiple alignments of reads to different species. In contrast, GRAMMy only determines relative abundances and does not perform any mapping of reads. Thus, it does not allow the analysis of gene expression or mismatch distributions. The latter also applies to MEGAN4, which performs no real resolution of ambiguous alignments and only assigns reads with multiple alignments to the lowest common ancestor of the corresponding species. Thus, neither GRAMMy nor MEGAN4 offer the same possibilities for gene expression analysis of microbes and viruses and identification of missing genome sequences as ContextMap, while GASiC is both much too slow and too restrictive for this application. Finally, comparison against several other metagenomics tools showed that all of these had problems in identifying the correct microbial strains contained in the sample.

Although analysis of coverage, confidence and Jensen-Shannon divergence provided by ContextMap requires some user interaction, in particular for picking thresholds, the same applies to GRAMMy, which also provides no natural cut-off on the predicted frequencies. In contrast, both GASiC and MEGAN4 basically do not allow any user interaction to fine-tune results. Despite the fact that GASiC calculates p-values, these are in most cases either 0 or 1 (at least in our application), allowing no tuning of thresholds to trade off sensitivity and specificity. Moreover, MEGAN4 provides no interface to resolve ambiguous mappings of reads assigned to an LCA or evaluate alignment quality, coverage or the other useful measures proposed here to improve the results. Finally, none of the other metagenomics tools provides any clear cutoff to determine the actual species in the sample, but only allow ranking of the possible hits, generally in terms of read numbers or estimated abundances. In any case, defining fixed thresholds for any application is likely not meaningful, as appropriate thresholds will strongly depend on the particular research question. For instance, if knowing the particular strain is of importance, e.g. in a diagnostic application where pathogenic or antibiotic-resistant strains have to be correctly identified, much lower values of Jensen-Shannon divergence would be allowed. In contrast, if only the genus or species

is relevant, one might even merge species or strains into one group if they are clustered together based on the mapping similarity measure we proposed.

An alternative approach that was not evaluated in this study is PathSeq (Kostic et al. [2011]), a software explicitly focused on identifying microbes from sequencing data of human tissues. We did not evaluate this software as it could only be run using Amazon Web Services, thus requiring payment for using the web services and making it not available for free. However, the pipeline basically consists of a mapping of reads against human sequences first and then a mapping of unaligned reads against microbial and viral sequences using BLAST, which is similar to the BLAST approach we evaluated in this study. Thus, we expect PathSeq to encounter the same problems, i.e. high numbers of non-unique hits due to similarities between microbial and viral species, no proper resolution of non-unique hits and misidentifications in case of missing genome sequences.

Finally, it should be noted that the metrics we proposed here for evaluating potential species hits are not limited to ContextMap but can be easily extended to other mapping tools or meta-transcriptomic pipelines to further post-process their output. For coverage and divergence of mismatch distributions, this is relatively straightforward but requires a strategy to address non-unique mappings. As shown for the novoalign results on the colorectal carcinoma data, mismatch distributions are not meaningful if high numbers of non-unique alignments are allowed. For calculation of confidence and species clusterings, a support score has to be defined to quantify the quality of an individual read alignment. Here, even simple alignment scores may be used, although the resolution of any approach based only on the individual read alignments is necessarily much lower than a more sophisticated approach taking also into account alignments of other reads as used by ContextMap. Thus, the methods proposed in this chapter will also be helpful for researchers preferring to keep to their already established pipelines and only post-process their results.

Chapter 5

Conclusion and Outlook

NGS and its application to RNA-seq allows for the quantification of all transcripts in a cell. The extremely high throughput rates of modern NGS machines result in millions of short sequencing reads per RNA-seq experiment. A crucial step in analyzing RNA-seq data generally is to determine a mapping of the sequencing reads to a given reference sequence (e.g. the genome).

In this work, we developed novel approaches for mapping RNA-seq data. In chapter 2, we introduced a straightforward mapping workflow that sequentially maps the sequencing reads to one reference sequence after another. The workflow aligns the sequencing reads first to a known transcriptome and subsequently the remaining unaligned reads to the reference genome. We applied our workflow in a recent study (section 2.3 and Windhager et al. [2012]), in which we analyzed a time-course RNA-seq experiment. The resulting read mapping allowed for visualizing the maturation of transcripts over time and provided evidence for very fast co-transcriptional splicing.

However, we also realized that our workflow and other existing RNA-seq mapping approaches have common drawbacks in their mapping strategies. These drawbacks either result from the fact that the underlying mapping approach does not consider all possible alignments of a sequencing read or has no general strategy for resolving ambiguous read alignments. Both problems can result in wrong read mappings, which has a direct influence on any downstream analyses.

For addressing these problems, we developed ContextMap, a context-based mapping approach (see chapter 3 and Bonfert et al. [2015]). The central idea of ContextMap is to determine the most likely origin of a read by considering all other reads aligned to the same genomic region (i.e. the read context). By evaluating the read context, ContextMap is able to accurately resolve ambiguous read alignments. We demonstrated in a proof of concept study (Bonfert et al. [2012]) that our first prototype implementation of ContextMap considerably improved existing mapping results determined by other mapping programs (e.g. MapSplice (Wang et al. [2010]) or TopHat (Trapnell et al. [2009])). However, by relying on other programs to provide a mapping result as input, ContextMap could not explore the whole alignment search space of all sequencing reads. Thus, ContextMap was not able to fully exploit its mapping potential.

Following the proof of concept study, we developed a standalone version of ContextMap. This implementation no longer depended on an existing mapping result in the input, but determined initial alignments itself using a modification of the Bowtie short read alignment program (Langmead et al. [2009]). Therefore, the standalone implementation of ContextMap was able to explore a larger alignment search space than its predecessor implementation by considering all alignments determined with the Bowtie modification. However, the ContextMap algorithm had several drawbacks. In particular, it was not capable of detecting reads crossing more than one exon-exon junction or to predict indels. Furthermore, due to the dependency on a modification of a specific Bowtie version, the ContextMap implementation could not benefit from newly developed algorithms for determining short read alignments.

In this thesis, we presented ContextMap 2, an extension of the original ContextMap algorithm. The key features of ContextMap 2 are the context-based resolution of ambiguous read alignments, the accurate prediction of reads crossing an arbitrary number of junctions and the detection of indels. Furthermore, we provide a plug-in interface for integrating alternative read alignment programs (e.g. Bowtie 2 (Langmead and Salzberg [2012]) or BWA (Li and Durbin [2009])) with improved accuracy or running times. We evaluated ContextMap 2 on synthetic and real-life data sets from a recent RGASP study (Engstrom et al. [2013]) and compared our results to other state-of-the-art approaches. Our results showed that ContextMap 2 had very low rates of incorrectly mapped sequencing reads, while the fraction of perfectly mapped reads was as high as of the best competing approaches. Moreover, the running time of ContextMap 2 was more than ∼2-fold lower than for programs with comparable high precision and recall values.

In addition to the mapping of RNA-seq data to a single species, ContextMap is also suitable for screening the data in parallel for transcripts of any species with a sequenced genome (see chapter 4 and Bonfert et al. [2013]). This feature is relevant in particular when RNA-seq data is derived from cells that were infected by viruses or microbes or in meta-transcriptomic studies. In such scenarios, it is very likely that sequencing reads can be aligned equally well to different genome sequences of related species. In chapter 4, we demonstrated on real-life and in-vitro data sets that ContextMap is able to accurately resolve those ambiguities. In addition, we developed mapping-derived statistics to assess confidence of identified species and misidentifications caused by local similarities between genomes or by completely missing genome sequences. Our results showed that our approach can be used to routinely mine for infections or contaminations in RNA-seq experiments. Additionally, as shown in a recent study (Rutkowski et al. [2015]), the parallel mapping of reads to a host species and a known pathogen allows to analyze the gene expression of both species at the same time. In the study by Rutkowski et al. [2015], we analyzed the impact of Herpes simplex virus-1 on human transcription and RNA processing.

Nevertheless, there is room for further improvements and extensions of ContextMap 2. Recently, we developed a method for predicting poly(A) cleavage sites by mapping reads containing poly(A)-tails. This method uses the context-based approach of ContextMap 2, as all reads supporting a putative cleavage site are considered for the exact localization of the site. Furthermore, the method is seamlessly integrated into ContextMap 2 and has only

little influence on the running time of a regular mapping run. Performance of the method was evaluated on three different cell lines for which RNA-seq data as well as RNA-pet data was made available by the ENCODE project (ENCODE Project Consortium [2012]). RNA-pet is a method that identifies 3' and 5' transcript ends using NGS technology (Ng et al. [2005]). We mapped the RNA-pet reads of each cell line to the human genome using ContextMap. Subsequently, the mapped RNA-pet reads from the 3' transcript ends were used to define a gold standard of cleavage sites in each cell line. Finally, cleavage sites were predicted with RNA-seq reads using ContextMap 2. We compared our results with KLEAT, a program for predicting poly(A) cleavage sites from assembled transcripts of RNA-seq reads (Birol et al. [2015]). Our evaluation shows that the predictions of ContextMap 2 are generally more accurate than the predictions of KLEAT. Currently, we are preparing a manuscript that contains a comprehensive evaluation of RNA-seq data for more ENCODE cell lines as well as recently published RNA-seq data sets of cells infected with various viruses.

Another extension of ContextMap 2 could be the development of alternative scoring models for evaluating the context of a sequencing read. These scoring models can be specifically designed for different types of sequencing data. For instance, the development of a scoring scheme for mapping ribosomal profiling reads might improve mapping accuracy of such data. Ribosome profiling is a technique that allows for the genome-wide measurement of translation by sequencing mRNA fragments protected by ribosomes (Ingolia et al. [2009]; Ingolia [2014]). Ideally, a mapping of ribosomal profiling reads to the genome allows for the identification of the exact genomic position of each ribosome bound to an mRNA. Thus, the reading frame of each ribosome can be inferred from the read mapping. Different ribosomes that translate an mRNA into the same protein elongate the mRNA in the same reading frame. Therefore, an associated scoring model could assign a high support score to a read, if many other reads are aligned to the same reading frame indicating for a frequently translated mRNA. On the contrary, if only few other reads are aligned to the same reading frame, then a low score will be assigned.

Finally, we are confident that technological advances in sequencing (e.g. increasing read lengths) can be successfully integrated into our context-based mapping strategy in order to improve mapping quality and running times.

Appendix A

Supplementary Material for chapter 3

In this chapter, we present the Supplementary Material for chapter 3. All figures and tables shown here were taken from the Supplementary Material of the ContextMap 2 article that was published at BMC Bioinformatics in 2015 (Bonfert et al. [2015]). I moved parts of the original Supplementary Material to chapter 3 and slightly modified the layout of the tables.

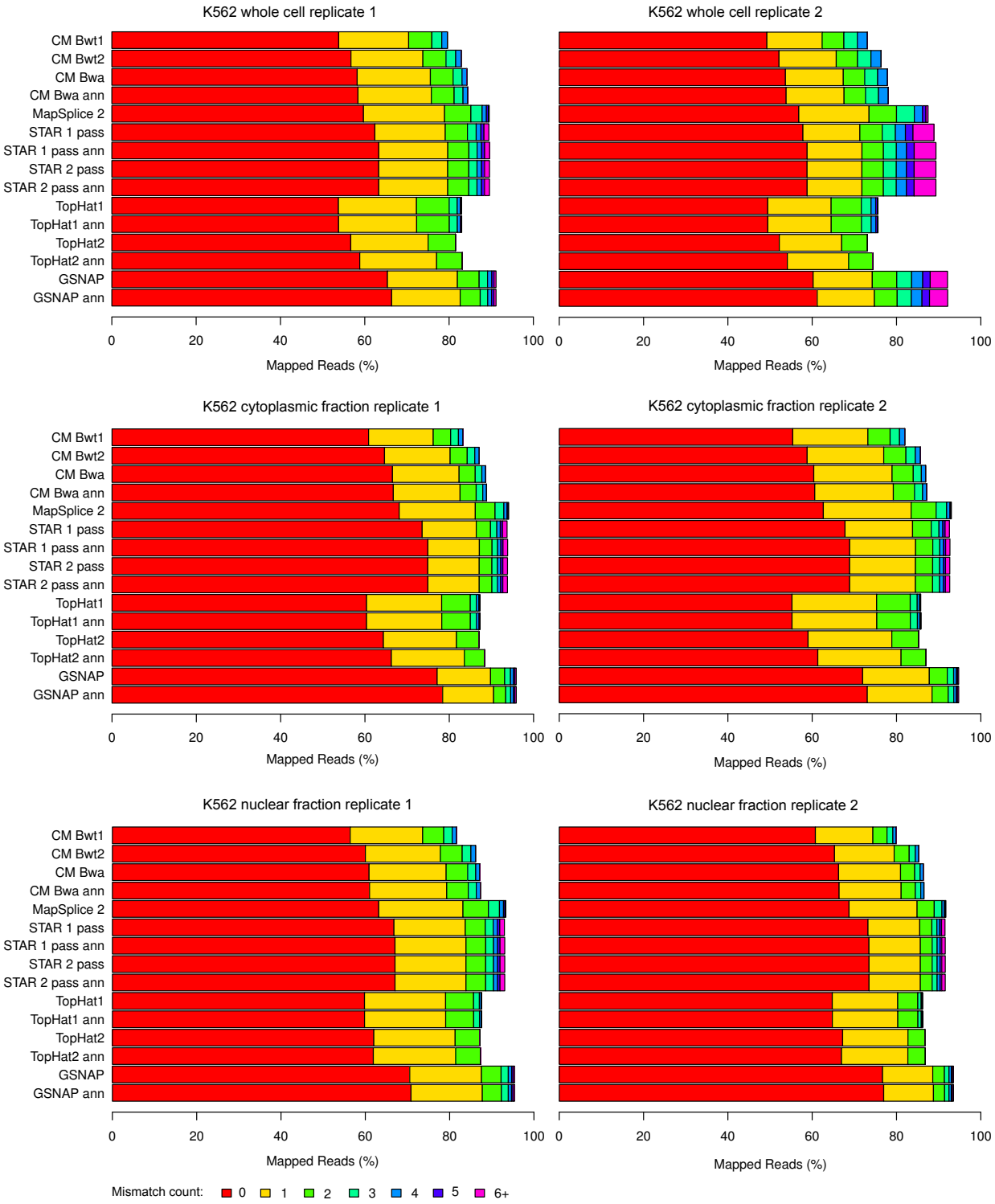

Figure A.1: Percentage of mapped reads and mismatch distribution for the mapped reads for all evaluated real-life data sets.

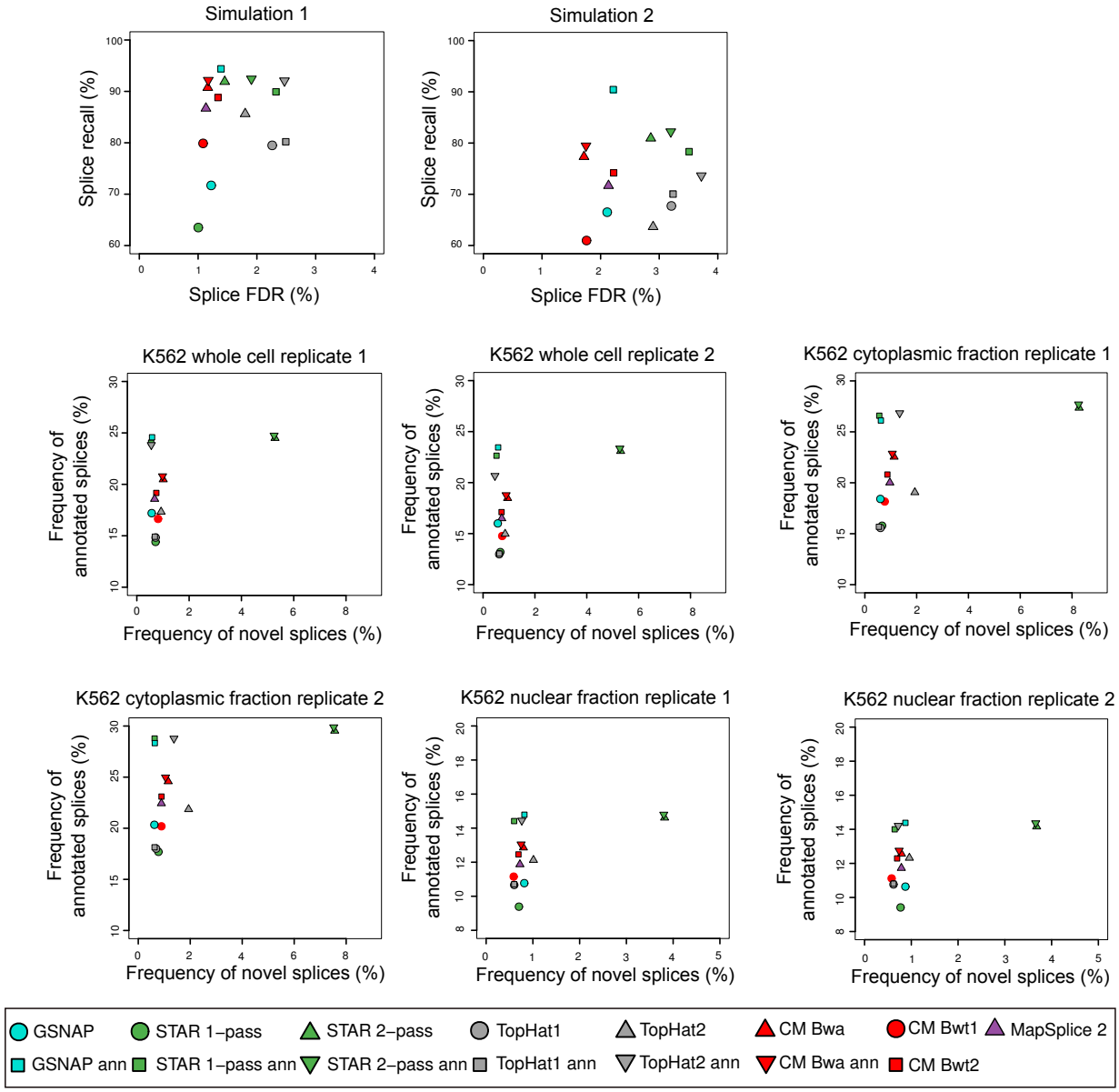

Figure A.2: Top row: Comparison of splice recall (y-axis) versus splice false discovery rate (FDR=1-precision, x-axis) on simulation 1 and 2 for all evaluated RNA-seq mapping programs. Bottom rows: Comparison of the frequency of predicted novel splices to the frequency of annotated splices for the Ensembl annotation for all evaluated real-life data sets and all evaluated RNA-seq mapping programs. See main manuscript for definitions.

