Charles University

Faculty of Science

Study programme: Bioinformatics

Branch of study: BBINF

Tereza Čalounová

De novo transcriptomics and its use in non-model organisms

De novo transkriptomika a její využití u nemodelových organismů

Bachelor's thesis

Supervisor: Mgr. Tomáš Pluskal, Ph.D.

Prague, 2021

Prohlášení:

Prohlašuji, že jsem závěrečnou práci zpracovala samostatně a že jsem uvedla všechny použité informační zdroje a literaturu. Tato práce ani její podstatná část nebyla předložena k získání jiného nebo stejného akademického titulu.

V Praze, 13. 8. 2021 Tereza Čalounová

I would like to express my gratitude to my supervisor Mgr. Tomáš Pluskal, Ph.D. for his helpfulness, guidance, immediate feedback and having patience with me during the process of writing this thesis. I will be forever indebted to him for taking his time away during the very happy moments happening at the time of my thesis deadline. I am very grateful to him for giving me the opportunity to explore new areas, gain knowledge, perspectives and skills. I have learnt a lot from this experience.

A huge thanks goes to the awesome Tito Damiani. His dedication to help me was incredible. I hope I can pay it back one day!

Another big thanks goes to Martin Mokrejš for his very comprehensive in-depth critique of my thesis which I really appreciate. It definitely opened my eyes to see the issues in my text but also in *de novo* transcriptomics in general.

I would also like to express my gratitude to my family and friends for always being caring and extremely supportive. I am very lucky to have them.

Last but not least, I would like to thank Evžen Wybitul for his constructive feedback, helpful suggestions and cheering me up in my emotional local minima.

Abstract

The rise of second generation sequencing enabled the study of non-model organisms. Without the requirement of having a reference genome, *de novo* transcriptomics allows the study of functional elements of their genomes. That way, the great complexity of non-model organisms can be explored.

This thesis gives a comprehensive overview of the *de novo* transcriptomics experiment workflow from a bioinformatics perspective. The emphasis was placed on both theoretical background and practical approaches. This work also highlights new methods in *de novo* transcriptomics which may start to dominate in the near future. The practical part of the work presents *transXpress* — a *de novo* transcriptome assembly and annotation pipeline. Its use is demonstrated on a non-model plant long pepper (*Piper longum*) with medicinal potential.

Keywords: transcriptomics, *de novo* transcriptomics, transcriptome, RNA-Seq, non-model organism, assembly

Abstrakt

Rozvoj sekvenování druhé generace umožnil studium nemodelových organismů. Bez nutnosti mít referenční genom k dispozici, *de novo* transkriptomika umožňuje studium funkčních elementů genomů. Díky tomu je možné zkoumat komplexitu nemodelových organismů.

Tato práce poskytuje ucelený přehled kroků *de novo* studia transkriptomů z pohledu bioinformatiky. Důraz byl kladen na teoretické základy i na praktické přístupy. Práce rovněž představí nové metody *de novo* transkriptomiky, které mohou v blízké budoucnosti začít dominovat. Praktická část práce představuje *transXpress* — pipeline pro *de novo* sestavování transkriptomů a jejich anotaci. Jeho použití je ukázáno na nemodelové rostlině pepřovníku dlouhém (*Piper longum*), který má medicinální potenciál.

Klíčová slova: transkriptomika, *de novo* transkriptomika, transkriptom, RNA-Seq, nemodelový organismus, assembly

Table of contents

Abbreviations

DNA	Deoxyribonucleic acid
RNA	Ribonucleic acid
cDNA	Complementary DNA
rRNA	Ribosomal RNA
mRNA	Messenger RNA
bp	Base pairs
SGS	Second generation sequencing
TGS	Third generation sequencing
RNA-Seq	RNA-sequencing
ONT	Oxford nanopore technologies
PacBio	Pacific Biosciences
SMRT	Single-molecule real-time
CLR	Continuous long read
CCS	Circular consensus sequence
GC	Guanine-cytosine
QC	Quality control
DE	Differential expression
BUSCO	Benchmarking universal single-copy orthologs
EM	Expectation-maximization
EST	Expressed sequence tag

Introduction

It is widely acknowledged that several significant achievements in biology during the 20th century were in large part thanks to the use of the so-called "model organisms". Broadly speaking, a model organism is "a simplified, tractable system that is inherently convenient to study a particular area of biology" (Russell *et al.*, 2017). Without doubt, the sharp focus on a handful of model organisms has paid off over the last century. However, while convenient for studying many aspects of biology, model organisms are not necessarily the best systems for all possible questions. On the other hand, it is estimated that there are 8 to 10 million species of plants and animals world-wide (Sweetlove, 2011), with approximately 1.5 million currently documented in the Catalogue of Life (Sullivan, 2015). It goes without saying that such (bio)diversity cannot be fully captured by a few dozen of model organisms.