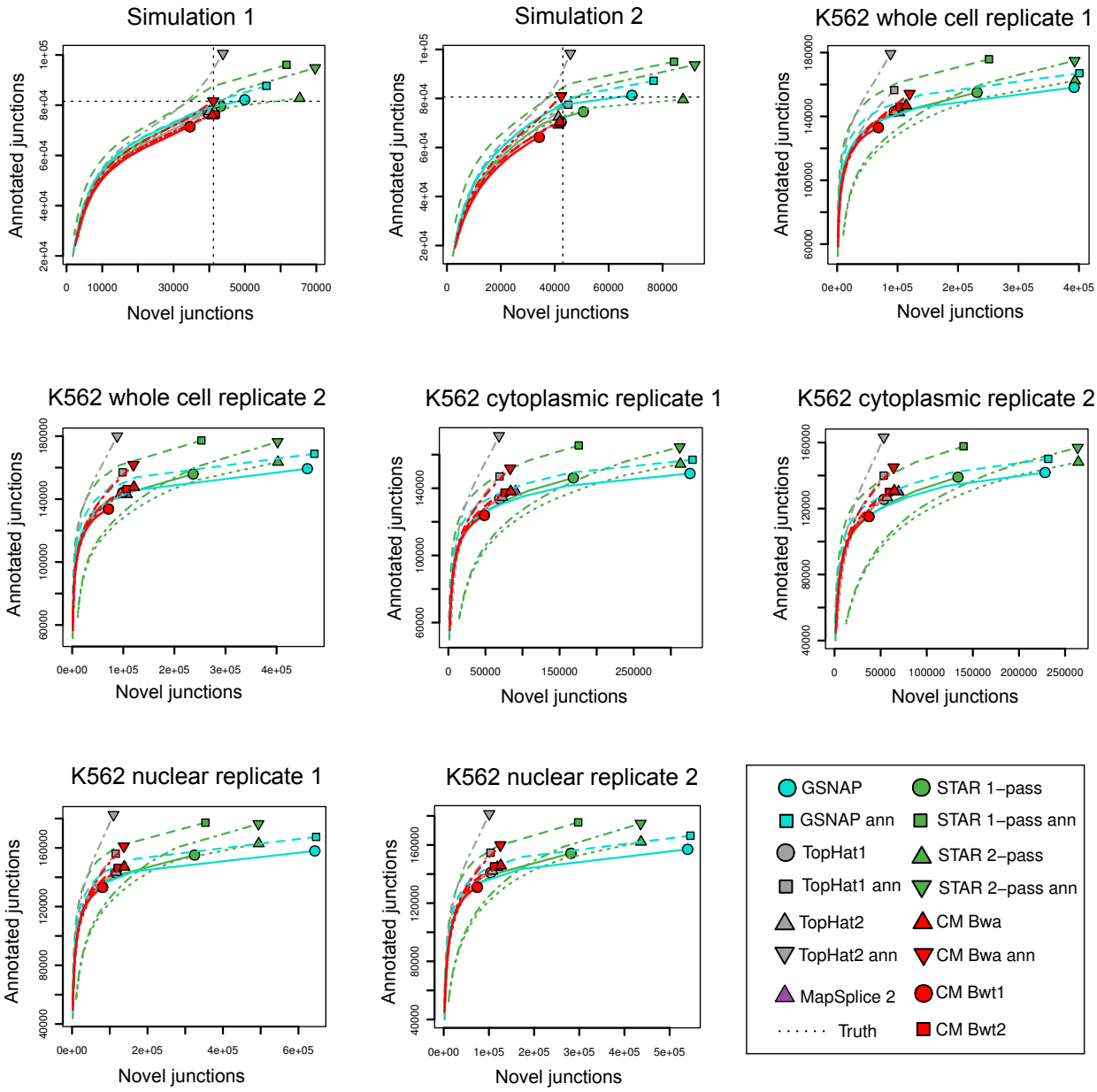

Figure A.3: Comparison of the number of annotated and novel junctions for all evaluated data sets and all evaluated RNA-seq mapping programs. To obtain receiver operation characteristic (ROC)-like curves, numbers were also calculated at increasing thresholds on the number of supporting reads for each junction.

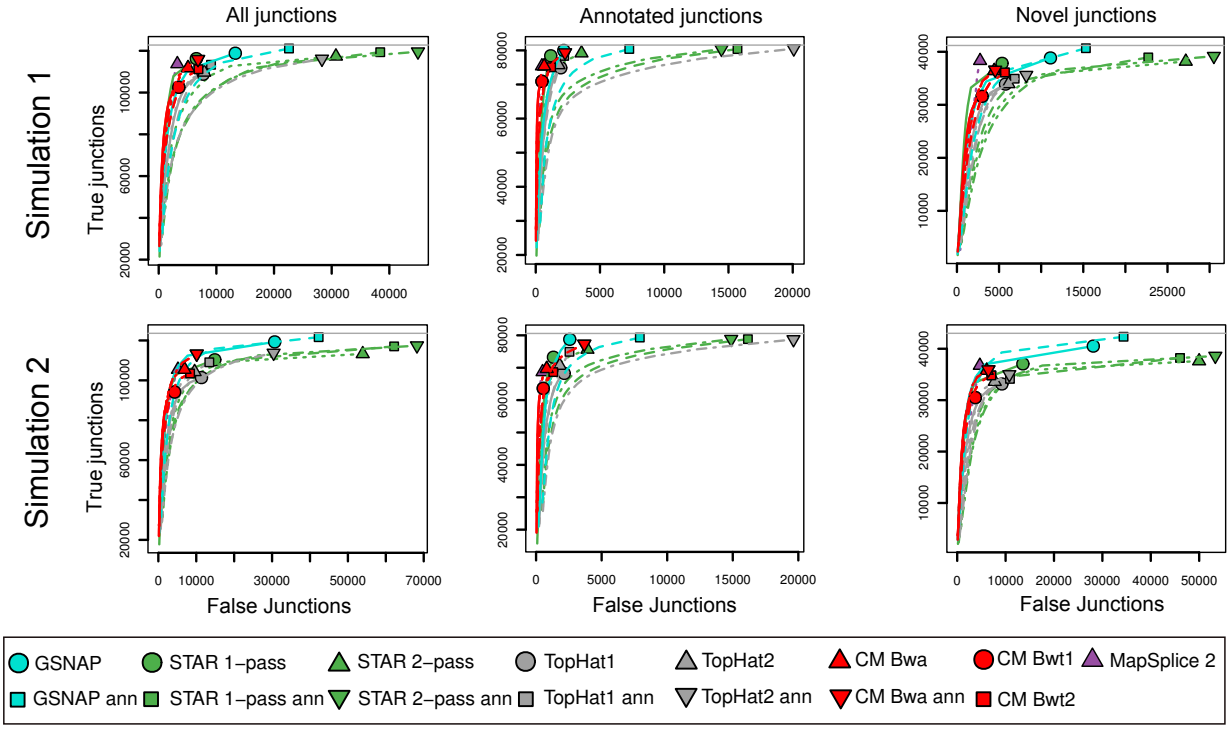

Figure A.4: Comparison of true and false junctions for all evaluated RNA-seq mapping programs. Number of correctly predicted (true) and incorrectly (false) junctions were compared for all junctions and annotated and novel junctions separately (symbols). To obtain receiver operation characteristic (ROC)-like curves, numbers were also calculated at increasing thresholds on the number of supporting reads for each junction. In contrast to the RGASP evaluation, we also included junctions covered by only 1 read.

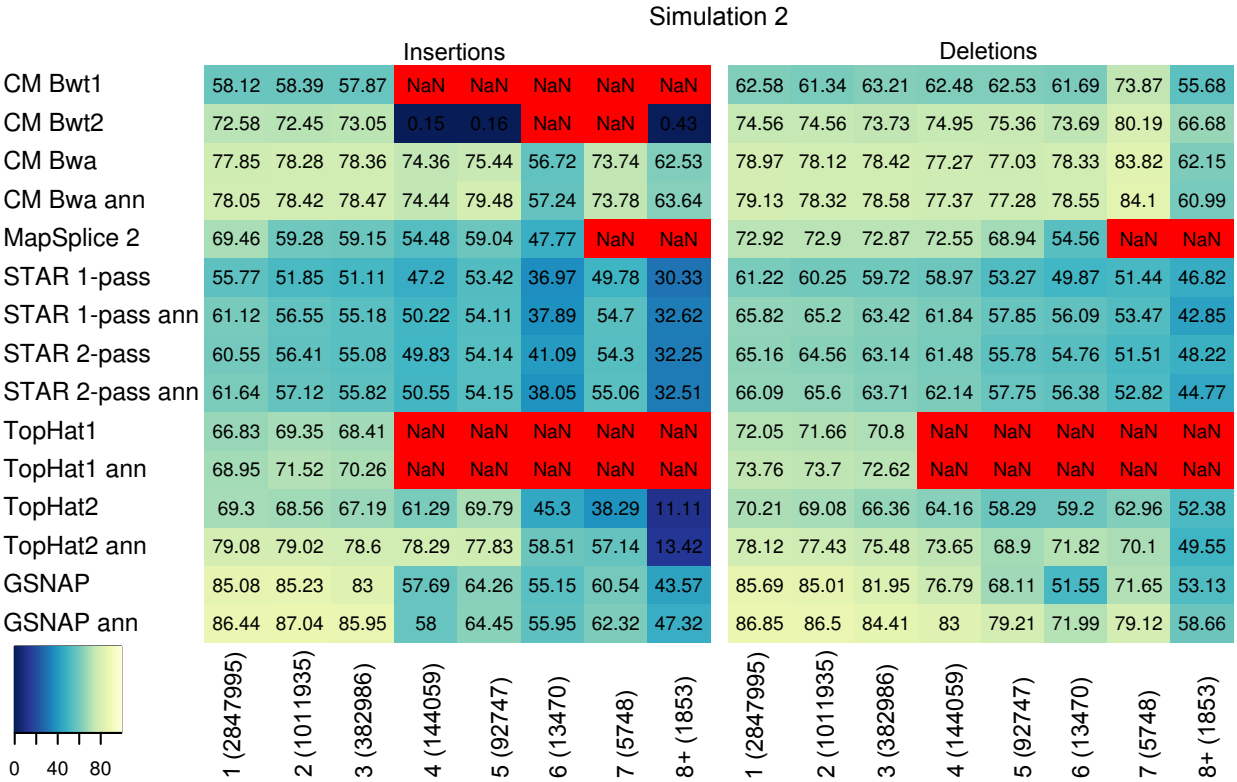

Figure A.5: F-Measure [in %] for insertion and deletion prediction for all programs on simulation 2. NaN indicates that no insertion or deletion of that size were predicted. Insertion and deletion size are shown below the column of the heatmap. The numbers in parentheses indicate the number of simulated reads for each insertion or deletion size. Recall and precision values are listed in Supplementary Tables 6 and 7.

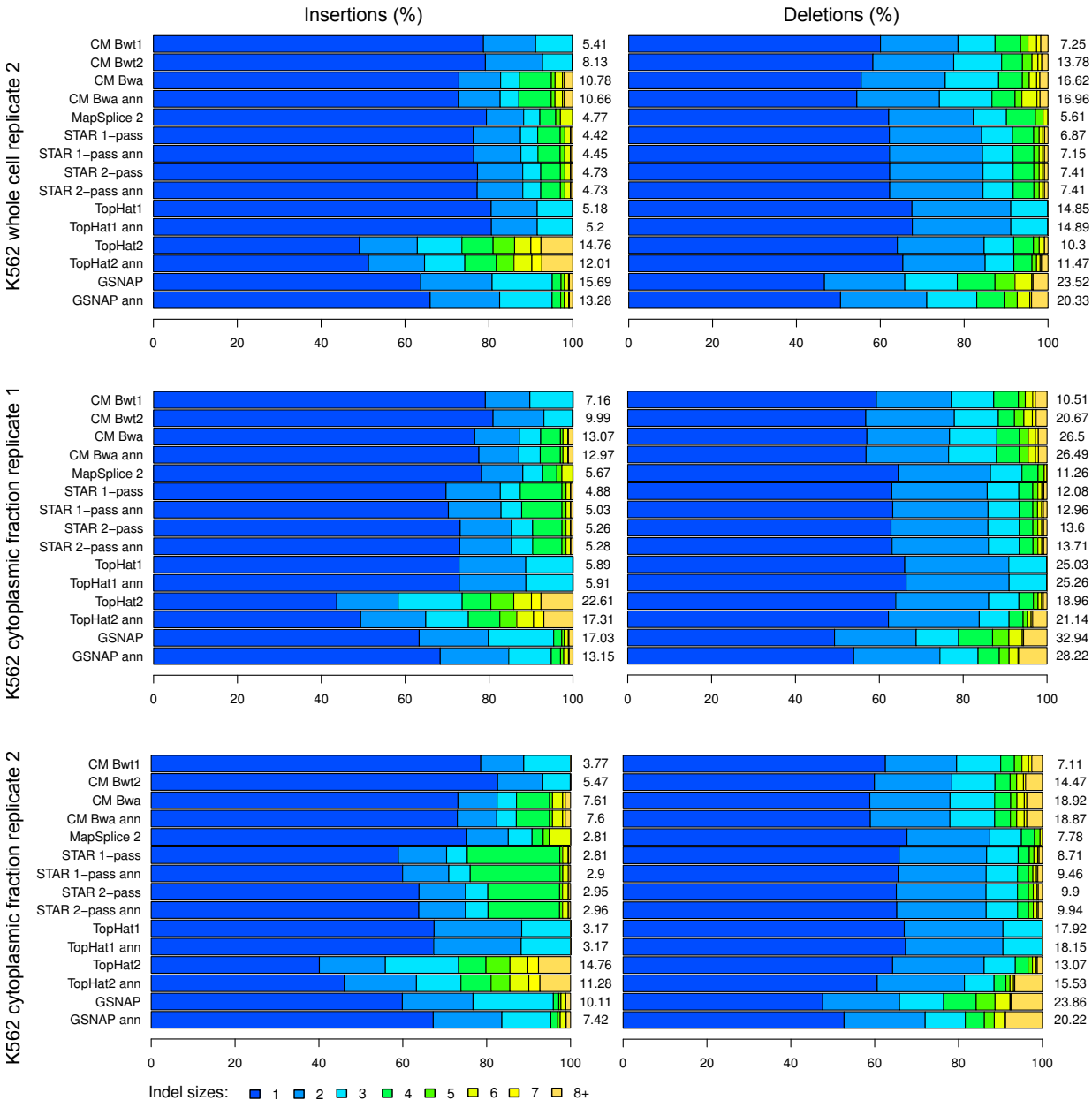

Figure A.6: Fraction of mapped reads with different indel sizes among all reads with indels for the second replicate of the K562 whole cell sample and both replicates of the K562 cytoplasmic fraction sample. Numbers next to the barplots indicate the number of mapped reads with indels divided by 10^5 (i.e. number of reads per 100,000).

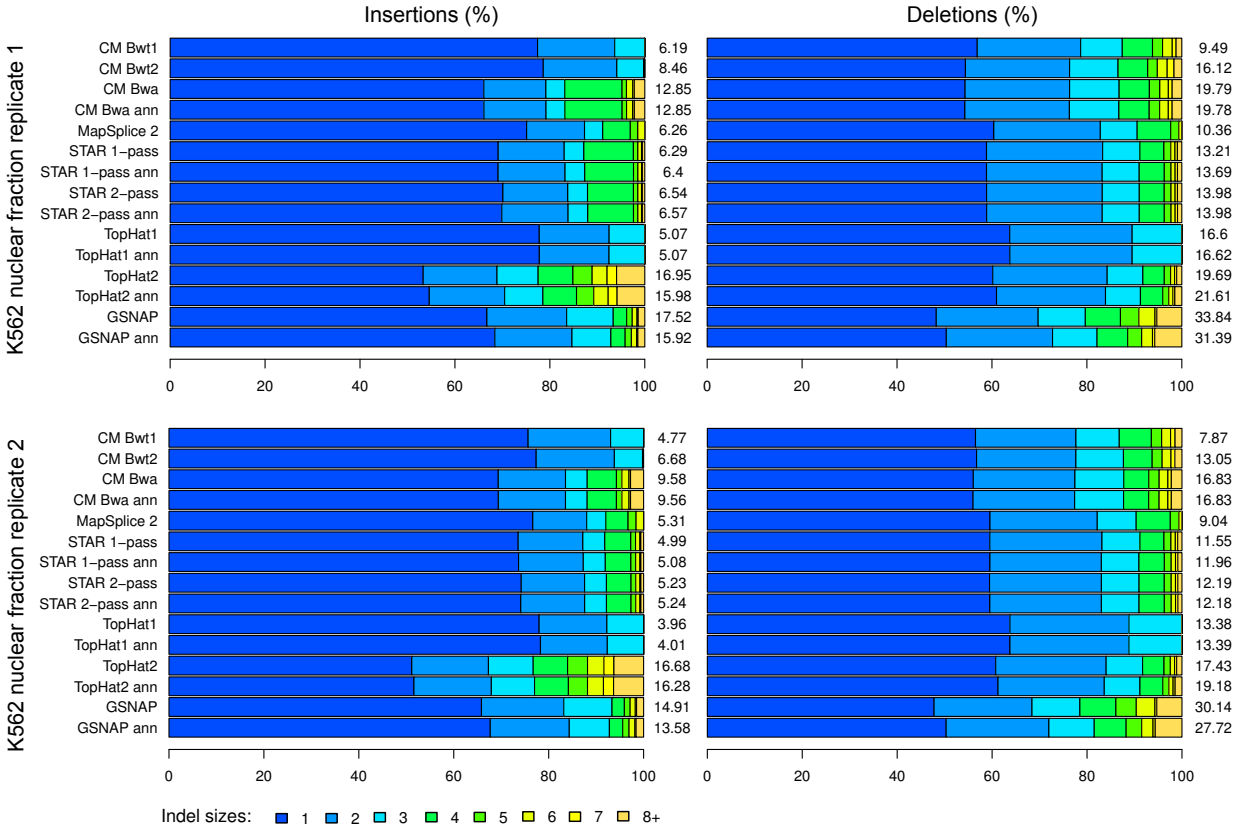

Figure A.7: Fraction of mapped reads with different indel sizes among all reads with indels for both replicates of the K562 nuclear fraction sample. Numbers next to the barplots indicate the number of mapped reads with indels divided by 10^5 (i.e. number of reads per 100,000).

Program	Uniquely mapped reads					All reads (primary alignments only)				
	Overall mapped reads	Perfectly mapped reads	Part correctly mapped	Correctly mapped bases	Incorrectly mapped bases	Overall mapped reads	Perfectly mapped reads	Part correctly mapped	Correctly mapped bases	Incorrectly mapped bases
A Simulation 1 All Reads										
CM Bwt1	89.28	87.82	0.60	88.38	0.90	89.28	87.82	0.60	88.38	0.90
CM Bwt2 ($k = 3$)	95.12	93.44	0.57	93.97	1.15	95.12	93.44	0.57	93.97	1.15
CM Bwt2 ($k = 10$)	95.78	94.45	0.58	95.00	0.78	95.78	94.45	0.58	95.00	0.78
CM Bwa	95.99	94.68	0.55	95.20	0.79	95.99	94.68	0.55	95.20	0.79
CM Bwa ann	96.24	94.95	0.54	95.46	0.78	96.24	94.95	0.54	95.46	0.78
MapSplice 2	96.54	93.47	2.49	95.52	0.61	98.89	94.78	2.52	96.86	1.61
STAR 1-pass	96.14	84.61	11.20	94.87	0.47	98.72	85.81	11.35	96.20	1.70
STAR 1-pass ann	95.56	88.83	6.60	95.06	0.16	98.81	90.64	6.94	97.19	1.27
STAR 2-pass	95.48	89.11	6.24	95.01	0.16	98.82	91.05	6.57	97.26	1.23
STAR 2-pass ann	95.29	89.00	6.16	94.84	0.15	98.81	91.08	6.52	97.26	1.25
TopHat1	93.37	90.12	1.96	92.00	1.37	95.23	90.87	2.00	92.79	2.44
TopHat1 ann	93.51	90.21	2.00	92.13	1.37	95.39	90.97	2.05	92.94	2.45
TopHat2	91.38	90.41	0.46	90.84	0.54	93.81	91.45	0.56	91.96	1.85
TopHat2 ann	92.00	91.35	0.36	91.69	0.31	94.62	92.64	0.54	93.16	1.46
GSNAP	95.65	82.99	12.39	94.68	0.37	99.23	84.90	12.66	96.84	1.75
GSNAP ann	95.72	87.04	8.62	95.30	0.07	99.24	89.07	8.82	97.52	1.35
B Simulation 2 All Reads										
CM Bwt1	80.59	78.74	0.95	79.64	0.95	80.59	78.74	0.95	79.64	0.95
CM Bwt2 ($k = 3$)	86.38	84.08	0.98	85.01	1.37	86.38	84.08	0.98	85.01	1.37
CM Bwt2 ($k = 10$)	87.39	85.30	1.00	86.25	1.13	87.39	85.30	1.00	86.25	1.13
CM Bwa	87.91	86.20	0.92	87.07	0.84	87.91	86.20	0.92	87.07	0.84
CM Bwa ann	88.28	86.58	0.91	87.44	0.83	88.28	86.58	0.91	87.44	0.83
MapSplice 2	93.53	85.58	7.03	90.77	0.94	95.93	86.97	7.13	92.23	1.84
STAR 1-pass	93.36	72.55	20.39	90.75	0.62	96.23	73.72	20.74	92.21	1.96
STAR 1-pass ann	93.33	76.53	16.55	91.66	0.36	96.71	78.10	17.11	93.73	1.60
STAR 2-pass	93.24	76.80	16.14	91.58	0.39	96.77	78.54	16.74	93.85	1.58
STAR 2-pass ann	93.08	76.85	15.98	91.51	0.35	96.77	78.67	16.61	93.90	1.59
TopHat1	83.98	80.94	2.04	82.90	1.08	86.09	81.76	2.14	83.82	2.27
TopHat1 ann	84.40	81.32	2.08	83.32	1.07	86.53	82.16	2.19	84.26	2.27
TopHat2	75.53	74.31	0.87	75.13	0.40	77.92	75.29	0.97	76.18	1.74
TopHat2 ann	77.05	75.94	0.83	76.73	0.32	79.65	77.14	1.02	78.10	1.55
GSNAP	94.28	70.94	22.95	92.48	0.47	97.95	72.54	23.45	94.55	2.01
GSNAP ann	94.31	74.66	19.48	93.17	0.19	97.97	76.35	19.91	95.27	1.70

Table A.1: Fraction [in %] of overall mapped reads, perfectly mapped reads, part correctly mapped reads (of all simulated reads) as well as fraction of correctly and incorrectly mapped bases (of all bases in all simulated reads) on both simulated data sets. Results are shown separately for uniquely mapped reads and all mapped reads. In the latter case, only the primary alignment was evaluated. "CM Bwt1", "CM Bwt2" , "CM Bwa" denote ContextMap 2 used with Bowtie, Bowtie 2, and BWA as underlying alignment program, respectively. ContextMap 2 with Bowtie 2 was run with the maximum number of alignments reported per read (k) set to 3 (default setting used for evaluating mapping quality) and 10, respectively. If a gene annotation was provided, "ann" was added to the name of the respective program.

Program	Uniquely mapped reads					All reads (primary alignments only)				
	Overall mapped reads	Perfectly mapped reads	Part correctly mapped	Correctly mapped bases	Incorrectly mapped bases	Overall mapped reads	Perfectly mapped reads	Part correctly mapped	Correctly mapped bases	Incorrectly mapped bases
A Simulation 1 Unspliced Reads										
CM Bwt1	89.99	89.19	0.11	89.29	0.70	89.99	89.19	0.11	89.29	0.70
CM Bwt2 ($k = 3$)	95.65	94.49	0.11	94.58	1.07	95.65	94.49	0.11	94.58	1.07
CM Bwt2 ($k = 10$)	96.26	95.37	0.13	95.48	0.78	96.26	95.37	0.13	95.48	0.78
CM Bwa	96.45	95.59	0.12	95.69	0.76	96.45	95.59	0.12	95.69	0.76
CM Bwa ann	96.48	95.59	0.15	95.72	0.75	96.48	95.59	0.15	95.72	0.75
MapSplice 2	96.32	95.28	0.74	95.81	0.30	99.05	96.86	0.75	97.40	1.43
STAR 1-pass	96.01	90.69	5.17	95.68	0.15	98.70	92.04	5.25	97.10	1.41
STAR 1-pass ann	95.76	90.43	5.22	95.46	0.12	98.73	92.08	5.33	97.21	1.32
STAR 2-pass	95.73	90.42	5.21	95.45	0.10	98.73	92.11	5.33	97.24	1.30
STAR 2-pass ann	95.65	90.29	5.24	95.34	0.12	98.73	92.04	5.37	97.21	1.33
TopHat1	93.76	92.87	0.13	93.00	0.77	95.77	93.70	0.14	93.83	1.94
TopHat1 ann	93.76	92.86	0.13	92.99	0.77	95.77	93.70	0.14	93.83	1.94
TopHat2	92.20	91.78	0.14	91.92	0.29	94.73	92.89	0.15	93.03	1.70
TopHat2 ann	92.19	91.67	0.19	91.85	0.34	94.65	92.85	0.25	93.08	1.57
GSNAP	95.46	87.50	7.85	95.03	0.11	99.21	89.64	8.03	97.35	1.52
GSNAP ann	95.45	87.51	7.89	95.08	0.05	99.21	89.68	8.09	97.44	1.42
B Simulation 2 Unspliced Reads										
CM Bwt1	83.86	82.70	0.38	83.06	0.80	83.86	82.70	0.38	83.06	0.80
CM Bwt2 ($k = 3$)	88.16	86.48	0.40	86.86	1.30	88.16	86.48	0.40	86.86	1.30
CM Bwt2 ($k = 10$)	89.25	87.72	0.41	88.10	1.15	89.25	87.72	0.41	88.10	1.15
CM Bwa	89.50	88.36	0.39	88.72	0.78	89.50	88.36	0.39	88.72	0.78
CM Bwa ann	89.53	88.37	0.41	88.75	0.78	89.53	88.37	0.41	88.75	0.78
MapSplice 2	94.19	89.61	4.06	92.49	0.52	97.01	91.31	4.13	94.25	1.54
STAR 1-pass	93.74	79.84	13.68	92.64	0.26	96.65	81.13	13.90	94.14	1.64
STAR 1-pass ann	93.66	79.71	13.74	92.55	0.25	96.77	81.20	14.02	94.30	1.58
STAR 2-pass	93.56	79.59	13.76	92.45	0.26	96.79	81.18	14.07	94.33	1.57
STAR 2-pass ann	93.51	79.52	13.77	92.39	0.26	96.79	81.14	14.09	94.31	1.59
TopHat1	85.55	84.41	0.59	84.97	0.58	87.77	85.29	0.63	85.89	1.88
TopHat1 ann	85.56	84.41	0.59	84.98	0.58	87.78	85.29	0.63	85.90	1.87
TopHat2	77.90	77.20	0.51	77.69	0.21	80.44	78.25	0.54	78.76	1.68
TopHat2 ann	78.01	77.15	0.55	77.68	0.33	80.41	78.22	0.62	78.80	1.61
GSNAP	94.29	75.62	18.47	93.21	0.18	98.10	77.39	18.89	95.38	1.77
GSNAP ann	94.28	75.64	18.50	93.26	0.13	98.10	77.42	18.93	95.44	1.71

Table A.2: Fraction [in %] of overall mapped reads, perfectly mapped reads, part correctly mapped reads (of all simulated *unspliced* reads) as well as fraction of correctly and incorrectly mapped bases (of all bases in all simulated *unspliced* reads) on both simulated data sets. Results are shown separately for uniquely mapped reads and all mapped reads. In the latter case, only the primary alignment was evaluated. "CM Bwt1", "CM Bwt2" , "CM Bwa" denote ContextMap 2 used with Bowtie, Bowtie 2, and BWA as underlying alignment program, respectively. ContextMap 2 with Bowtie 2 was run with the maximum number of alignments reported per read (k) set to 3 (default setting used for evaluating mapping quality) and 10, respectively. If a gene annotation was provided, "ann" was added to the name of the respective program.

Program	Uniquely mapped reads					All reads (primary alignments only)				
	Overall mapped reads	Perfectly mapped reads	Part correctly mapped	Correctly mapped bases	Incorrectly mapped bases	Overall mapped reads	Perfectly mapped reads	Part correctly mapped	Correctly mapped bases	Incorrectly mapped bases
A Simulation 1 Spliced Reads										
CM Bwt1	86.40	82.22	2.58	84.69	1.71	86.40	82.22	2.58	84.69	1.71
CM Bwt2 ($k = 3$)	92.96	89.13	2.43	91.45	1.51	92.96	89.13	2.43	91.45	1.51
CM Bwt2 ($k = 10$)	93.82	90.71	2.44	93.04	0.78	93.82	90.71	2.44	93.04	0.78
CM Bwa	94.09	90.99	2.28	93.17	0.92	94.09	90.99	2.28	93.17	0.92
CM Bwa ann	95.27	92.32	2.13	94.38	0.89	95.27	92.32	2.13	94.38	0.89
MapSplice 2	97.42	86.04	9.68	94.36	1.88	98.23	86.27	9.77	94.66	2.37
STAR 1-pass	96.68	59.73	35.86	91.57	1.79	98.81	60.31	36.32	92.53	2.88
STAR 1-pass ann	94.73	82.28	12.26	93.43	0.35	99.14	84.77	13.53	97.11	1.03
STAR 2-pass	94.46	83.70	10.47	93.22	0.41	99.18	86.72	11.66	97.34	0.95
STAR 2-pass ann	93.82	83.73	9.90	92.80	0.29	99.16	87.13	11.23	97.46	0.93
TopHat1	91.77	78.88	9.43	87.93	3.84	93.03	79.29	9.64	88.53	4.50
TopHat1 ann	92.48	79.36	9.66	88.63	3.85	93.81	79.82	9.89	89.30	4.51
TopHat2	88.01	84.78	1.78	86.42	1.59	90.02	85.57	2.23	87.56	2.46
TopHat2 ann	91.24	90.04	1.08	91.06	0.18	94.51	91.82	1.75	93.47	1.04
GSNAP	96.44	64.51	31.00	93.21	1.44	99.31	65.51	31.61	94.78	2.68
GSNAP ann	96.82	85.14	11.58	96.20	0.18	99.36	86.57	11.81	97.84	1.07
B Simulation 2 Spliced Reads										
CM Bwt1	67.13	62.39	3.31	65.56	1.57	67.13	62.39	3.31	65.56	1.57
CM Bwt2 ($k = 3$)	79.05	74.17	3.38	77.39	1.65	79.05	74.17	3.38	77.39	1.65
CM Bwt2 ($k = 10$)	79.70	75.35	3.42	78.61	1.09	79.70	75.35	3.42	78.61	1.09
CM Bwa	81.33	77.29	3.11	80.27	1.06	81.33	77.29	3.11	80.27	1.06
CM Bwa ann	83.11	79.19	2.97	82.05	1.05	83.11	79.19	2.97	82.05	1.05
MapSplice 2	90.80	68.99	19.29	83.68	2.67	91.47	69.07	19.48	83.91	3.05
STAR 1-pass	91.80	42.50	48.05	82.96	2.14	94.50	43.11	48.93	84.26	3.30
STAR 1-pass ann	91.98	63.42	28.13	87.96	0.80	96.47	65.30	29.87	91.37	1.69
STAR 2-pass	91.89	65.27	25.99	87.96	0.95	96.70	67.61	27.80	91.90	1.62
STAR 2-pass ann	91.34	65.81	25.10	87.89	0.73	96.71	68.46	26.98	92.22	1.59
TopHat1	77.46	66.62	8.04	74.35	3.11	79.17	67.23	8.38	75.26	3.91
TopHat1 ann	79.59	68.59	8.23	76.49	3.10	81.39	69.26	8.60	77.50	3.90
TopHat2	65.76	62.38	2.35	64.57	1.19	67.56	63.07	2.73	65.54	2.02
TopHat2 ann	73.10	70.93	1.99	72.81	0.29	76.50	72.68	2.67	75.18	1.32
GSNAP	94.22	51.61	41.43	89.48	1.66	97.36	52.53	42.26	91.15	3.02
GSNAP ann	94.43	70.60	23.51	92.82	0.44	97.45	71.93	23.97	94.59	1.65

Table A.3: Fraction [in %] of overall mapped reads, perfectly mapped reads, part correctly mapped reads (of all simulated *spliced* reads) as well as fraction of correctly and incorrectly mapped bases (of all bases in all simulated *spliced* reads) on both simulated data sets. Results are shown separately for uniquely mapped reads and all mapped reads. In the latter case, only the primary alignment was evaluated. "CM Bwt1", "CM Bwt2" , "CM Bwa" denote ContextMap 2 used with Bowtie, Bowtie 2, and BWA as underlying alignment program, respectively. ContextMap 2 with Bowtie 2 was run with the maximum number of alignments reported per read (k) set to 3 (default setting used for evaluating mapping quality) and 10, respectively. If a gene annotation was provided, "ann" was added to the name of the respective program.

Program	Recall for simulated reads spanning different numbers of junctions					Precision for simulated reads spanning different numbers of junctions				
	1 (13808336)	2 (598297)	3 (11781)	2* (548382)	3* (6908)	1	2	3	2*	3*
A Simulation 1										
CM Bwt1	85.64	7.72	0.0	8.26	0.0	98.15	92.09	-	92.14	-
CM Bwt2	90.3	65.87	33.6	71.56	57.3	98.07	97.04	99.3	97.03	99.3
CM Bwa	91.94	72.05	36.44	78.27	62.13	98.34	97.12	99.4	97.09	99.37
CM Bwa ann	93.27	74.89	37.07	81.19	62.97	98.35	97.34	98.76	97.18	98.37
MapSplice 2	87.62	66.53	15.89	68.58	1.75	97.78	97.78	96.15	97.88	98.37
STAR 1-pass	65.28	18.47	2.58	19.18	0.75	95.74	94.68	87.86	94.89	94.55
STAR 1-pass ann	90.54	71.62	65.19	73.06	72.05	96.77	94.95	90.29	94.93	95.6
STAR 2-pass	92.41	76.57	71.54	78.23	78.58	97.74	96.91	96.23	96.94	98.19
STAR 2-pass ann	92.88	78.15	73.25	79.49	78.42	97.36	96.28	94.57	96.12	96.41
TopHat1	79.94	67.5	49.45	72.01	66.24	97.44	91.0	88.99	91.05	87.92
TopHat1 ann	80.41	70.21	55.28	73.63	67.49	97.23	90.27	88.77	90.16	87.22
TopHat2	86.13	78.64	63.27	83.07	80.2	98.06	97.35	98.07	97.39	98.75
TopHat2 ann	92.27	87.42	80.47	89.59	88.26	97.55	94.44	92.16	94.65	92.55
GSNAP	73.31	28.39	10.86	27.27	6.56	95.99	92.54	63.04	94.66	98.26
GSNAP ann	94.87	82.17	68.42	83.78	74.22	98.12	96.09	94.88	96.9	98.54
B Simulation 2										
	1 (14962090)	2 (622980)	3 (11701)	2* (509939)	3* (5602)	1	2	3	2*	3*
CM Bwt1	65.27	5.0	0.0	5.94	0.0	97.54	89.21	-	89.16	-
CM Bwt2	75.74	50.33	19.63	60.99	41.0	97.17	95.92	99.39	95.51	99.39
CM Bwa	78.63	57.26	20.96	69.35	43.75	97.77	96.28	98.71	95.79	98.63
CM Bwa ann	80.73	59.85	21.49	72.08	44.59	97.75	96.59	99.02	95.66	98.39
MapSplice 2	72.71	45.76	12.89	51.51	0.71	96.33	94.97	97.42	94.83	97.56
STAR 1-pass	51.7	11.13	0.63	12.73	0.34	94.78	93.35	89.16	92.83	70.37
STAR 1-pass ann	79.05	53.75	44.14	55.96	50.34	95.13	90.88	90.23	89.59	93.28
STAR 2-pass	81.47	59.59	47.77	63.35	56.19	95.87	92.55	92.05	91.45	94.11
STAR 2-pass ann	82.74	61.49	48.43	64.08	56.16	95.67	91.54	89.06	90.12	92.15
TopHat1	68.33	55.08	33.33	65.21	61.53	96.57	89.63	87.96	89.59	92.12
TopHat1 ann	70.45	58.86	44.22	67.64	63.39	96.58	88.91	88.37	88.47	89.72
TopHat2	64.11	56.92	38.39	65.37	60.55	96.96	95.63	95.37	95.62	95.36
TopHat2 ann	73.8	68.66	64.37	71.27	73.97	96.37	91.22	88.94	91.42	91.34
GSNAP	68.15	23.82	8.97	22.19	3.86	95.39	86.77	82.22	93.58	96.0
GSNAP ann	90.99	76.41	56.16	78.55	60.96	97.41	92.81	88.85	94.33	96.09

Table A.4: Recall and precision [in %] for spliced reads with different number of spanned junctions for simulation 1 and 2. Columns marked with an asterisk show results only for reads for which all exons except the first and last exon had length $\geq$ 20 nt.

Program	Recall for individual insertion lengths								Precision for individual insertion lengths							
	1	2	3	4	5	6	7	8	1	2	3	4	5	6	7	8
Simulation 1 Insertions																
CM Bwt1	56.6	60.0	57.0	0.0	0.0	0.0	0.0	0.0	96.9	97.7	93.7	-	-	-	-	-
CM Bwt2	72.0	77.8	74.1	1.6	0.5	0.0	0.0	0.0	97.0	98.2	93.3	69.6	30.7	0.0	-	0.0
CM Bwa	78.9	83.5	82.7	69.9	70.3	60.1	73.4	64.1	97.9	98.3	95.8	92.0	92.5	92.4	93.7	37.0
CM Bwa ann	79.2	83.7	83.0	69.9	70.3	60.1	73.4	64.1	98.0	98.4	96.0	92.0	92.4	92.4	93.6	37.2
MapSplice 2	73.7	60.4	59.8	55.1	50.7	39.7	0.0	0.0	97.4	98.2	97.0	93.1	94.8	90.6	-	-
STAR 1-pass	49.8	49.5	48.0	45.6	40.4	26.8	40.8	31.5	97.6	98.4	97.1	96.2	97.5	99.1	98.5	95.6
STAR 1-pass ann	55.8	54.5	51.7	47.4	51.9	28.5	41.0	41.8	97.9	98.3	96.8	96.1	96.5	99.1	98.6	96.6
STAR 2-pass	54.4	53.8	51.2	47.3	51.3	27.7	41.1	41.8	97.8	98.4	97.2	96.3	97.9	82.7	98.5	91.9
STAR 2-pass ann	56.1	54.9	51.9	47.8	51.9	28.7	41.1	41.8	97.9	98.3	96.8	96.1	97.9	99.1	98.5	96.6
TopHat1	65.9	72.7	72.2	0.0	0.0	0.0	0.0	0.0	94.8	96.2	90.3	-	-	-	-	-
TopHat1 ann	68.8	76.0	74.9	0.0	0.0	0.0	0.0	0.0	94.9	96.4	90.7	-	-	-	-	-
TopHat2	70.5	72.5	69.6	65.9	64.0	55.0	70.0	52.4	88.0	89.7	76.8	63.5	38.2	18.3	21.4	1.1
TopHat2 ann	84.8	86.0	84.8	82.7	86.8	85.9	86.4	90.1	94.0	94.1	85.9	81.3	65.2	41.5	43.5	4.2
GSNAP	85.2	87.2	85.8	53.9	54.2	49.6	54.7	54.6	91.1	88.1	67.8	87.9	93.7	62.7	86.2	18.0
GSNAP ann	87.0	88.5	87.2	53.9	54.9	50.7	55.3	54.9	95.5	95.7	82.7	89.2	92.9	67.7	93.9	52.1

Simulation 1 Deletions	Recall for individual deletion lengths								Precision for individual deletion lengths							
	1	2	3	4	5	6	7	8	1	2	3	4	5	6	7	8
CM Bwt1	61.2	60.1	58.0	54.1	69.7	43.5	51.1	50.3	97.7	97.1	91.8	95.3	92.2	60.8	92.4	84.9
CM Bwt2	74.8	73.3	73.3	64.1	80.6	68.3	52.9	61.0	97.7	97.3	93.9	87.2	93.4	85.0	81.4	93.1
CM Bwa	78.9	76.7	79.1	67.5	84.8	73.6	51.3	66.2	98.3	97.9	94.9	95.0	93.0	77.8	82.8	79.4
CM Bwa ann	79.3	77.2	79.4	69.4	85.0	73.6	51.3	67.2	98.3	98.1	95.0	96.4	93.1	77.9	92.2	79.6
MapSplice 2	77.9	76.0	79.9	68.0	83.3	63.7	0.0	0.0	98.4	98.7	94.9	96.4	99.3	99.9	-	-
STAR 1-pass	57.1	56.2	59.6	47.0	54.9	43.1	28.7	29.9	97.9	97.3	94.9	96.2	95.3	96.2	91.8	66.4
STAR 1-pass ann	64.5	62.5	65.0	53.7	57.9	43.8	29.1	30.8	98.4	96.5	96.5	92.3	94.6	97.8	96.9	55.7
STAR 2-pass	63.2	60.2	64.0	52.5	55.9	43.8	29.1	30.2	98.6	97.6	96.6	96.2	95.3	97.6	92.0	55.6
STAR 2-pass ann	64.9	62.8	65.7	54.2	57.9	43.8	29.1	30.8	98.5	96.6	96.6	92.3	94.6	97.8	96.9	55.7
TopHat1	69.7	69.2	71.0	0.0	0.0	0.0	0.0	0.0	95.3	96.2	92.6	-	-	-	-	-
TopHat1 ann	72.6	72.6	73.7	0.0	0.0	0.0	0.0	0.0	95.4	96.4	92.8	-	-	-	-	-
TopHat2	70.4	68.6	67.4	57.0	65.3	43.4	53.8	31.9	93.1	95.6	92.8	93.6	89.8	84.0	87.0	83.8
TopHat2 ann	84.0	82.0	81.3	73.6	81.5	66.9	72.9	37.9	95.7	96.3	94.2	91.6	93.5	77.7	93.8	64.3
GSNAP	86.5	83.8	85.9	78.9	81.9	76.2	68.9	70.3	92.4	87.7	71.0	47.7	24.7	9.8	74.3	57.0
GSNAP ann	87.7	85.8	87.0	82.7	83.3	78.3	84.6	74.1	96.2	93.9	87.6	85.3	66.8	48.1	89.7	63.6

Table A.5: Recall and precision for insertions and deletions in simulation 1.