A major breakthrough in this regard has been the advent of second generation sequencing (SGS; also known as next-generation sequencing), which enabled the study of non-model organisms (Bräutigam and Gowik, 2010). Although SGS technologies were initially employed to study whole genomes, currently the most common application of SGS in non-model species is transcriptome characterization (Ekblom and Galindo, 2011). "Transcriptome" is defined as the complete set of RNA transcripts in a cell (Wang, Gerstein and Snyder, 2009). Accordingly, transcriptome analysis (a.k.a. "transcriptomics", "RNA sequencing" or "RNA-seq") is the study of the transcriptome using high-throughput SGS methods. A main distinction can be made according to the availability, or not, of a reference genome (i.e., reference-based vs *de novo* transcriptomics). Whereas reference-based transcriptomics is mainly limited to model organisms, *de novo* transcriptomics can be used for the study of non-model organisms with genomic sequences that are yet to be determined (Wang, Gerstein and Snyder, 2009).

The aim of this work is to provide a comprehensive overview of a *de novo* transcriptomics experiment workflow. The work is divided into two main sections — theoretical and practical.

The theoretical section will review all steps of the *de novo* transcriptomics experiment workflow with the emphasis on both theoretical background and practical solutions giving examples of existing tools. I will also highlight future directions by using newly developed technologies.

In the practical section, *transXpress de novo* transcriptomics pipeline, to which I contributed, will be described as a user-friendly way to perform *de novo* transcriptome assembly and annotation. It will be demonstrated on an example of a non-model plant of medicinal interest, *Piper longum*.

1. *De novo* transcriptomics

The central dogma of molecular biology stated by Francis Crick in 1957 (Crick, 1958) outlines the flow of information from DNA to protein with messenger RNA (mRNA) as a mediator of information. Since eukaryotic genomes are by large proportion composed of non-coding sequences (98% in case of humans) (Santosh, Varshney and Yadava, 2015), the transcriptomic study of only the protein-coding portion (i.e., mRNA sequences) represents an efficient and fundamental approach in functional genomics. It tells us which genes are "turned on" and being transcribed as mRNA, molecules that carry instructions for making proteins. Therefore, although transcriptomes, in the broad sense, account for all types of transcripts, transcriptomic studies usually focus on the protein-coding transcripts. In this work, the term transcriptome is mainly used to refer to the repertoire of protein-coding transcripts, i.e., mRNA molecules.

Without the requirement of having a reference genome, d*e novo* transcriptomics enables the study of transcriptomes of non-model organisms (Martin and Wang, 2011). Using the currently established sequencing technology (i.e. second generation sequencing), it is not possible to capture transcripts in their native form. In fact, only small fragments of the transcripts get sequenced and these millions of fragments of the "transcriptome puzzle" need to be assembled back together. With the advent of a new technology (i.e. third generation sequencing), this difficult and error-prone step of reconstruction can be avoided. In any case, the ultimate goal is to collect a complete set of transcripts.

A premise that one gene corresponds to one transcript does not hold. The reason for that is a phenomenon called alternative splicing. Nascent transcript (i.e. pre-mRNA) resulting from a gene transcription consists of protein coding (i.e. exons) and non-coding (i.e. introns) parts. Pre-mRNA undergoes a process called splicing, in which the introns are removed, resulting in the mature mRNA molecule. However, splicing can create a range of transcripts by, for example, skipping an exon or including an intron. These different transcripts corresponding to a single gene are called isoforms. All of them should, ideally, be included in the studied transcriptome.

Transcriptome as a catalogue of all protein-coding transcripts within an organism is a rich source of information. It is up to the researcher and possible applications how this information can be further used. A few examples of applications in recent research are provided in chapter 1.2.

1.1. History and current state

The term 'transcriptome' was coined by Charles Auffray in 1996 (Piétu *et al.*, 1999), although transcriptomes had been studied since the first publication in 1979 (Sim *et al.*, 1979).