Program	Recall for individual insertion lengths								Precision for individual insertion lengths							
	1	2	3	4	5	6	7	8	1	2	3	4	5	6	7	8
Simulation 2 Insertions																
CM Bwt1	41.7	41.9	41.9	0.0	0.0	0.0	0.0	0.0	95.8	96.2	93.6	-	-	-	-	-
CM Bwt2	58.3	58.1	59.7	0.1	0.1	0.0	0.0	0.2	96.2	96.3	94.2	63.7	94.9	0.0	0.0	44.4
CM Bwa	65.0	65.5	66.6	61.2	62.1	43.6	64.6	61.8	97.1	97.3	95.1	94.7	96.1	81.1	85.8	63.2
CM Bwa ann	65.2	65.7	66.8	61.4	67.6	44.2	64.7	61.8	97.1	97.3	95.1	94.6	96.4	81.3	85.8	65.6
MapSplice 2	54.6	42.8	42.7	38.3	42.9	33.2	0.0	0.0	95.5	96.6	96.0	94.4	94.5	85.3	-	-
STAR 1-pass	39.4	35.8	35.2	31.8	37.5	26.1	33.9	20.8	95.3	93.8	93.0	91.6	92.8	63.1	93.6	55.8
STAR 1-pass ann	44.9	40.4	39.3	34.6	38.3	28.7	38.6	23.1	95.8	94.1	92.4	91.7	92.4	55.6	93.7	55.6
STAR 2-pass	44.3	40.2	39.2	34.2	38.3	31.2	38.2	23.0	95.8	94.3	92.5	91.9	92.3	60.2	93.7	53.8
STAR 2-pass ann	45.4	41.0	40.0	34.9	38.3	28.9	39.0	23.1	95.9	94.1	92.5	91.7	92.3	55.7	93.7	54.9
TopHat1	52.2	54.6	54.0	0.0	0.0	0.0	0.0	0.0	93.0	94.9	93.3	-	-	-	-	-
TopHat1 ann	54.7	57.3	56.3	0.0	0.0	0.0	0.0	0.0	93.3	95.1	93.5	-	-	-	-	-
TopHat2	55.4	54.5	53.9	48.2	58.9	47.4	48.2	49.1	92.5	92.4	89.2	84.3	85.7	43.3	31.8	6.3
TopHat2 ann	67.9	67.8	68.2	69.6	67.9	62.1	67.6	61.1	94.6	94.7	92.7	89.4	91.2	55.3	49.5	7.5
GSNAP	78.3	78.8	78.9	41.7	48.2	43.1	47.3	38.5	93.2	92.8	87.5	93.8	96.5	76.5	84.2	50.1
GSNAP ann	80.0	80.9	81.4	42.1	48.3	43.7	47.7	38.1	94.1	94.1	91.0	93.1	96.8	77.8	90.0	62.6

Simulation 2 Deletions	Recall for individual deletion lengths								Precision for individual deletion lengths							
	1	2	3	4	5	6	7	8	1	2	3	4	5	6	7	8
CM Bwt1	46.2	45.5	47.9	46.5	47.9	48.3	61.0	44.5	96.7	94.3	93.0	95.3	90.0	85.2	93.5	74.3
CM Bwt2	60.9	61.3	61.2	62.3	64.4	63.7	73.0	58.0	96.2	95.1	92.7	94.1	90.9	87.5	88.9	78.4
CM Bwa	66.4	66.0	67.1	64.7	66.9	70.5	76.6	59.9	97.4	95.7	94.3	95.8	90.7	88.2	92.6	64.6
CM Bwa ann	66.7	66.3	67.3	64.9	67.3	70.8	77.1	59.9	97.4	95.7	94.4	95.7	90.8	88.1	92.5	62.1
MapSplice 2	58.5	58.5	58.8	58.1	55.2	39.0	0.0	0.0	96.8	96.6	95.8	96.6	91.8	90.6	-	-
STAR 1-pass	45.0	44.3	44.3	43.3	38.3	34.6	36.0	31.9	95.7	94.3	91.5	92.5	87.3	89.2	90.1	88.3
STAR 1-pass ann	50.1	49.9	48.5	47.0	43.1	41.2	38.0	33.2	96.0	94.1	91.7	90.2	87.9	87.6	90.4	60.5
STAR 2-pass	49.3	49.1	48.1	46.4	40.9	39.6	36.1	33.2	96.1	94.2	91.7	91.1	87.8	88.7	90.0	87.9
STAR 2-pass ann	50.4	50.3	48.8	47.3	43.0	41.5	37.3	33.3	96.1	94.1	91.8	90.4	87.8	87.7	90.3	68.2
TopHat1	58.1	58.3	56.9	0.0	0.0	0.0	0.0	0.0	94.8	93.1	93.5	-	-	-	-	-
TopHat1 ann	60.4	60.9	59.3	0.0	0.0	0.0	0.0	0.0	94.8	93.3	93.7	-	-	-	-	-
TopHat2	56.1	54.6	51.2	48.8	42.8	43.7	46.9	37.9	93.8	94.1	94.2	93.7	91.2	91.8	95.7	85.0
TopHat2 ann	66.5	65.4	62.9	61.1	55.3	59.0	55.3	46.5	94.7	95.0	94.5	92.8	91.3	91.7	95.8	53.1
GSNAP	79.1	79.4	78.2	74.6	72.3	76.3	60.6	61.2	93.4	91.5	86.1	79.1	64.4	38.9	87.5	47.0
GSNAP ann	80.6	81.0	80.0	78.3	75.9	79.3	72.1	71.4	94.1	92.8	89.3	88.2	82.8	65.9	87.7	49.8

Table A.6: Recall and precision for insertions and deletions in simulation 2.

Appendix B

Supplementary Material for chapter 4

In this chapter, we present the Supplementary Material for chapter 4. All figures and tables shown here were taken from the Supplementary Material of an article that was published at PLoS ONE in 2013 (Bonfert et al. [2013]). I moved parts of the original Supplementary Material to chapter 4, removed some figures and slightly modified the layout of the tables.

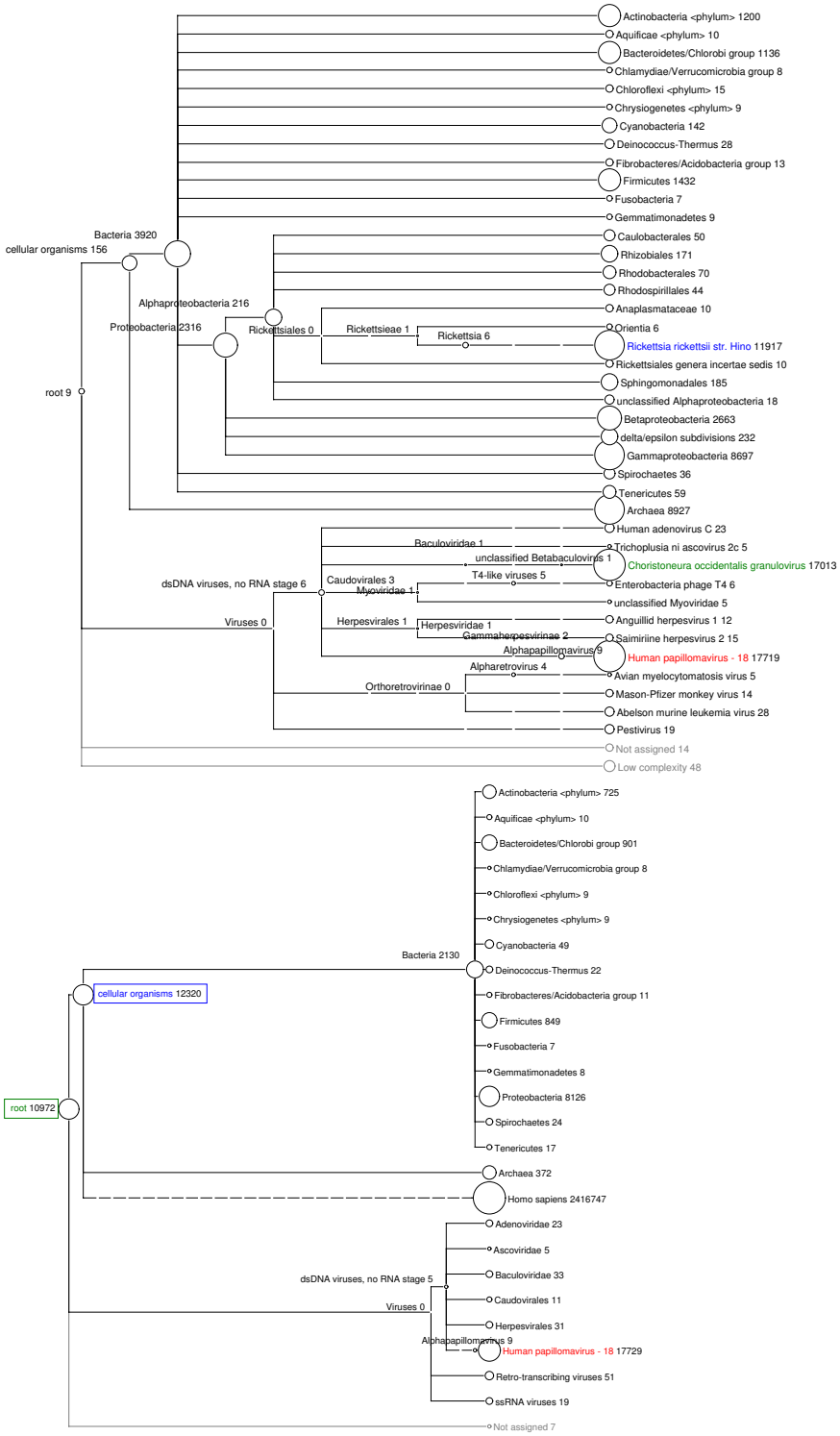

Figure B.1: Phylogenetic tree of the species identified by MEGAN4 for the miR-155 transfected HeLa cells. Assigned read numbers are annotated next to the species name and node size is proportional to the number of reads assigned to the node. On the top, results are shown for megablast runs only against microbial and viral genomes, on the bottom results including also rRNA and mitochondrial genome sequences. HPV-18 is indicated in red. Blue and green indicates reads mapped to one bacterial and viral species, respectively, if human sequences are not also used for mapping. If they are used, these reads are assigned to the node "cellular organism" and the root, respectively.

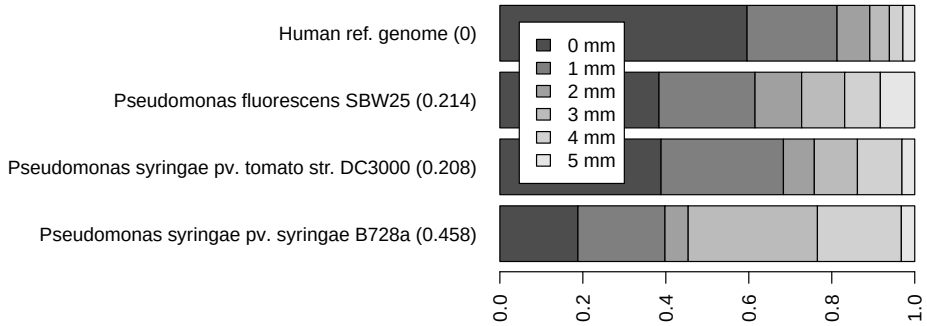

Figure B.2: Average mismatch (mm) distributions across all tumor and normal tissue samples in the colorectal carcinoma data set for the three *Pseudomonas* strains. Distributions are compared against the average mismatch distribution for the human reference genome. Numbers in parentheses indicate the divergence ($\sqrt{D_{JS}}$) of the mismatch distribution from the reference genome.

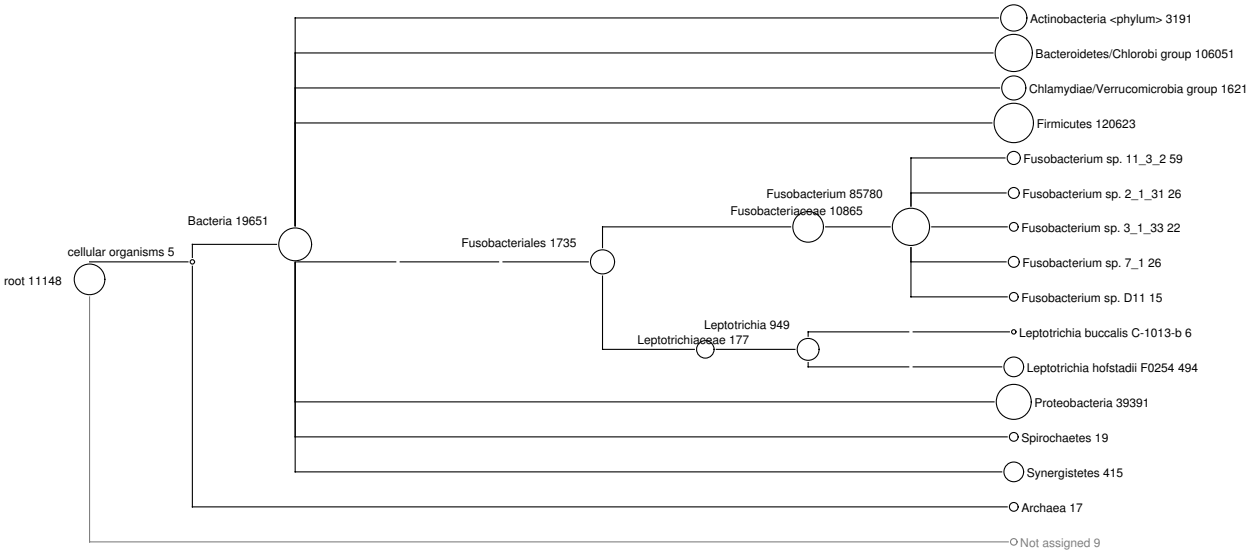

Figure B.3: Phylogenetic tree of the species identified by MEGAN4 on the colorectal carcinoma samples for patient 1 after aligning with megablast against viral and microbial genomes and the human microbiome. Only reads were used that were not mapped to human sequences by ContextMap. Assigned read numbers are annotated next to the species name and node size is proportional to the number of reads assigned to the node.

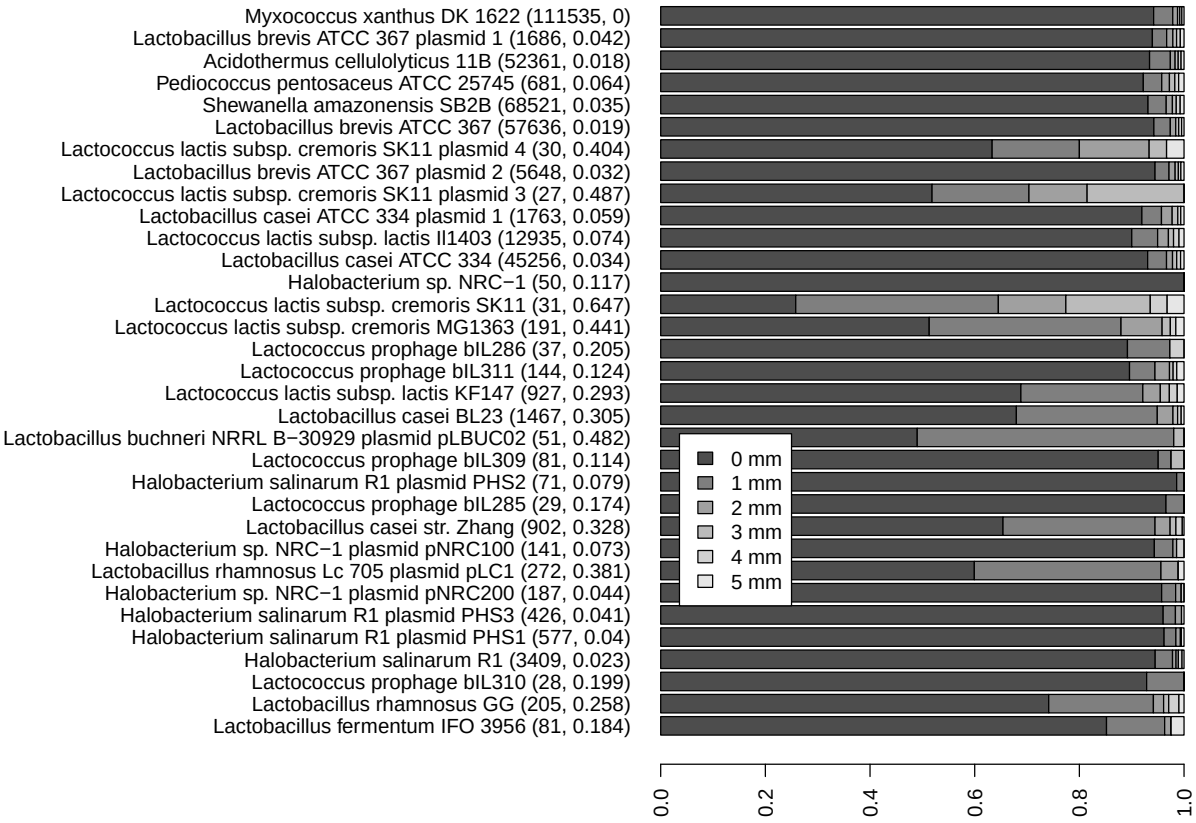

Figure B.4: Average mismatch (mm) distributions for the microbe and virus hits identified by ContextMap on the microbial community data set. Results are shown for species with coverage $> 10^{-5}$ and at least 20 reads. Numbers in parentheses indicate the number of reads mapped to the species and the divergence ($\sqrt{D_{JS}}$) of the mismatch distribution from the reference genome, in this case *Myxococcus xanthus DK 1622*.