First attempts to study transcriptomes were using EST (expressed sequence tag) sequencing. This method sequences short sections (200-800 bp) of complementary DNA (cDNA) called tags that are then mapped to a genome to identify genes from which the transcripts originated (Parkinson and Blaxter, 2009).

In the mid 1990s, microarray technology took over sequencing based transcriptomics and dominated until the advent of RNA sequencing (RNA-Seq) in the mid 2000s (McGettigan, 2013). The microarray technique is based on hybridization of fluorescently labelled cDNA to an array of complementary RNA probes corresponding to individual genes. Microarray technology allowed higher throughput and lower cost compared to EST sequencing (Kukurba and Montgomery, 2015).

In 2006, the first RNA-Seq paper was published (Bainbridge *et al.*, 2006) and microarrays were quickly superseded by the second generation sequencing (SGS) approach (McGettigan, 2013). Since 2010, RNA-Seq has dominated transcriptomics and is rising as demonstrated by the increasing number of publications in PubMed illustrated in Figure 1.

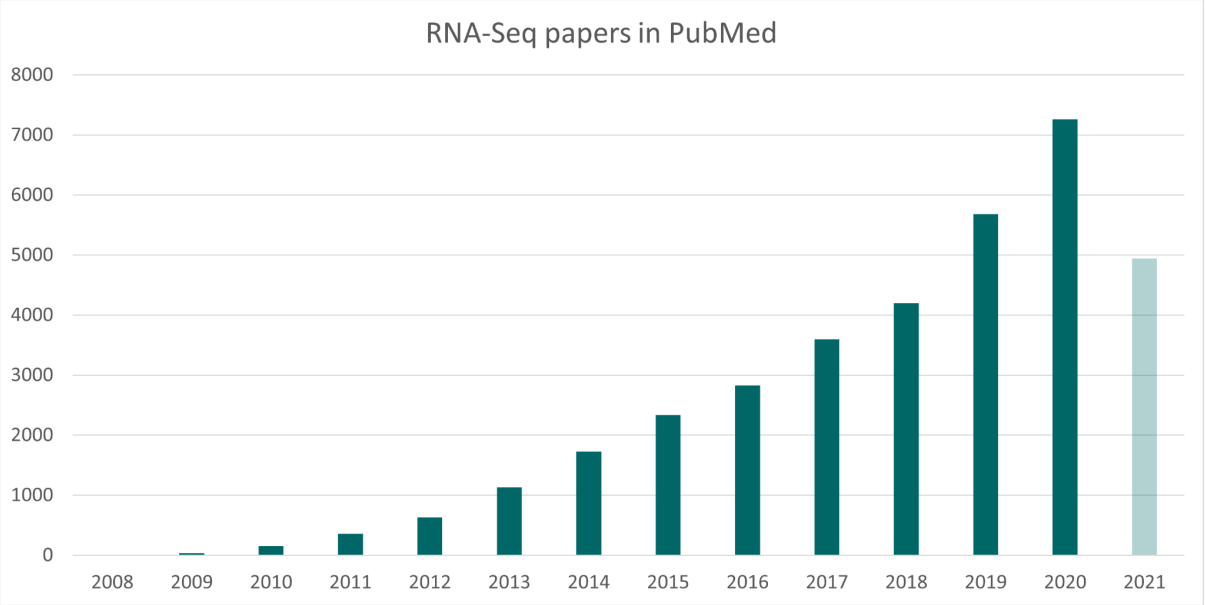

Figure 1. Number of papers containing the "RNA-Seq" keyword in PubMed by year (as of August 2021).

RNA-Seq allows obtaining complete transcript sequences and is the only of these methods which can be used without already knowing the organism's genome sequence — *de novo*.

The "golden standard" sequencing platform for RNA-Seq is Illumina Inc., formerly Solexa Inc. By sequencing fragments of cDNA, it produces short accurate reads. From these reads, the original transcripts need to be computationally reconstructed — assembled.

In 2009 a new sequencing technology producing long reads was introduced by Pacific Biosciences (PacBio) (Eid *et al.*, 2009), followed by another long read-producing technology by Oxford Nanopore Technologies (ONT). ONT allows direct sequencing of not only the cDNA but also native RNA (Garalde *et al.*, 2018). These long read technologies are referred to as third generation sequencing (TGS). As their long read lengths allow capturing full-length transcripts, they seem to be a very promising technology for *de novo* transcriptomics by eliminating the difficult assembly step.