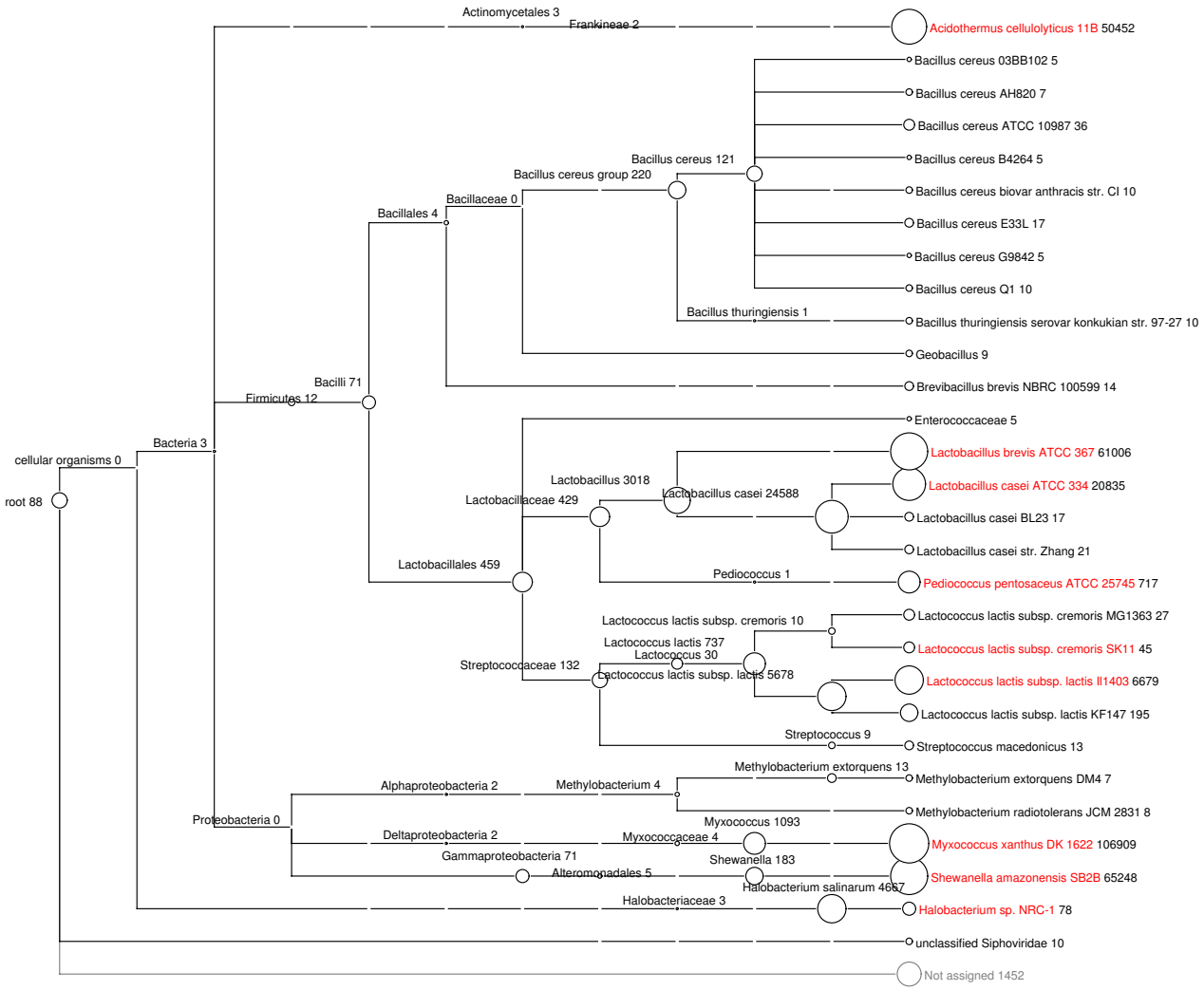

Figure B.5: Phylogenetic tree of the species identified by MEGAN4 for the *in-vitro* simulated microbial community. Assigned read numbers are annotated next to the species name and node size is proportional to the number of reads assigned to the node. Species contained in the sample are colored in red.

species	DB	#reads tumor	$\sqrt{D_{JS}}$ tumor	#reads normal	$\sqrt{D_{JS}}$ normal	enrichment
Escherichia coli 83972	HMP	6707	0.1164	0	NA	1526.07
Escherichia coli MS 200-1	HMP	3844	0.1103	3	0.4135	546.95
Escherichia coli MS 187-1	HMP	1857	0.0835	0	NA	423.35
Selenomonas sputigena ATCC 35185	RS	13586	0.1258	39	0.2827	351.15
ADGF01000000	HMP	1085	0.0347	0	NA	247.83
Fusobacterium sp. 11_3_2	HMP	24678	0.1053	126	0.1255	214.20
Fusobacterium sp. 3_1_33	HMP	50588	0.0730	307	0.1343	184.34
ACAC01000000	HMP	3098	0.1023	25	0.1169	117.59
Bacteroides sp. 2_1	HMP	8436	0.1461	128	0.4059	72.15
ACID01000000	HMP	1169	0.0841	17	0.3428	60.67
Fusobacterium sp. D11	HMP	11375	0.1422	222	0.1167	56.99
Escherichia coli MS 45-1	HMP	1777	0.0506	38	0.0469	47.11
Escherichia coli MS 21-1	HMP	2028	0.1151	54	0.1480	39.17
Granulicatella adiacens ATCC 49175	HMP	3475	0.1031	120	0.0656	31.65
Bacteroides fragilis YCH46	RS	24448	0.1415	927	0.1620	29.83
Bacteroides fragilis NCTC 9343	RS	3605	0.1132	172	0.1178	23.19
Bilophila sp. 4_1_30	HMP	2360	0.1065	123	0.0689	21.00
Bilophila wadsworthia 3_1_6	HMP	1753	0.1255	94	0.2020	20.19
Bacteroides fragilis 638R	RS	7508	0.1325	497	0.1965	17.01
Gemella morbillorum M424	HMP	2130	0.0935	144	0.0869	16.29
Clostridium asparagiforme DSM 15981	HMP	6062	0.0949	424	0.4643	16.08
Solobacterium moorei F0204	HMP	1015	0.1098	88	0.0678	12.47
ACAA01000000	HMP	5239	0.0889	643	0.1148	9.20
Peptostreptococcus stomatis DSM 17678	HMP	1877	0.1233	423	0.0790	5.00
Collinsella aerofaciens ATCC 25986	HMP	1008	0.1308	352	0.1924	3.23
Desulfovibrio piger ATCC 29098	HMP	1253	0.1457	936	0.2271	1.52

Table B.1: Species identified by ContextMap in RNA-seq data of tumor and normal tissue for patient 1 from the colorectal carcinoma data set. This table shows results for all species with at least 1000 mapped reads in at least one sample and $\sqrt{D_{JS}} \leq 0.15$ in the tumor tissue. The second column indicates the database from which the genome sequence was obtained: RS=RefSeq and HMP=the Human Microbiome Project. The last column indicates the enrichment of the particular species in the tumor sample compared to the normal tissue. Only hits with an enrichment ≥ 1 are shown. For calculating enrichment, read numbers were first divided by the number of reads mapped to any species in the corresponding sample. Furthermore, a pseudocount of 5 was used for each sample to address the problem of 0 reads in one sample. This means that enrichment is calculated as $\frac{(\#\text{ species reads in tumor}+5)\cdot(\#\text{ mapped reads for normal tissue})}{(\#\text{ mapped reads for tumor})\cdot(\#\text{ species reads in normal tissue}+5)}$.

Program	# Reads	Reference Set	Max Virtual memory [MB]	Max Resident Set Size [MB]	Real Time [min]	User CPU Time [min]
Simulated microbial community						
ContextMap	484,629	microbes, viruses, yeast	5295	2489	9	29
GASiC	484,629	genus	3018	2821	298	905
GRAMMy	484,629	genus	588	331	443	443
BLAST (megablast)	484,629	microbes	2568	2440	43	43
MetaPhyler	484,629	microbes	3538	3478	1	2
SOrt-ITEMS	484,629	microbes	2975	2915	1594	677
MARTA	484,629	microbes	6845	4870	2207	13164
MLTreeMap	484,629	microbes	287	224	2659	2636
ClaMS	484,629	microbes	47153	34050	138	275
Phymm/PhymmBL	484,629	microbes	35865	35819	5666	5528
RNA-seq of colorectal carcinoma (Patient 1)						
ContextMap	5,343,842	rDNA, hg19, microbes, hm, viruses	16480	10641	276	1771
BLAST (megablast)	404,234	microbes, hm, viruses	29766	28612	1301	1288
Novoalign	404,234	microbes, hm, viruses	15223	15127	42	38
RNA-seq of HeLa-cells (miR-155 set)						
ContextMap	29,595,334	rRNA, hg19, microbes, viruses	25077	15957	1358	8336
BLAST (megablast)	29,595,334	rRNA, mtDNA, microbes, viruses	2495	2382	497	493

hg19 = human reference genome, hm = human microbiome

Table B.2: Runtime and memory requirements of ContextMap and all evaluated tools on all three data sets (sorted according to data set size). The third column indicates the reference set. Here, 'genus' indicates that only genomes from the same genus as the microbes in the simulated microbial community were used. Both the maximum virtual memory and the resident set size (portion of a process's memory held in RAM) are shown. For programs implemented in Java (ContextMap, MARTA, ClaMS) the latter is more informative, as Java will allocate the amount of memory provided by the -Xmx option regardless of whether it needs it or not. For runtime both real and user CPU time were determined using the unix program 'time' and rounded to minutes.
The GRAMMy and GASiC runs on all species were aborted after 48 hours without results. Thus, only the results for a mapping against the 'genus' set is shown. Runtime and memory to obtain the BLAST input for GRAMMy are not included in its runtime and memory. MG-RAST and PhyloPhytiaS are provided as web servers and thus could not be evaluated.
For BLAST-based approaches that perform analysis individually for each read (BLAST, Phymm/PhymmBL, MARTA, and SOrt-ITEMS), the read set was split into five subsets to perform some parallelization and approaches were run separately on each read set. CPU times for the five runs were added up and the average memory of any of the runs is shown in the table. Please note that this a lower bound on the maximum memory required as all reads combined may require more memory.

Species	Type	Coverage	CM #reads	CM confid.	$\sqrt{D_{JS}}$	BLAST #reads	BLAST conf.
L. brevis ATCC 367 plasmid 1	S	1.1e-01	1686	1.000	0.042	1605	1.00000
Acidothermus cellulolyticus 11B	S	2.0e-02	52361	1.000	0.018	50458	0.99988
Pediococcus pentosaceus ATCC 25745	S	3.4e-04	681	0.999	0.064	843	0.85053
Shewanella amazonensis SB2B	S	1.5e-02	68521	0.997	0.035	65504	0.99609
L. brevis ATCC 367	S	2.3e-02	57636	0.992	0.019	56240	0.98131
Myxococcus xanthus DK 1622	S	1.1e-02	111535	0.985	0.000	108008	0.98982
L. lactis subsp. cremoris SK11 plasmid 4	S	6.1e-04	30	0.956	0.404	101	0.24752
L. brevis ATCC 367 plasmid 2	S	1.4e-01	5648	0.768	0.032	5449	0.76766
L. lactis subsp. cremoris SK11 plasmid 3	S	3.1e-04	27	0.587	0.487	117	0.11111
L. casei ATCC 334 plasmid 1	S	5.4e-02	1763	0.567	0.059	1917	0.54199
L. lactis subsp. lactis Il1403	S	5.1e-03	12935	0.435	0.074	13305	0.45133
L. casei ATCC 334	S	1.4e-02	45256	0.393	0.034	46082	0.42958
Halobacterium sp. NRC-1	S	2.2e-05	50	0.370	0.117	3456	0.00376
L. lactis subsp. cremoris SK11	S	1.2e-05	31	0.344	0.647	658	0.00912
L. lactis subsp. cremoris MG1363	R	6.8e-05	191	0.262	0.441	871	0.03100
Lactococcus prophage bIL286	P	8.6e-04	37	0.195	0.205	74	0.00000
Lactococcus prophage bIL311	P	8.8e-03	144	0.182	0.124	166	0.00000
L. lactis subsp. lactis KF147	R	3.3e-04	927	0.171	0.293	6537	0.02937
L. casei BL23	R	4.3e-04	1467	0.153	0.305	21019	0.00081
Lactobacillus buchneri NRRL B-30929 plasmid pLBUC02	R?	2.3e-03	51	0.148	0.482	640	0.00000
Lactococcus prophage bIL309	P	2.0e-03	81	0.138	0.114	205	0.00976
Halobacterium salinarum R1 plasmid PHS2	R	3.5e-04	71	0.121	0.079	540	0.00000
Lactococcus prophage bIL285	P	8.2e-04	29	0.116	0.174	176	0.00000
L. casei str. Zhang	R	2.8e-04	902	0.110	0.328	18885	0.00111
Halobacterium sp. NRC-1 plasmid pNRC100	S	7.1e-04	141	0.109	0.073	959	0.00104
Lactobacillus rhamnosus Lc 705 plasmid pLC1	R?	3.2e-03	272	0.105	0.381	884	0.00000
Halobacterium sp. NRC-1 plasmid pNRC200	S	4.9e-04	187	0.093	0.044	1269	0.00079
Halobacterium salinarum R1 plasmid PHS3	R	1.4e-03	426	0.090	0.041	509	0.00000
Halobacterium salinarum R1 plasmid PHS1	R	3.6e-03	577	0.084	0.040	823	0.00000
Halobacterium salinarum R1	R	1.6e-03	3409	0.070	0.023	3414	0.00029
Lactococcus prophage bIL310	P	1.8e-03	28	0.041	0.199	101	0.00000
Lactobacillus rhamnosus GG	R?	5.0e-05	205	0.034	0.258	1220	0.00000
Lactobacillus fermentum IFO 3956	R?	2.7e-05	81	0.022	0.184	692	0.00000

L. brevis = Lactobacillus brevis, L. lactis = Lactococcus lactis, L. casei = Lactobacillus casei

Table B.3: List of microbe and virus hits identified by ContextMap (CM) on the *in-vitro* simulated microbe community data with a coverage $> 10^{-5}$ and at least 20 reads. Entries are sorted according to ContextMap confidence. The type of the hit is indicated in the following way: S = the species is contained in the sample; R = a close relation is contained in the sample; R? = a more distant relation is contained in the sample; P = a prophage of a species in the sample. For both ContextMap and BLAST, the number of reads mapped to the species and the confidence are provided. In case of BLAST, the number of mapped reads includes also reads that can be mapped equally well to any other species, thus including multiple mappings. BLAST confidence is defined as the fraction of reads that can be mapped uniquely to this species using BLAST.

Species	Type	# reads	P-value
Lactobacillus casei BL23		41484	0.89
Lactobacillus casei str. Zhang		38760	0.97
Lactococcus lactis subsp. lactis Il1403	S	14438	0.00
Shewanella amazonensis SB2B	S	68526	0.00
Lactobacillus brevis ATCC 367	S	58571	0.00
Lactobacillus brevis ATCC 367 plasmid 2	S	5700	0.00
Acidothermus cellulolyticus 11B	S	52376	0.00
Myxococcus xanthus DK 1622	S	111547	0.00
Lactobacillus casei ATCC 334	S	48607	0.00

Table B.4: List of taxa identified by GASiC with p-value < 1. Species contained in the sample are indicated by an S in the second column. Please note that GASiC performs mapping independently for each species. Thus, reads can be mapped to more than one species. An additional 113 species identified by GASiC with a p-value of 1 are not shown.

Species	Type	Abundance
Lactobacillus brevis ATCC 367	S	0.27110
Acidothermus cellulolyticus 11B	S	0.20900
Lactobacillus casei ATCC 334	S	0.16500
Shewanella amazonensis SB2B	S	0.15400
Myxococcus xanthus DK 1622	S	0.11970
Lactococcus lactis subsp. lactis Il1403	S	0.05579
Halobacterium sp. NRC-1	S	0.01847
Pediococcus pentosaceus ATCC 25745	S	0.00400
Lactococcus lactis subsp. lactis KF147		0.00169
Lactococcus lactis subsp. cremoris SK11	S	0.00036
Lactobacillus casei str. Zhang		0.00028
Lactobacillus casei BL23		0.00023
Halobacterium salinarum R1		0.00022
Lactococcus lactis subsp. cremoris MG1363		0.00016

Table B.5: List of taxa identified by GRAMMy with a relative abundance of at least 0.1% (14 out of 63 species identified in total). Species contained in the sample are indicated by an S in the second column. GRAMMy only estimates relative abundances of species in the sample from alignments (in this case BLAST alignments), but performs no resolution of non-unique mappings.

species	abundance	avg. E-value	avg. % ident	avg. alignment length	# hits
Myxococcus xanthus	19161	-7.07	98.75	27.24	9355
Shewanella amazonensis	6438	-7.16	99.12	27.27	3195
Lactobacillus brevis	5416	-7.34	97.35	27.73	2481
Lactobacillus casei	4837	-6.44	98.93	26.08	2899
Acidothermus cellulolyticus	4311	-6.24	99.15	25.76	2367
Lactococcus lactis	2014	-6.26	97.75	26.23	1828
Lactobacillus paracasei	1567	-6.3	98.91	25.94	1129
Halobacterium salinarum	590	-7.26	98.99	27.43	590
unassigned	566	-6.66	92.85	27.72	566
Stigmatella aurantiaca	223	-5.54	87.66	26.58	223
Shewanella baltica	181	-6.25	97.78	25.74	181
Saccharomyces cerevisiae	175	-7.49	98.12	27.51	175
Pediococcus pentosaceus	144	-6.18	98.16	25.4	144
Shewanella sp.	142	-5.37	96.2	25.02	142
Brevibacillus brevis	132	-5.45	85.71	27.55	132
Shewanella putrefaciens	112	-5.85	95.03	25.81	112
Shewanella oneidensis	100	-5.77	93.95	26.01	100
Lactobacillus plantarum	94	-5.39	93.05	25.71	94
Bacillus thuringiensis	72	-6.04	97.58	25.19	72
Halobacterium sp.	70	-6.98	99.02	26.87	68
Shewanella violacea	46	-5.71	87.91	27.46	46
Shewanella loihica	44	-5.81	96.85	25.45	44
Shewanella sp. W3-18-1	42	-5.37	96.69	24.87	42
Shewanella denitrificans	41	-5.84	94.37	25.65	41
Shewanella sp. MR-7	41	-5.16	98.14	24.37	41
Shewanella sp. MR-4	39	-5.44	96.77	24.93	39
Shewanella pealeana	36	-5.23	97.08	24.24	36
Shewanella sp. ANA-3	36	-5.75	98	25.4	36
Shewanella frigidimarina	33	-5.12	91.54	25.57	33
Enterococcus faecalis	31	-6.19	93.8	27.54	31
Shewanella sediminis	27	-5.67	96.81	24.96	27
Bacillus cereus	25	-6.03	95.82	26.17	25
Vibrio cholerae	21	-6.68	97.06	27.24	21

Table B.6: This table shows the evaluation results for MG-RAST on the *in-vitro* simulated microbial community. MG-RAST estimates abundance of individual species based on a protein similarity search between predicted proteins and a reference database. Here, we used the following cutoffs: maximum E-value = 1e-5, minimum identity = 60%, minimum alignment length = 15 amino acids and minimum abundance=20. As MG-RAST only identifies species but not individual strains, it cannot distinguish the *cremoris SK11* subspecies. Furthermore, several species not contained in the sample were found to be more abundant than *Pediococcus pentosaceus* and *Halobacterium sp.*, which are contained in the sample. The latter probably represents *Halobacterium sp. NRC-1*, whereas *Halobacterium salinarum*, which was also found, probably represents the *R1* strain.

genus	% abundance	depth of coverage	number of reads	similarity with reference
Lactobacillus	45.77	3.74	666	99.33
Acidothermus	17.81	1.45	296	99.58
Shewanella	17.1	1.39	289	99.4
Lactococcus	9.25	0.75	143	99.58
Myxococcus	7.43	0.6	144	99.66
Halobacterium	1.16	0.09	18	99.77
Myxococcales{order}	0.37	0.03	7	92.14
Bacillus	0.33	0.02	4	98.75
Pediococcus	0.24	0.01	3	100
Actinomycetales{order}	0.2	0.01	4	91.25
Firmicutes{phylum}	0.13	0.01	2	90
Gammaproteobacteria{class}	0.06	0	1	90
Sphingomonas	0.05	0	1	96
Halobacteriaceae{family}	0.04	0	1	93

Table B.7: This table shows the results for MetaPhyler on the *in-vitro* simulated microbial community. MetaPhyler performs taxonomic classification based on phylogenetic marker genes. As a consequence, the number of reads assigned is relatively small, as few originate from the marker genes. Furthermore, MetaPhyler only performs classification of the genus and not species or strains. Thus, performance in distinguishing the *Lactococcus* and *Halobacterium* species/strains cannot be evaluated. The genera contained in the sample are correctly identified with the exception of *Bacillus* (which is found to be more frequent than *Pediococcus*) and *Sphingomonas*.

species	read count
Lactobacillus	115169
Myxococcus	89568
Shewanella	66810
Acidothermus	43734
Lactococcus	14820
Myxococcales	10574
Lactobacillaceae	4225
Halobacterium	3699
cellular organisms	3654
Lactobacillales	3312
Gammaproteobacteria	2906
Myxococcaceae	2692
Bacteria	2638
Cystobacterineae	2343
Actinomycetales	2090
Bacillus	1544
Proteobacteria	1219
Shewanellaceae	1196
Firmicutes	1096
Saccharomyces	1076
Streptococcaceae	1022
Paenibacillaceae	878
Pediococcus	841
root	791
Alteromonadales	754
Halobacteriaceae	655
Bacilli	498
Bacillaceae	329
Acidothermaceae	320
Streptococcus	315
Bacillales	272
Chondromyces	262
Enterobacteriaceae	258
Brevibacillus	252
Actinobacteria	171
Enterococcus	167
Vibrio	145
Polyangiaceae	140
Frankineae	137
Streptomyces	103

Table B.8: This table shows the results for SOrt-ITEMS on the *in-vitro* simulated microbial community. SOrt-ITEMS assigns reads to a taxon based on significant BLAST hits and performs read assignment at the genus level or higher. All hits with at least 100 assigned reads are shown. The top-ranked hits indeed correspond to genera contained in the sample. However, *Pediococcus* is ranked very low, below other taxa not contained in the sample.

species	read count	avg. E-value	avg. score
Myxococcus xanthus	82489	8.3e-36	196
Shewanella amazonensis	52513	3.1e-35	194
Lactobacillus brevis	49061	1.5e-35	195
Acidothermus cellulolyticus	39097	1.1e-35	195
Lactobacillus casei	36942	3.2e-35	194
Lactococcus lactis	11457	6.0e-35	193
Halobacterium salinarum	3622	1.1e-35	196
Pediococcus pentosaceus	630	1.3e-34	192
Bacillus cereus	335	6.7e-34	181
Lactobacillus rhamnosus	215	1.9e-37	192
Pediococcus claussenii	88	4.6e-43	195
Lactobacillus plantarum	63	9.5e-36	194
Lactobacillus buchneri	55	1.8e-33	190
Lactobacillus fermentum	51	1.2e-38	192
Lactobacillus helveticus	48	8.3e-37	189
Myxococcus fulvus	43	2.1e-34	178
Methylobacterium extorquens	24	8.9e-42	187
Lactobacillus delbrueckii	17	1.8e-43	195
Bacillus anthracis	17	1.2e-34	175
Streptococcus thermophilus	16	3.8e-38	192
Bacillus thuringiensis	14	1.4e-36	176
Methylobacterium radiotolerans	12	1.7e-37	186
Brevibacillus brevis	11	1.8e-38	185

Table B.9: This table shows the results for MARTA, an approach for performing taxonomic classification for BLAST hits, on the *in-vitro* simulated microbial community. All identified species with >10 assigned reads are shown. MARTA performs classification only at the species- not strain-level, thus performance in distinguishing *Halobacterium sp. NRC-1* and *Lactococcus lactis subsp. cremoris SK11* cannot be evaluated. However, all 8 species contained in the sample are ranked higher than all other species in terms of read counts.

placement weight [%]	species
22.1802	Acidothermus cellulolyticus (351607)
12.2243	Myxococcus xanthus (246197)
9.0737	Shewanella oneidensis (211586)
9.0737	Lactococcus lactis 1403 (272623)
9.0107	Lactobacillus plantarum (220668)
6.4902	Pediococcus pentosaceus (278197)
2.8355	LCA of Lactobacillus plantarum (220668) and Pediococcus pentosaceus (278197)
1.8904	Enterococcus faecalis (226185)
1.7013	Halobacterium sp. (64091)
1.5753	LCA of Leuconostoc mesenteroides (203120) and Oenococcus oeni (203123)
1.5123	LCA of Leuconostoc mesenteroides (203120), Oenococcus oeni (203123), Lactobacillus plantarum (220668) and Pediococcus pentosaceus (278197)
1.1972	Streptococcus pneumoniae TIGR4 (170187)
1.0082	LCA of Streptococcus pneumoniae TIGR4 (170187), Leuconostoc mesenteroides (203120), Oenococcus oeni (203123), Lactobacillus plantarum (220668), Enterococcus faecalis (226185), Lactococcus lactis 1403 (272623) and Pediococcus pentosaceus (278197)
0.9452	Leuconostoc mesenteroides (203120)
0.8822	Oenococcus oeni (203123)
0.7561	Pseudoalteromonas haloplanktis (326442)
0.6931	LCA of Wigglesworthia glossinidia (36870), Haemophilus influenzae KW20 (71421), Escherichia coli K12 (83333), Buchnera aphidicola APS (107806), Colwellia psychrerythraea (167879), Shigella flexneri 301 (198214), Blochmannia floridanus (203907), Shewanella oneidensis (211586), Yersinia pestis CO92 (214092), Erwinia carotovora (218491), Salmonella enterica CT18 (220341), Mannheimia succiniciproducens (221988), Photorhabdus luminescens (243265), Vibrio cholerae N16961 (243277), Klebsiella pneumoniae (272620), Pasteurella multocida (272843), Citrobacter koseri (290338), Enterobacter sakazakii (290339), Photobacterium profundum (298386), Pseudoalteromonas haloplanktis (326442), Sodalis glossinidius (343509), Psychromonas ingrahamii (357804), Aeromonas hydrophila (380703), Serratia proteamaculans (399741) and Actinobacillus pleuropneumoniae (416269)
0.6301	Photobacterium profundum (298386)
0.6301	Listeria innocua (272626)
0.5671	Frankia sp. CcI3 (106370)
0.5041	LCA of Streptococcus pneumoniae TIGR4 (170187), Enterococcus faecalis (226185) and Lactococcus lactis 1403 (272623)
0.5041	LCA of Shewanella oneidensis (211586) and Psychromonas ingrahamii (357804)
0.5041	LCA of Escherichia coli K12 (83333), Shigella flexneri 301 (198214), Salmonella enterica CT18 (220341) and Citrobacter koseri (290338)
0.5041	Psychromonas ingrahamii (357804)
0.5041	Geobacillus kaustophilus (235909)
0.5041	Bacillus subtilis (224308)
0.5041	Aeromonas hydrophila (380703)

Table B.10: This table shows the results for MLTreeMap on the *in-vitro* simulated microbial community. All results with a placement weight of at least 0.05% are shown. Numbers in parenthesis indicate the taxon identifier of the corresponding species. LCA is short for lowest common ancestor. Although only species names are provided by MLTreeMap, classification is performed at the strain-level as indicated by the taxon identifiers. Here, the highest-ranking hits are enriched for the correct species but often only the correct genus is identified. The following species/strains are missed: *Lactobacillus brevis*, *Lactobacillus casei*, *Lactobacillus casei*, *Lactococcus lactis subsp. cremoris SK11*, and *Shewanella amazonensis*. Furthermore, *Halobacterium sp.* is ranked below *Enterococcus faecalis*, which is not contained in the sample.

species	read count
Desulfovibrio vulgaris	3063
Lactobacillus plantarum	2172
Lactobacillus casei	2170
Bifidobacterium animalis	1972
Bifidobacterium longum	1814
Xylella fastidiosa	1266
Lactococcus lactis	1046
Streptococcus equi	1020
Lactobacillus delbrueckii	740
Streptococcus suis	609
Pseudomonas putida	524
Pseudomonas aeruginosa	514
Vibrio cholerae	510
Prochlorococcus	409
Bacillus subtilis	398
Rhodobacter sphaeroides	363
Streptococcus pyogenes	317
Methanococcus maripaludis	316
Shewanella baltica	226
Clostridium difficile	178
Neisseria meningitidis	167
Mycobacterium tuberculosis	153
Streptococcus thermophilus	134
Haemophilus influenzae	131
Mycobacterium bovis	127
Actinobacillus pleuropneumoniae	127
Vibrio vulnificus	124
Xanthomonas oryzae	121
Streptococcus agalactiae	117
Francisella tularensis	112
Helicobacter pylori	110
Streptococcus pneumoniae	110

Table B.11: This table shows the results for PhyloPhytiaS, a composition-based approach for species identification, on the *in-vitro* simulated microbial community. PhyloPhytiaS performs classification only at the species- not the strain-level. Here, results were obtained using the generic model provided by the webserver and all hits with > 100 reads were retained. With the exception of two species, none of these predicted species are contained in the sample.

species	read count	average distance
Herpetosiphon aurantiacus DSM 785	30551	0.018
Dichelobacter nodosus VCS1703A	16059	0.019
Acidaminococcus fermentans DSM 20731	11506	0.018
Conexibacter woesei DSM 14684	11443	0.024
Synechococcus sp. RCC307	10286	0.018
Kribbella flavida DSM 17836	10142	0.022
Stenotrophomonas maltophilia JV3	9192	0.021
Stenotrophomonas maltophilia R551-3	9127	0.020
Moraxella catarrhalis RH4	8709	0.020
Cellulomonas fimi ATCC 484	8615	0.028
Leptothrix cholodnii SP-6	8414	0.021
Mycoplasma gallisepticum str. R(low)	8061	0.020
Beutenbergia cavernae DSM 12333	7185	0.025
Spirochaeta thermophila DSM 6192	6698	0.019
Methylibium petroleiphilum PM1	6696	0.018
Myxococcus fulvus HW-1	6468	0.019
Kineococcus radiotolerans SRS30216 plasmid pKRAD02	6358	0.024
Mycoplasma suis str. Illinois	6296	0.022
Treponema brennaborense DSM 12168	6075	0.019
Helicobacter pylori B8 plasmid HPB8p	5926	0.019
Anaeromyxobacter sp. Fw109-5	5803	0.025
Thermus thermophilus HB8	5506	0.026
Anaeromyxobacter dehalogenans 2CP-C	5395	0.027
Phenylobacterium zucineum HLK1	5089	0.022
Eubacterium eligens ATCC 27750 plasmid unnamed	5058	0.020
Nitrosopumilus maritimus SCM1	4950	0.021
Ramlibacter tataouinensis TTB310	4895	0.020
Micromonospora sp. L5	4785	0.022
Sanguibacter keddieii DSM 10542	4046	0.022
Cellvibrio gilvus ATCC 13127	3985	0.024
Blattabacterium sp. (Mastotermes darwiniensis) str. MADAR plasmid pMADAR_001	3714	0.020
Halogeometricum borinquense DSM 11551 plasmid pHBOR05	3646	0.019
Cellulomonas flavigena DSM 20109	3616	0.026
Mycobacterium ulcerans AGY99 plasmid pMUM001	3393	0.017
Brevundimonas subvibrioides ATCC 15264	3359	0.019
Rhodospirillum centenum SW	3295	0.021
Escherichia coli O26:H11 str. 11368 plasmid pO26_2	3292	0.018
Persephonella marina EX-H1	3275	0.017
Nakamurella multipartita DSM 44233	3255	0.019
Candidatus Riesia pediculicola USDA plasmid pPAN	3214	0.022
Staphylococcus epidermidis ATCC 12228 plasmid pSE-12228-03	3135	0.019
Helicobacter bizzozeronii CIII-1	3065	0.016

Table B.12: This table shows the results for ClaMS, a composition-based approach, on the *in-vitro* simulated microbial community. ClaMS models each sequence as a walk in a de Bruijn graph with underlying Markov chain properties. For each read to be binned, a signature is calculated and compared to a training set of signatures from genome sequence. If the normalized distance to the best signature match exceeds a certain threshold, it is assigned to this genome, otherwise the sequence is not binned. Here, we used a distance cutoff of 0.05 as the recommended cutoff of 0.01 resulted in no assigned reads. In the table all all hits with > 3000 reads are shown. As can be seen, none of these are are contained in the sample and only one belongs to a correct genus.

Phymm species	Phymm read count	PhymmBL species	PhymmBL read count
Shewanella amazonensis SB2B	41000	Myxococcus xanthus DK 1622	132734
Myxococcus xanthus DK 1622	40760	Shewanella amazonensis SB2B	91286
Lactobacillus brevis ATCC 367	31059	Acidothermus cellulolyticus 11B	64763
Acidothermus cellulolyticus 11B	26688	Lactobacillus brevis ATCC 367	54952
Lactobacillus brevis KB290	19695	Lactobacillus casei ATCC 334	41734
Lactobacillus casei ATCC 334	15668	Lactobacillus brevis KB290	27817
Myxococcus fulvus HW-1	13534	Lactococcus lactis subsp. lactis Il1403	9525
Lactobacillus casei str. Zhang	8154	Lactobacillus casei str. Zhang	9396
Corallococcus coralloides DSM 2259	6155	Lactococcus lactis subsp. lactis CV56	7056
Myxococcus stipitatus DSM 14675	5448	Lactobacillus casei LC2W	5083
Lactobacillus casei LC2W	4385	Lactobacillus casei W56	4209
Lactobacillus casei BL23	3926	Lactobacillus casei BD-II	4202
Lactobacillus casei BD-II	3801	Lactobacillus casei BL23	4164
Lactobacillus casei W56	3624	Halobacterium salinarum R1	2909
Lactococcus lactis subsp. lactis Il1403	3596	Lactococcus lactis subsp. lactis KF147	2882
Lactococcus lactis subsp. lactis CV56	3076	Halobacterium sp. NRC-1	2873
Halobacterium salinarum R1	2211	Lactococcus lactis subsp. lactis IO-1	1968
Halobacterium sp. NRC-1	2180	Pediococcus pentosaceus ATCC 25745	1166
Lactococcus lactis subsp. lactis KF147	1779	Myxococcus fulvus HW-1	913
Stigmatella aurantiaca DW4SLASH3-1	1770	Lactococcus lactis subsp. cremoris A76	702
Lactococcus lactis subsp. lactis IO-1	1764	Lactobacillus plantarum subsp. plantarum P-8	621
Azospirillum brasilense Sp245	1119	Lactobacillus plantarum WCFS1	334
Deinococcus gobiensis I-0	1113	Lactobacillus rhamnosus Lc 705	275
Lactobacillus plantarum subsp. plantarum P-8	1026	Bacillus cereus FRI-35	243
Pseudonocardia dioxanivorans CB1190	925	Lactococcus lactis subsp. cremoris SK11	208
Sinorhizobium fredii USDA 257	921	Nonlabens dokdonensis DSW-6	200
Kineococcus radiotolerans SRS30216	911	Pediococcus claussenii ATCC BAA-344	173
Frankia symbiont of Datisca glomerata	894	Lactococcus lactis subsp. cremoris MG1363	171
Lactococcus lactis subsp. cremoris A76	885	Bacillus cereus ATCC 10987	153
Streptomyces cattleya NRRL 8057 = DSM 46488	878	Bacillus cereus AH187	146

Table B.13: This table shows the results for Phymm, a composition-based approach, and PhymmBL, a hybrid approach combining Phymm and BLAST results, on the *in-vitro* simulated microbial community. The top 30 hits for either method are listed. In both cases, the correct strains are enriched towards the top of the tables. However, a number of related species or strains are ranked higher than correct hits, in particular higher than *Halobacterium sp. NRC-1*, *Pediococcus pentosaceus*, and *Lactococcus lactis subsp. cremoris SK11*. Here, the hybrid approach PhymmBL seems to perform better than the (only) composition-based Phymm approach as most of the highly ranked wrong hits are at least in the correct species even if not the correct strain.

Bibliography

Abecasis, G. R., Altshuler, D., Auton, A., Brooks, L. D., Durbin, R. M., Gibbs, R. A., Hurles, M. E., and McVean, G. A. (2010). A map of human genome variation from population-scale sequencing. *Nature*, 467(7319):1061–73.

Alamancos, G. P., Agirre, E., and Eyras, E. (2014). Methods to study splicing from high-throughput RNA sequencing data. *Methods Mol Biol*, 1126:357–397.

Altschul, S. F., Gish, W., Miller, W., Myers, E. W., and Lipman, D. J. (1990). Basic local alignment search tool. *J Mol Biol*, 215(3):403–10.

Bao, G., Wang, M., Doak, T. G., and Ye, Y. (2015). Strand-specific community RNA-seq reveals prevalent and dynamic antisense transcription in human gut microbiota. *Front Microbiol*, 6:896.

Berger, M. F., Levin, J. Z., Vijayendran, K., Sivachenko, A., Adiconis, X., Maguire, J., Johnson, L. A., Robinson, J., Verhaak, R. G., Sougnez, C., Onofrio, R. C., Ziaugra, L., Cibulskis, K., Laine, E., Barretina, J., Winckler, W., Fisher, D. E., Getz, G., Meyerson, M., Jaffe, D. B., Gabriel, S. B., Lander, E. S., Dummer, R., Gnirke, A., Nusbaum, C., and Garraway, L. A. (2010). Integrative analysis of the melanoma transcriptome. *Genome Res*, 20(4):413–27.

Birol, I., Jackman, S. D., Nielsen, C. B., Qian, J. Q., Varhol, R., Stazyk, G., Morin, R. D., Zhao, Y., Hirst, M., Schein, J. E., Horsman, D. E., Connors, J. M., Gascoyne, R. D., Marra, M. A., and Jones, S. J. (2009). De novo transcriptome assembly with ABySS. *Bioinformatics*, 25(21):2872–7.

Birol, I., Raymond, A., Chiu, R., Nip, K. M., Jackman, S. D., Kreitzman, M., Docking, T. R., Ennis, C. A., Robertson, A. G., and Karsan, A. (2015). Kleat: cleavage site analysis of transcriptomes. *Pacific Symposium on Biocomputing. Pacific Symposium on Biocomputing*, pages 347–58.

Bonfert, T., Csaba, G., Zimmer, R., and Friedel, C. C. (2012). A context-based approach to identify the most likely mapping for RNA-seq experiments. *BMC Bioinformatics*, 13 Suppl 6:S9.

Bonfert, T., Csaba, G., Zimmer, R., and Friedel, C. C. (2013). Mining RNA-seq data for infections and contaminations. *PLoS One*, 8(9):e73071.

Bonfert, T., Kirner, E., Csaba, G., Zimmer, R., and Friedel, C. C. (2015). ContextMap 2: fast and accurate context-based RNA-seq mapping. *BMC Bioinformatics*, 16(1):122.

Brady, A. and Salzberg, S. L. (2009). Phymm and PhymmBL: metagenomic phylogenetic classification with interpolated Markov models. *Nat Methods*, 6(9):673–676.