1.2. Use in the study of non-model organisms

Before the rise of RNA-Seq, investigating genetic information of non-model species was expensive, time consuming and laborious (Bräutigam and Gowik, 2010). RNA-Seq enabled the study of functional elements of genomes at a lower cost since only transcribed regions, which usually comprise a vast minority of each genome, are sequenced. Because genome information is available only for a small fraction of species, a reference-based approach cannot be used. Even if reference genomes do exist, they are not as accurate and complete as those for model organisms, which have been polished over the years (Fu *et al.*, 2018). Either *de novo* genome sequencing and assembly or *de novo* transcriptome sequencing and assembly are possible strategies. However, the complexity and size of genomes of most non-model species represent a limiting factor for their reconstruction. On the other hand, *de novo* transcriptomics represents an affordable strategy for studying the genetic repertoire of non-model organisms.

Exploring transcriptomes of non-model organisms will help to gain a more complex view on biology. To illustrate this, I selected a few examples of *de novo* transcriptomics studies using non-model organisms.

Gray whales, *Eschrichtius robustus*, are among the top 1% longest-lived mammals (Toren *et al.*, 2020). Studying aging is one of the top fields of interest in biomedical research. *De novo* transcriptomics can be applied to studying these long-lived non-model mammals. In a transcriptomic study of gray whales, longevity adaptations associated with DNA repair and ubiquitination were discovered (Toren *et al.*, 2020).

Yet another field of biomedical research interest is regeneration. Lizards are able to lose and regenerate tails. Transcriptomic analysis of lizard *Anolis carolinensis* revealed genes involved in repair mechanisms. These genes differed from those reported in traditional animal models used to study regeneration (Hutchins *et al.*, 2014).

Study of a rattlesnake, *Crotalus adamanteus*, identified genes expressed in its venom gland. Snake venom components have a potential to be used as drugs (Rokyta *et al.*, 2012).

Plants produce a variety of interesting compounds. Not surprisingly, many of them have medicinal potential. Nonetheless, these compounds often have very complex structures which makes it difficult to synthesize them in the lab. Using *de novo* transcriptomics, biosynthetic pathways of these compounds can be discovered, genes encoding enzymes involved in the pathway then get cloned and transferred to a host organism. The pathway can be further optimized to give higher yields of the compound of interest (Owen *et al.*, 2017).

Transcriptomic study on a mayapple plant, *Podophyllum hexandrum*, discovered a biosynthetic pathway of podophyllotoxin, a molecule that is used as a precursor of the chemotherapeutic drug etoposide. It enabled large-scale production of the etoposide aglycone in tobacco plants (Lau and Sattely, 2015).

1.3. Workflow

A *de novo* transcriptomics experiment usually consists of 8 steps (See Figure 2): 1) RNA isolation and purification, 2) library preparation, 3) RNA-Seq, 4) RNA-Seq quality control, 5) preprocessing of sequencing reads, 6) *de novo* assembly, 7) assembly quality evaluation and 8) downstream analysis. Steps 2) to 8) will be described in detail in the following chapters.

Figure 2. Typical steps of a de novo transcriptomics experiment.

The experiment begins with the isolation of RNA from a sample of interest. The most abundant type of RNA is ribosomal RNA (rRNA). rRNA usually constitutes over 90% of the total RNA in the cell (Hallberg and Bruns, 1976). On the other hand, transcriptomic experiments are typically interested in coding RNA molecules. There are two standard methods addressing this — capturing of messenger RNA (mRNA) using poly(A) selection or rRNA depletion (Conesa *et al.*, 2016).

2. RNA sequencing (RNA-Seq)

Historically, DNA sequencing was developed by Frederick Sanger and colleagues in 1977 (Sanger, Nicklen and Coulson, 1977). We now refer to it as Sanger sequencing (or first generation sequencing). In the mid 2000s, Sanger sequencing was superseded by second generation sequencing. With today's technology there is a possibility to either use the established short read second generation sequencing (SGS) or the new long read third generation sequencing (TGS).

RNA Sequencing (RNA-Seq) enables us to determine the sequences of RNA molecules in a sample by sequencing their corresponding cDNA. For designing an RNA-Seq experiment, there are several considerations which need to be taken because they have an impact on the quality of the resulting assembly and subsequent analyses. The quality of input data has more impact on the quality of the assembly than the assembly algorithm used (Smith-Unna *et al.*, 2016).