Brenner, S., Jacob, F., and Meselson, M. (1961). An unstable intermediate carrying information from genes to ribosomes for protein synthesis. *Nature*, 190:576–581.

Buch, S., Schafmayer, C., Volzke, H., Becker, C., Franke, A., von Eller-Eberstein, H., Kluck, C., Bassmann, I., Brosch, M., Lammert, F., Miquel, J. F., Nervi, F., Wittig, M., Rosskopf, D., Timm, B., Holl, C., Seeger, M., ElSharawy, A., Lu, T., Egberts, J., Fandrich, F., Folsch, U. R., Krawczak, M., Schreiber, S., Nurnberg, P., Tepel, J., and Hampe, J. (2007). A genome-wide association scan identifies the hepatic cholesterol transporter ABCG8 as a susceptibility factor for human gallstone disease. *Nat Genet*, 39(8):995–9.

Burrows, M. and Wheeler, D. J. (1994). A block-sorting lossless data compression algorithm. Technical Report 124, Digital Systems Research Center.

Castellarin, M., Warren, R. L., Freeman, J. D., Dreolini, L., Krzywinski, M., Strauss, J., Barnes, R., Watson, P., Allen-Vercoe, E., Moore, R. A., and Holt, R. A. (2012). Fusobacterium nucleatum infection is prevalent in human colorectal carcinoma. *Genome Res*, 22(2):299–306.

Chen, L. Y., Wei, K.-C., Huang, A. C.-Y., Wang, K., Huang, C.-Y., Yi, D., Tang, C. Y., Galas, D. J., and Hood, L. E. (2012). RNASEQR–a streamlined and accurate RNA-seq sequence analysis program. *Nucleic Acids Res*, 40(6):e42.

Clement, J. Q., Qian, L., Kaplinsky, N., and Wilkinson, M. F. (1999). The stability and fate of a spliced intron from vertebrate cells. *RNA*, 5(2):206–20.

Crick, F. H. (1958). On protein synthesis. *Symp Soc Exp Biol*, 12:138–63.

Dahm, R. (2008). Discovering DNA: Friedrich Miescher and the early years of nucleic acid research. *Hum Genet*, 122(6):565–81.

Djebali, S., Davis, C. A., Merkel, A., Dobin, A., Lassmann, T., Mortazavi, A., Tanzer, A., Lagarde, J., Lin, W., Schlesinger, F., et al. (2012). Landscape of transcription in human cells. *Nature*, 489(7414):101–108.

Dobin, A., Davis, C. A., Schlesinger, F., Drenkow, J., Zaleski, C., Jha, S., Batut, P., Chaisson, M., and Gingeras, T. R. (2013). STAR: ultrafast universal RNA-seq aligner. *Bioinformatics*, 29(1):15–21.

Dohm, J. C., Lottaz, C., Borodina, T., and Himmelbauer, H. (2008). Substantial biases in ultra-short read data sets from high-throughput DNA sequencing. *Nucleic Acids Res*, 36(16):e105.

Dolken, L., Ruzsics, Z., Radle, B., Friedel, C. C., Zimmer, R., Mages, J., Hoffmann, R., Dickinson, P., Forster, T., Ghazal, P., and Koszinowski, U. H. (2008). High-resolution gene expression profiling for simultaneous kinetic parameter analysis of RNA synthesis and decay. *RNA*, 14(9):1959–72.

ENCODE Project Consortium (2012). An integrated encyclopedia of DNA elements in the human genome. *Nature*, 489(7414):57–74.

Endres, D. and Schindelin, J. (2003). A new metric for probability distributions. *Information Theory, IEEE Transactions on*, 49(7):1858–1860.

Engstrom, P. G., Steijger, T., Sipos, B., Grant, G. R., Kahles, A., Ratsch, G., Goldman, N., Hubbard, T. J., Harrow, J., Guigo, R., and Bertone, P. (2013). Systematic evaluation of spliced alignment programs for RNA-seq data. *Nat Methods*, 10(12):1185–91.

Fatica, A. and Bozzoni, I. (2014). Long non-coding RNAs: new players in cell differentiation and development. *Nat Rev Genet*, 15(1):7–21.

Ferragina, P. and Manzini, G. (2000). Opportunistic data structures with applications. In *Proceedings of the 41st Annual Symposium on Foundations of Computer Science*, page 390. IEEE Computer Society. ACM ID: 796543.

Finotello, F. and Di Camillo, B. (2015). Measuring differential gene expression with RNA-seq: challenges and strategies for data analysis. *Brief Funct Genomics*, 14(2):130–42.

Flicek, P. and Birney, E. (2009). Sense from sequence reads: methods for alignment and assembly. *Nat Methods*, 6(11 Suppl):S6–S12.

Fonseca, N. A., Rung, J., Brazma, A., and Marioni, J. C. (2012). Tools for mapping high-throughput sequencing data. *Bioinformatics*, 28(24):3169–77.

Friedel, C. C. and Dolken, L. (2009). Metabolic tagging and purification of nascent RNA: implications for transcriptomics. *Mol Biosyst*, 5(11):1271–8.

Friedel, C. C., Dolken, L., Ruzsics, Z., Koszinowski, U. H., and Zimmer, R. (2009). Conserved principles of mammalian transcriptional regulation revealed by RNA half-life. *Nucleic Acids Res*, 37(17):e115.

Fuller, C. W., Middendorf, L. R., Benner, S. A., Church, G. M., Harris, T., Huang, X., Jovanovich, S. B., Nelson, J. R., Schloss, J. A., Schwartz, D. C., and Vezenov, D. V. (2009). The challenges of sequencing by synthesis. *Nat Biotechnol*, 27(11):1013–23.

Garber, M., Grabherr, M. G., Guttman, M., and Trapnell, C. (2011). Computational methods for transcriptome annotation and quantification using RNA-seq. *Nat Methods*, 8(6):469–477.

Grant, G. R., Farkas, M. H., Pizarro, A. D., Lahens, N. F., Schug, J., Brunk, B. P., Stoeckert, C. J., Hogenesch, J. B., and Pierce, E. A. (2011). Comparative analysis of RNA-Seq alignment algorithms and the RNA-Seq unified mapper (RUM). *Bioinformatics*, 27(18):2518–2528.

Guo, H., Ingolia, N. T., Weissman, J. S., and Bartel, D. P. (2010). Mammalian microRNAs predominantly act to decrease target mRNA levels. *Nature*, 466(7308):835–840.

Hafner, M., Landthaler, M., Burger, L., Khorshid, M., Hausser, J., Berninger, P., Rothballer, A., Ascano, Jr, M., Jungkamp, A. C., Munschauer, M., Ulrich, A., Wardle, G. S., Dewell, S., Zavolan, M., and Tuschl, T. (2010). Transcriptome-wide identification of RNA-binding protein and microRNA target sites by PAR-CLIP. *Cell*, 141(1):129–41.

Harrison, P. M., Zheng, D., Zhang, Z., Carriero, N., and Gerstein, M. (2005). Transcribed processed pseudogenes in the human genome: an intermediate form of expressed retrosequence lacking protein-coding ability. *Nucleic Acids Res*, 33(8):2374–83.

Hayden, E. C. (2014). Technology: The $1,000 genome. *Nature*, 507(7492):294–5.

Hoagland, M. B., Stephenson, M., Scott, J. F., Hechtl, L. I., and Zamecnik, P. C. (1958). A soluble ribonucleic acid intermediate in protein synthesis. *J Biol Chem*, 231(1):241–57.

Homer, N., Merriman, B., and Nelson, S. F. (2009). BFAST: an alignment tool for large scale genome resequencing. *PLoS One*, 4(11):e7767.

Horton, M., Bodenhausen, N., and Bergelson, J. (2010). MARTA: a suite of Java-based tools for assigning taxonomic status to DNA sequences. *Bioinformatics*, 26(4):568–569.

Hunten, S., Kaller, M., Drepper, F., Oeljeklaus, S., Bonfert, T., Erhard, F., Dueck, A., Eichner, N., Friedel, C. C., Meister, G., Zimmer, R., Warscheid, B., and Hermeking, H. (2015). p53-regulated networks of protein, mRNA, miRNA and lncRNA expression revealed by integrated pSILAC and NGS analyses. *Mol Cell Proteomics*, 14(10):2609–2629.

Hurd, P. J. and Nelson, C. J. (2009). Advantages of next-generation sequencing versus the microarray in epigenetic research. *Brief Funct Genomic Proteomic*, 8(3):174–83.

Huson, D. H., Mitra, S., Ruscheweyh, H.-J., Weber, N., and Schuster, S. C. (2011). Integrative analysis of environmental sequences using MEGAN4. *Genome Res*, 21(9):1552–1560.

Inagaki, Y., Tsunokawa, Y., Takebe, N., Nawa, H., Nakanishi, S., Terada, M., and Sugimura, T. (1988). Nucleotide sequences of cDNAs for human papillomavirus type 18 transcripts in HeLa cells. *J Virol*, 62(5):1640–1646.

Ingolia, N. T. (2014). Ribosome profiling: new views of translation, from single codons to genome scale. *Nat Rev Genet*, 15(3):205–13.

Ingolia, N. T., Ghaemmaghami, S., Newman, J. R., and Weissman, J. S. (2009). Genome-wide analysis in vivo of translation with nucleotide resolution using ribosome profiling. *Science*, 324(5924):218–23.

International Human Genome Sequencing Consortium (2004). Finishing the euchromatic sequence of the human genome. *Nature*, 431(7011):931–45.

Jha, A., Panzade, G., Pandey, R., and Shankar, R. (2015). A legion of potential regulatory sRNAs exists beyond the typical microRNAs microcosm. *Nucleic Acids Res.*

Johnson, D. S., Mortazavi, A., Myers, R. M., and Wold, B. (2007). Genome-wide mapping of in vivo protein-DNA interactions. *Science*, 316(5830):1497–502.

Jonas, S. and Izaurralde, E. (2015). Towards a molecular understanding of microRNA-mediated gene silencing. *Nat Rev Genet*, 16(7):421–33.

Kent, W. J. (2002). BLAT–the BLAST-like alignment tool. *Genome Res*, 12(4):656–664.

Kim, D., Pertea, G., Trapnell, C., Pimentel, H., Kelley, R., and Salzberg, S. L. (2013). TopHat2: accurate alignment of transcriptomes in the presence of insertions, deletions and gene fusions. *Genome Biol*, 14(4):R36.

Kim, H. D., Shay, T., O'Shea, E. K., and Regev, A. (2009). Transcriptional regulatory circuits: predicting numbers from alphabets. *Science*, 325(5939):429–32.

Konstantinidis, K. T., Ramette, A., and Tiedje, J. M. (2006). The bacterial species definition in the genomic era. *Philos Trans R Soc Lond B Biol Sci*, 361(1475):1929–1940.

Kostic, A. D., Ojesina, A. I., Pedamallu, C. S., Jung, J., Verhaak, R. G. W., Getz, G., and Meyerson, M. (2011). PathSeq: software to identify or discover microbes by deep sequencing of human tissue. *Nat Biotechnol*, 29(5):393–396.

Ku, C. S., Loy, E. Y., Salim, A., Pawitan, Y., and Chia, K. S. (2010). The discovery of human genetic variations and their use as disease markers: past, present and future. *J Hum Genet*, 55(7):403–15.

Lamond, A. I., Konarska, M. M., Grabowski, P. J., and Sharp, P. A. (1988). Spliceosome assembly involves the binding and release of U4 small nuclear ribonucleoprotein. *Proc Natl Acad Sci U S A*, 85(2):411–5.

Lander, E., Linton, L., Birren, B., Nusbaum, C., Zody, M., Baldwin, J., Devon, K., Dewar, K., Doyle, M., FitzHugh, W., et al. (2001). Initial sequencing and analysis of the human genome. *Nature*, 409:860–921.

Langmead, B. and Salzberg, S. L. (2012). Fast gapped-read alignment with Bowtie 2. *Nat Methods*, 9(4):357–9.

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

Lasa, I., Toledo-Arana, A., Dobin, A., Villanueva, M., de los Mozos, I. R., Vergara-Irigaray, M., Segura, V., Fagegaltier, D., Penades, J. R., Valle, J., Solano, C., and Gingeras, T. R. (2011). Genome-wide antisense transcription drives mRNA processing in bacteria. *Proc Natl Acad Sci U S A*, 108(50):20172–7.

Lee, W. P., Stromberg, M. P., Ward, A., Stewart, C., Garrison, E. P., and Marth, G. T. (2014). MOSAIK: a hash-based algorithm for accurate next-generation sequencing short-read mapping. *PLoS One*, 9(3):e90581.

Leinonen, R., Akhtar, R., Birney, E., Bower, L., Cerdeno-TÃ¡rraga, A., Cheng, Y., Cleland, I., Faruque, N., Goodgame, N., Gibson, R., Hoad, G., Jang, M., Pakseresht, N., Plaister, S., Radhakrishnan, R., Reddy, K., Sobhany, S., Ten Hoopen, P., Vaughan, R., Zalunin, V., and Cochrane, G. (2011). The European Nucleotide Archive. *Nucleic Acids Res*, 39(Database issue):D28–31.

Li, B., Ruotti, V., Stewart, R. M., Thomson, J. A., and Dewey, C. N. (2010). RNA-Seq gene expression estimation with read mapping uncertainty. *Bioinformatics*, 26(4):493–500.

Li, H. and Durbin, R. (2009). Fast and accurate short read alignment with Burrows-Wheeler transform. *Bioinformatics*, 25(14):1754–60.

Li, H. and Durbin, R. (2010). Fast and accurate long-read alignment with Burrows-Wheeler transform. *Bioinformatics*, 26(5):589–595.

Li, H., Handsaker, B., Wysoker, A., Fennell, T., Ruan, J., Homer, N., Marth, G., Abecasis, G., Durbin, R., and 1000 Genome Project Data Processing Subgroup (2009). The Sequence Alignment/Map format and SAMtools. *Bioinformatics*, 25(16):2078–2079.

Li, H. and Homer, N. (2010). A survey of sequence alignment algorithms for next-generation sequencing. *Brief Bioinform*, 11(5):473–83.

Li, H., Ruan, J., and Durbin, R. (2008a). Mapping short DNA sequencing reads and calling variants using mapping quality scores. *Genome Res*, 18(11):1851–1858.

Li, R., Li, Y., Kristiansen, K., and Wang, J. (2008b). SOAP: short oligonucleotide alignment program. *Bioinformatics*, 24(5):713–4.

Lim, Y. W., Schmieder, R., Haynes, M., Willner, D., Furlan, M., Youle, M., Abbott, K., Edwards, R., Evangelista, J., Conrad, D., and Rohwer, F. (2012). Metagenomics and metatranscriptomics: Windows on CF-associated viral and microbial communities. *J Cyst Fibros*.

Lin, H., Zhang, Z., Zhang, M. Q., Ma, B., and Li, M. (2008). ZOOM! Zillions of oligos mapped. *Bioinformatics*, 24(21):2431–7.

Lin, J. (1991). Divergence measures based on the Shannon entropy. *IEEE Transactions on Information theory*, 37:145–151.

Lindner, M. S. and Renard, B. Y. (2013). Metagenomic abundance estimation and diagnostic testing on species level. *Nucleic Acids Res*, 41(1):e10.

Lindner, R. and Friedel, C. C. (2012). A comprehensive evaluation of alignment algorithms in the context of RNA-seq. *PLoS One*, 7(12):e52403.

Lipman, D. J. and Pearson, W. R. (1985). Rapid and sensitive protein similarity searches. *Science*, 227(4693):1435–41.

Liu, B., Gibbons, T., Ghodsi, M., Treangen, T., and Pop, M. (2011). Accurate and fast estimation of taxonomic profiles from metagenomic shotgun sequences. *BMC Genomics*, 12 Suppl 2:S4.

Liu, L., Li, Y., Li, S., Hu, N., He, Y., Pong, R., Lin, D., Lu, L., and Law, M. (2012). Comparison of next-generation sequencing systems. *J Biomed Biotechnol*, 2012:251364.

Luo, H., Sun, S., Li, P., Bu, D., Cao, H., and Zhao, Y. (2013). Comprehensive characterization of 10,571 mouse large intergenic noncoding RNAs from whole transcriptome sequencing. *PLoS One*, 8(8):e70835.

Magaldi, T. G., Almstead, L. L., Bellone, S., Prevatt, E. G., Santin, A. D., and DiMaio, D. (2012). Primary human cervical carcinoma cells require human papillomavirus E6 and E7 expression for ongoing proliferation. *Virology*, 422(1):114–124.

Maher, C. A., Kumar-Sinha, C., Cao, X., Kalyana-Sundaram, S., Han, B., Jing, X., Sam, L., Barrette, T., Palanisamy, N., and Chinnaiyan, A. M. (2009). Transcriptome sequencing to detect gene fusions in cancer. *Nature*, 458(7234):97–101.

Malone, J. H. and Oliver, B. (2011). Microarrays, deep sequencing and the true measure of the transcriptome. *BMC Biol*, 9:34.

Marcinowski, L., Lidschreiber, M., Windhager, L., Rieder, M., Bosse, J. B., Rdle, B., Bonfert, T., Gyry, I., de Graaf, M., Prazeres da Costa, O., Rosenstiel, P., Friedel, C. C., Zimmer, R., Ruzsics, Z., and Dlken, L. (2012). Real-time transcriptional profiling of cellular and viral gene expression during lytic cytomegalovirus infection. *PLoS Pathog*, 8(9):e1002908.

Margulies, M., Egholm, M., Altman, W. E., Attiya, S., Bader, J. S., Bemben, L. A., Berka, J., Braverman, M. S., Chen, Y. J., Chen, Z., Dewell, S. B., Du, L., Fierro, J. M., Gomes, X. V., Godwin, B. C., He, W., Helgesen, S., Ho, C. H., Irzyk, G. P., Jando, S. C., Alenquer, M. L., Jarvie, T. P., Jirage, K. B., Kim, J. B., Knight, J. R., Lanza, J. R.,

Leamon, J. H., Lefkowitz, S. M., Lei, M., Li, J., Lohman, K. L., Lu, H., Makhijani, V. B., McDade, K. E., McKenna, M. P., Myers, E. W., Nickerson, E., Nobile, J. R., Plant, R., Puc, B. P., Ronan, M. T., Roth, G. T., Sarkis, G. J., Simons, J. F., Simpson, J. W., Srinivasan, M., Tartaro, K. R., Tomasz, A., Vogt, K. A., Volkmer, G. A., Wang, S. H., Wang, Y., Weiner, M. P., Yu, P., Begley, R. F., and Rothberg, J. M. (2005). Genome sequencing in microfabricated high-density picolitre reactors. *Nature*, 437(7057):376–80.

Martin, J. A. and Wang, Z. (2011). Next-generation transcriptome assembly. *Nat Rev Genet*, 12(10):671–82.

Mattick, J. S. and Makunin, I. V. (2006). Non-coding RNA. *Hum Mol Genet*, 15 Spec No 1:R17–29.

Maxam, A. M. and Gilbert, W. (1977). A new method for sequencing DNA. *Proc Natl Acad Sci U S A*, 74(2):560–4.

McHardy, A. C., Martin, H. G., Tsirigos, A., Hugenholtz, P., and Rigoutsos, I. (2007). Accurate phylogenetic classification of variable-length DNA fragments. *Nat Methods*, 4(1):63–72.

Melvin, W. T., Milne, H. B., Slater, A. A., Allen, H. J., and Keir, H. M. (1978). Incorporation of 6-thioguanosine and 4-thiouridine into RNA. Application to isolation of newly synthesised RNA by affinity chromatography. *Eur J Biochem*, 92(2):373–9.

Metzker, M. L. (2010). Sequencing technologies - the next generation. *Nat Rev Genet*, 11(1):31–46.