The sequencing reads are stored in a file in FASTQ format. This format consists of the read sequences and corresponding quality scores (Phred) (Ewing and Green, 1998; Ewing *et al.*, 1998) for every base in the sequence. The quality score is encoded as an ASCII character and represents the predicted probability of a correct base call (Cock *et al.*, 2010).

2.1. Comparison of short read and long read sequencing

Short read sequencing, represented primarily by Illumina, has been a golden standard sequencing technology for *de novo* transcriptomics. This second generation sequencing technique produces reads of short lengths (typically 100 or 150 bp) in both high quantity and quality.

Illumina sequencing technology is based on sequencing-by-synthesis. cDNA fragments with ligated adapter sequences are immobilized on a flow cell. Every cDNA fragment is amplified to form clusters of the same sequences. Subsequently these fragments are sequenced by incorporation of fluorescently-labeled nucleotides. When a labeled nucleotide is incorporated, the fluorescent tag is cleaved off and produces a signal.

These signals are detected for every cluster in parallel, providing information about the base composition (Metzker, 2010).

The main limitation of SGS reads is their short length, with which it is not possible to capture full-length transcripts. Full-length transcripts have to be reconstructed (assembled) *in silico*. Despite this disadvantage, the high accuracy and quantity of SGS reads make them very suitable for transcript quantification and differential expression analysis (Stark, Grzelak and Hadfield, 2019).

The third generation sequencing technologies are represented by Pacific Biosciences (PacBio) single-molecule real-time (SMRT) sequencing and Oxford Nanopore Technologies (ONT) sequencing. A common denominator of these technologies is the long read length sufficient to capture full-length transcripts, sequencing in real time but unfortunately also high error rate, higher price per base and lower throughput compared to SGS (Weirather *et al.*, 2017; Stark, Grzelak and Hadfield, 2019).

With the PacBio sequencing technology, a cDNA molecule to be sequenced is prepared by ligating hairpin adapters to the ends of the molecule. That way a circular template molecule called SMRTbell is created. A DNA polymerase binds to this template molecule and traverses it in several passes during which signals are emitted and detected in real time. It creates a continuous long read (CLR) consisting of subreads and the adapter sequences in between. These subreads are then aligned to produce a circular consensus sequence (CCS) reads[1] (Rhoads and Au, 2015). This step reduces the error rate from approximately 13% down to 0.1% depending on the number of passes, the length of the transcript and the longevity of the polymerase. 4 passes are estimated to be needed for 99% accuracy (Q20 - Phred quality score 20), 9 passes for 99.9% accuracy (Q30) (Amarasinghe *et al.*, 2020).

Oxford Nanopore Technologies (ONT) sequencing is a different sequencing method, based on measuring ionic fluctuations as a DNA or RNA molecule traverses through a nanopore in a membrane. Similarly to SMRTbell, molecules to be sequenced are also prepared for sequencing by hairpin adapter ligation. The molecule and its reverse complement are bound to a hairpin linker. As this template molecule traverses through

[1] also known as “HiFi” read

the nanopore, both the original molecule and its reverse complement are sequenced. From these 2 reads, a consensus 2D read is made. This consensus step improves the base calling accuracy. Error rate of ONT 2D reads is around 10-15% (Weirather *et al.*, 2017). A similar idea of CCS has also been applied to ONT sequencing with the INC-seq protocol, further improving the accuracy. The INC-seq protocol uses rolling circle amplification to produce long cDNA molecules consisting of repeats of the original template. These molecules are then sequenced and consensus of the repeats is created (Li *et al.*, 2016).

Taken together, long read sequencing represents a very promising direction for future *de novo* transcriptomics. Long read sequencing virtually eliminates the assembly step and thus allows performing downstream analysis directly on the input data. However, TGS is still in the early stage of development, the technology has to be further improved and tools for long read RNA-Seq need to be developed. Low throughput is a limitation of using long reads for quantification and differential expression analyses. If differential expression is the goal of the study, short read sequencing is preferred. Short reads and long reads can also be combined in a hybrid approach, taking advantage of the pros of both SGS and TGS (Fu *et al.*, 2018; Hölzer and Marz, 2019; Prjibelski *et al.*, 2020).