Meyer, F., Paarmann, D., D'Souza, M., Olson, R., Glass, E. M., Kubal, M., Paczian, T., Rodriguez, A., Stevens, R., Wilke, A., Wilkening, J., and Edwards, R. A. (2008). The metagenomics RAST server - a public resource for the automatic phylogenetic and functional analysis of metagenomes. *BMC Bioinformatics*, 9:386.

Mighell, A. J., Smith, N. R., Robinson, P. A., and Markham, A. F. (2000). Vertebrate pseudogenes. *FEBS Lett*, 468(2-3):109–14.

Miller, C., Schwalb, B., Maier, K., Schulz, D., Dumcke, S., Zacher, B., Mayer, A., Sydow, J., Marcinowski, L., Dolken, L., Martin, D. E., Tresch, A., and Cramer, P. (2011). Dynamic transcriptome analysis measures rates of mRNA synthesis and decay in yeast. *Mol Syst Biol*, 7:458.

Monzoorul Haque, M., Ghosh, T. S., Komanduri, D., and Mande, S. S. (2009). SOrt-ITEMS: Sequence orthology based approach for improved taxonomic estimation of metagenomic sequences. *Bioinformatics*, 25(14):1722–1730.

Moore, R. A., Warren, R. L., Freeman, J. D., Gustavsen, J. A., Chnard, C., Friedman, J. M., Suttle, C. A., Zhao, Y., and Holt, R. A. (2011). The sensitivity of massively parallel

sequencing for detecting candidate infectious agents associated with human tissue. *PLoS One*, 6(5):e19838.

Morgan, J. L., Darling, A. E., and Eisen, J. A. (2010). Metagenomic sequencing of an in vitro-simulated microbial community. *PLoS One*, 5(4):e10209.

Morris, K. V. and Mattick, J. S. (2014). The rise of regulatory RNA. *Nat Rev Genet*, 15(6):423–37.

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

Mullaney, J. M., Mills, R. E., Pittard, W. S., and Devine, S. E. (2010). Small insertions and deletions (INDELs) in human genomes. *Hum Mol Genet*, 19(R2):R131–6.

Nam, K., Lee, G., Trambley, J., Devine, S. E., and Boeke, J. D. (1997). Severe growth defect in a Schizosaccharomyces pombe mutant defective in intron lariat degradation. *Mol Cell Biol*, 17(2):809–18.

Needleman, S. B. and Wunsch, C. D. (1970). A general method applicable to the search for similarities in the amino acid sequence of two proteins. *J Mol Biol*, 48(3):443–53.

Ng, P., Wei, C.-L., Sung, W.-K., Chiu, K. P., Lipovich, L., Ang, C. C., Gupta, S., Shahab, A., Ridwan, A., Wong, C. H., Liu, E. T., and Ruan, Y. (2005). Gene identification signature (gis) analysis for transcriptome characterization and genome annotation. *Nature methods*, 2:105–11.

NIH HMP Working Group, Peterson, J., Garges, S., Giovanni, M., McInnes, P., Wang, L., Schloss, J. A., Bonazzi, V., McEwen, J. E., Wetterstrand, K. A., et al. (2009). The NIH Human Microbiome Project. *Genome Res*, 19(12):2317–2323.

O'Neil, D., Glowatz, H., and Schlumpberger, M. (2013). Ribosomal RNA depletion for efficient use of RNA-seq capacity. *Curr Protoc Mol Biol*, Chapter 4:Unit 4.19.

Orr, H. T. and Zoghbi, H. Y. (2007). Trinucleotide repeat disorders. *Annu Rev Neurosci*, 30:575–621.

Ozsolak, F. and Milos, P. M. (2011). RNA sequencing: advances, challenges and opportunities. *Nat Rev Genet*, 12(2):87–98.

Pan, Q., Shai, O., Lee, L. J., Frey, B. J., and Blencowe, B. J. (2008). Deep surveying of alternative splicing complexity in the human transcriptome by high-throughput sequencing. *Nat Genet*, 40(12):1413–5.

Park, P. J. (2009). ChIP-seq: advantages and challenges of a maturing technology. *Nat Rev Genet*, 10(10):669–80.

Pati, A., Heath, L. S., Kyrpides, N. C., and Ivanova, N. (2011). ClaMS: A Classifier for Metagenomic Sequences. *Stand Genomic Sci*, 5(2):248–253.

Pauli, A., Valen, E., Lin, M. F., Garber, M., Vastenhouw, N. L., Levin, J. Z., Fan, L., Sandelin, A., Rinn, J. L., Regev, A., and Schier, A. F. (2012). Systematic identification of long noncoding RNAs expressed during zebrafish embryogenesis. *Genome Res*, 22(3):577–91.

Peano, C., Pietrelli, A., Consolandi, C., Rossi, E., Petiti, L., Tagliabue, L., De Bellis, G., and Landini, P. (2013). An efficient rRNA removal method for RNA sequencing in GC-rich bacteria. *Microb Inform Exp*, 3(1):1.

Pfeiffer, F., Schuster, S. C., Broicher, A., Falb, M., Palm, P., Rodewald, K., Ruepp, A., Soppa, J., Tittor, J., and Oesterhelt, D. (2008). Evolution in the laboratory: the genome of Halobacterium salinarum strain R1 compared to that of strain NRC-1. *Genomics*, 91(4):335–346.

Rabani, M., Levin, J. Z., Fan, L., Adiconis, X., Raychowdhury, R., Garber, M., Gnirke, A., Nusbaum, C., Hacohen, N., Friedman, N., Amit, I., and Regev, A. (2011). Metabolic labeling of RNA uncovers principles of RNA production and degradation dynamics in mammalian cells. *Nat Biotechnol*, 29(5):436–42.

Richard, H., Schulz, M. H., Sultan, M., Nurnberger, A., Schrinner, S., Balzereit, D., Dagand, E., Rasche, A., Lehrach, H., Vingron, M., Haas, S. A., and Yaspo, M. L. (2010). Prediction of alternative isoforms from exon expression levels in RNA-Seq experiments. *Nucleic Acids Res*, 38(10):e112.

Robert, C. and Watson, M. (2015). Errors in RNA-Seq quantification affect genes of relevance to human disease. *Genome Biol*, 16:177.

Rutkowski, A. J., Erhard, F., L'Hernault, A., Bonfert, T., Schilhabel, M., Crump, C., Rosenstiel, P., Efstathiou, S., Zimmer, R., Friedel, C. C., and Dolken, L. (2015). Widespread disruption of host transcription termination in HSV-1 infection. *Nat Commun*, 6:7126.

Sanger, F., Nicklen, S., and Coulson, A. R. (1977). DNA sequencing with chain-terminating inhibitors. *Proc Natl Acad Sci U S A*, 74(12):5463–7.

Schwanhausser, B., Busse, D., Li, N., Dittmar, G., Schuchhardt, J., Wolf, J., Chen, W., and Selbach, M. (2011). Global quantification of mammalian gene expression control. *Nature*, 473(7347):337–42.

Schweiger, M.-R., Ottinger, M., You, J., and Howley, P. M. (2007). Brd4-independent transcriptional repression function of the papillomavirus e2 proteins. *J Virol*, 81(18):9612–9622.

Shalem, O., Dahan, O., Levo, M., Martinez, M. R., Furman, I., Segal, E., and Pilpel, Y. (2008). Transient transcriptional responses to stress are generated by opposing effects of mRNA production and degradation. *Mol Syst Biol*, 4:223.

Sherry, S. T., Ward, M. H., Kholodov, M., Baker, J., Phan, L., Smigielski, E. M., and Sirotkin, K. (2001). dbSNP: the NCBI database of genetic variation. *Nucleic Acids Res*, 29(1):308–11.

Shumway, M., Cochrane, G., and Sugawara, H. (2010). Archiving next generation sequencing data. *Nucleic Acids Res*, 38(Database issue):D870–1.

Simpson, J. T., Wong, K., Jackman, S. D., Schein, J. E., Jones, S. J., and Birol, I. (2009). ABySS: a parallel assembler for short read sequence data. *Genome Res*, 19(6):1117–23.

Sirbu, A., Kerr, G., Crane, M., and Ruskin, H. J. (2012). RNA-Seq vs dual- and single-channel microarray data: sensitivity analysis for differential expression and clustering. *PLoS One*, 7(12):e50986.

Smit, A., Hubley, R., and Green, P. (1996). RepeatMasker Open-3.0. 1996-2010 http://www.repeatmasker.org.

Smith, T. F. and Waterman, M. S. (1981). Identification of common molecular subsequences. *J Mol Biol*, 147(1):195–7.

Stark, M., Berger, S. A., Stamatakis, A., and von Mering, C. (2010). Mltreemap–accurate maximum likelihood placement of environmental dna sequences into taxonomic and functional reference phylogenies. *BMC Genomics*, 11:461.

Stein, L. D. (2010). The case for cloud computing in genome informatics. *Genome Biol*, 11(5):207.

Stephens, Z. D., Lee, S. Y., Faghri, F., Campbell, R. H., Zhai, C., Efron, M. J., Iyer, R., Schatz, M. C., Sinha, S., and Robinson, G. E. (2015). Big Data: Astronomical or Genomical? *PLoS Biol*, 13(7):e1002195.

Sultan, M., Schulz, M. H., Richard, H., Magen, A., Klingenhoff, A., Scherf, M., Seifert, M., Borodina, T., Soldatov, A., Parkhomchuk, D., Schmidt, D., O'Keeffe, S., Haas, S., Vingron, M., Lehrach, H., and Yaspo, M. L. (2008). A global view of gene activity and alternative splicing by deep sequencing of the human transcriptome. *Science*, 321(5891):956–60.

Tang, F., Barbacioru, C., Wang, Y., Nordman, E., Lee, C., Xu, N., Wang, X., Bodeau, J., Tuch, B. B., Siddiqui, A., Lao, K., and Surani, M. A. (2009). mRNA-Seq whole-transcriptome analysis of a single cell. *Nat Methods*, 6(5):377–82.

Trapnell, C., Pachter, L., and Salzberg, S. L. (2009). TopHat: discovering splice junctions with RNA-Seq. *Bioinformatics*, 25(9):1105–11.

Trapnell, C. and Salzberg, S. L. (2009). How to map billions of short reads onto genomes. *Nat Biotechnol*, 27(5):455–7.

Valles, S. M., Oi, D. H., Yu, F., Tan, X.-X., and Buss, E. A. (2012). Metatranscriptomics and pyrosequencing facilitate discovery of potential viral natural enemies of the invasive Caribbean crazy ant, Nylanderia pubens. *PLoS One*, 7(2):e31828.

van Dijk, E. L., Auger, H., Jaszczyszyn, Y., and Thermes, C. (2014). Ten years of next-generation sequencing technology. *Trends Genet*, 30(9):418–26.

Van Noorden, R., Maher, B., and Nuzzo, R. (2014). The top 100 papers. *Nature*, 514(7524):550–3.

Vanin, E. F. (1985). Processed pseudogenes: characteristics and evolution. *Annu Rev Genet*, 19:253–72.

Venter, J. C., Adams, M. D., Myers, E. W., Li, P. W., Mural, R. J., Sutton, G. G., Smith, H. O., Yandell, M., Evans, C. A., Holt, R. A., et al. (2001). The sequence of the human genome. *Science*, 291(5507):1304–51.

Walboomers, J. M., Jacobs, M. V., Manos, M. M., Bosch, F. X., Kummer, J. A., Shah, K. V., Snijders, P. J., Peto, J., Meijer, C. J., and Muoz, N. (1999). Human papillomavirus is a necessary cause of invasive cervical cancer worldwide. *J Pathol*, 189(1):12–19.

Wang, K., Singh, D., Zeng, Z., Coleman, S. J., Huang, Y., Savich, G. L., He, X., Mieczkowski, P., Grimm, S. A., Perou, C. M., MacLeod, J. N., Chiang, D. Y., Prins, J. F., and Liu, J. (2010). MapSplice: accurate mapping of RNA-seq reads for splice junction discovery. *Nucleic Acids Res*, 38(18):e178.

Wang, R. S., Zhang, X. S., and Chen, L. (2007). Inferring transcriptional interactions and regulator activities from experimental data. *Mol Cells*, 24(3):307–15.

Wang, Z., Gerstein, M., and Snyder, M. (2009). RNA-Seq: a revolutionary tool for transcriptomics. *Nat Rev Genet*, 10(1):57–63.

Watson, J. D. and Crick, F. H. (1953). Molecular structure of nucleic acids; a structure for deoxyribose nucleic acid. *Nature*, 171(4356):737–8.

Westermann, A. J., Gorski, S. A., and Vogel, J. (2012). Dual RNA-seq of pathogen and host. *Nat Rev Microbiol*, 10(9):618–630.

Windhager, L., Bonfert, T., Burger, K., Ruzsics, Z., Krebs, S., Kaufmann, S., Malterer, G., L'Hernault, A., Schilhabel, M., Schreiber, S., Rosenstiel, P., Zimmer, R., Eick, D., Friedel, C. C., and Dolken, L. (2012). Ultrashort and progressive 4sU-tagging reveals key characteristics of RNA processing at nucleotide resolution. *Genome Res*, 22(10):2031–42.

Wu, T. D. and Nacu, S. (2010). Fast and SNP-tolerant detection of complex variants and splicing in short reads. *Bioinformatics*, 26(7):873–81.

Wu, T. D. and Watanabe, C. K. (2005). GMAP: a genomic mapping and alignment program for mRNA and EST sequences. *Bioinformatics*, 21(9):1859–75.

Xia, L. C., Cram, J. A., Chen, T., Fuhrman, J. A., and Sun, F. (2011). Accurate genome relative abundance estimation based on shotgun metagenomic reads. *PLoS One*, 6(12):e27992.

Xiong, X., Frank, D. N., Robertson, C. E., Hung, S. S., Markle, J., Canty, A. J., McCoy, K. D., Macpherson, A. J., Poussier, P., Danska, J. S., and Parkinson, J. (2012). Generation and analysis of a mouse intestinal metatranscriptome through Illumina based RNA-sequencing. *PLoS One*, 7(4):e36009.

Yang, J., Benyamin, B., McEvoy, B. P., Gordon, S., Henders, A. K., Nyholt, D. R., Madden, P. A., Heath, A. C., Martin, N. G., Montgomery, G. W., Goddard, M. E., and Visscher, P. M. (2010). Common SNPs explain a large proportion of the heritability for human height. *Nat Genet*, 42(7):565–9.

Yassour, M., Pfiffner, J., Levin, J. Z., Adiconis, X., Gnirke, A., Nusbaum, C., Thompson, D. A., Friedman, N., and Regev, A. (2010). Strand-specific RNA sequencing reveals extensive regulated long antisense transcripts that are conserved across yeast species. *Genome Biol*, 11(8):R87.

Yu, J., Cliften, P. F., Juehne, T. I., Sinnwell, T. M., Sawyer, C. S., Sharma, M., Lutz, A., Tycksen, E., Johnson, M. R., Minton, M. R., Klotz, E. T., Schriefer, A. E., Yang, W., Heinz, M. E., Crosby, S. D., and Head, R. D. (2015). Multi-platform assessment of transcriptional profiling technologies utilizing a precise probe mapping methodology. *BMC Genomics*, 16(1):710.

Yu, K. and Zhang, T. (2012). Metagenomic and metatranscriptomic analysis of microbial community structure and gene expression of activated sludge. *PLoS One*, 7(5):e38183.

Zhang, Z., Harrison, P. M., Liu, Y., and Gerstein, M. (2003). Millions of years of evolution preserved: a comprehensive catalog of the processed pseudogenes in the human genome. *Genome Res*, 13(12):2541–58.

zur Hausen, H. (2002). Papillomaviruses and cancer: from basic studies to clinical application. *Nat Rev Cancer*, 2(5):342–350.

Acknowledgements

At this point, I would like to express my deep appreciation to all the people who supported and encouraged me during my time as a PhD student. First of all, I am very grateful to my supervisor, Prof. Dr. Caroline C. Friedel, for giving me the opportunity to write this thesis. She was always open for valuable discussions and gave me the freedom to realize my own ideas. Without her helpful advice this work would not have been possible.

I also thank Dr. Gergely Csaba and Prof. Dr. Ralf Zimmer for a successful and great scientific cooperation. Furthermore, I would like to thank Franziska Schneider for many amusing conversations. Additionally, I am grateful to all lab members for a great time and for so many funny table soccer games during lunch break.

Moreover, I am grateful to Prof. Dr. Dmitrij Frishman for reviewing my thesis and to Prof. Dr. Christian Böhm and PD Dr. Matthias Schubert for participating my dissertation committee.

Finally, I would like to thank my family, girlfriend and friends for all the motivation and support they gave me during the last years.

Publications

Thomas Bonfert, Caroline C. Friedel.
Poly(A) cleavage site prediction from RNA-seq data.
Manuscript in preparation

Thomas Bonfert, Evelyn Kirner, Gergely Csaba, Ralf Zimmer, Caroline C. Friedel.
ContextMap 2: Fast and accurate context-based RNA-seq mapping.
BMC Bioinformatics, vol 16, pp. 122, 2015.

Andrzej J. Rutkowski, Florian Erhard, Anne L'Hernault, Thomas Bonfert, Markus Schilhabel, Colin Crump, Philip Rosenstiel, Stacey Efstathiou, Ralf Zimmer, Caroline C. Friedel, Lars Dölken.
Wide-spread disruption of host transcription termination in HSV-1 infection.
Nature Communications, vol 6, no. 7126, 2015.

Sabine Hünten, Markus Kaller, Friedel Drepper, Silke Oeljeklaus, Thomas Bonfert, Florian Erhard, Anne Dueck, Norbert Eichner, Caroline C. Friedel, Gunter Meister, Ralf Zimmer, Bettina Warscheid, Heiko Hermeking.
p53-regulated networks of protein, mRNA, miRNA and lncRNA expression revealed by integrated pSILAC and NGS analyses.
Molecular & Cellular Proteomics, vol 14, no. 10, pp. 2609, 2015

Thomas Bonfert, Gergely Csaba, Ralf Zimmer, Caroline C. Friedel.
Mining RNA-Seq Data for Infections and Contaminations.
PLoS ONE, vol 8, no. 9, pp. e73071, 2013.

Roland H. Friedel, Caroline C. Friedel, Thomas Bonfert, Roland Rad, Philippe Soriano.
Clonal expansion of transposon insertions identifies candidate cancer genes in a PiggyBac mutagenesis screen.
PLoS ONE, vol 8, no. 8, pp. e72338, 2013.

Thomas Bonfert, Gergely Csaba, Ralf Zimmer, Caroline C. Friedel.
A context-based approach to identify the most likely mapping for RNA-seq experiments.
BMC Bioinformatics, vol 13(Suppl 6), pp. S9, 2012.

Lisa Marcinowski, Michael Lidschreiber, Lukas Windhager, Martina Rieder, Jens B. Bosse, Bernd Rädle, Thomas Bonfert, Ildiko György, Miranda de Graaf, Olivia Prazeres da Costa, Philip Rosenstiel, Caroline C. Friedel, Ralf Zimmer, Zsolt Ruzsics, Lars Dölken.
Real-time Transcriptional Profiling of Cellular and Viral Gene Expression during Lytic Cytomegalovirus Infection.
PLoS Pathog, vol 8, no. 9, pp. e1002908, 2012.

Lukas Windhager, Thomas Bonfert, Kaspar Burger, Zsolt Ruzsics, Stefan Krebs, Stefanie Kaufmann, Georg Malterer, Anne L'Hernault, Markus Schilhabel, Stefan Schreiber, Philip Rosenstiel, Ralf Zimmer, Dirk Eick, Caroline C. Friedel, Lars Dölken.
Ultrashort and progressive 4sU-tagging reveals key characteristics of RNA processing at nucleotide resolution.
Genome Research, vol 22, pp. 2031, 2012.