2.2. Library preparation and considerations

Library preparation for sequencing differs based on the design of the RNA-Seq experiment and the used technologies. Common steps for short read sequencing library preparation involve RNA fragmentation, conversion to complementary DNA (cDNA), adapter ligation and amplification. For long-read sequencing no fragmentation is done, RNA may or may not be converted to cDNA based on the used sequencing platform (ONT allows both RNA and cDNA sequencing whereas with PacBio only cDNA can be sequenced), amplification may or may not be done (Stark, Grzelak and Hadfield, 2019).

For short-read sequencing, single-end or paired-end sequencing is possible. In single-end sequencing the read corresponds to one end of a cDNA fragment whereas in paired-end sequencing there are two reads corresponding to a cDNA fragment, one for every end. For *de novo* transcriptome assembly, paired-end reads are preferred over single-end reads because they help the assembly by giving the information how far

apart the read pairs are (Grabherr *et al.*, 2011; Conesa *et al.*, 2016). The insert size of paired end reads can have an effect on the connectivity of the transcriptome assemblies, with longer insert sizes yielding higher connectivity (Hara *et al.*, 2015).

Some RNA-Seq protocols (denoted as "stranded") retain the information about which strand the transcript originates from, either by using a different adapter for the strand before reverse transcription or by chemical modification (Levin *et al.*, 2010). Strand-specific sequencing is preferred to non-strand-specific in case of *de novo* transcriptomics (Conesa *et al.*, 2016), because different coverage of forward and reverse reads can differentiate between sense and antisense transcripts as illustrated in Figure 3 (Stark, Grzelak and Hadfield, 2019).

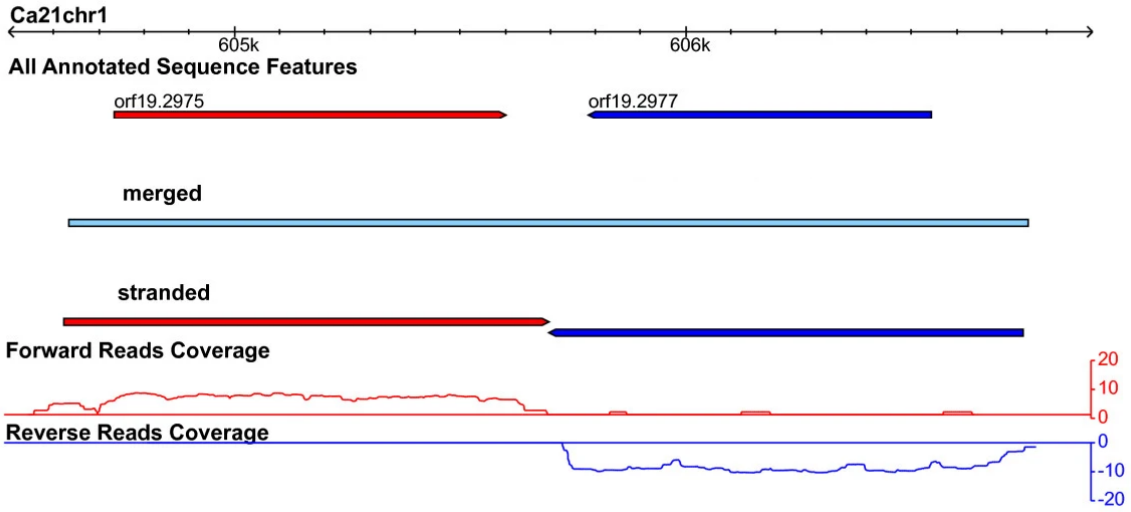

Figure 3. Strand specificity can help to differentiate between sense and antisense transcripts. At the top there are sense (red) and antisense (blue) transcripts, at the very bottom the coverage of forward (red) and reverse (blue) reads is shown. When this stranded information is available, sense and antisense transcripts can be correctly reconstructed (above the read coverage separate red and blue bars), otherwise transcripts would get mistakenly merged (light blue in the middle of the figure). Adapted from (Martin et al., 2010), with permission.

Another short read library consideration is the read length. Longer lengths of reads are preferable because they contain more information, are less ambiguous than the shorter ones and enable identification of longer transcripts. (Stark, Grzelak and Hadfield, 2019)

When the goal of a transcriptomic experiment is not a differential expression analysis but rather cataloging transcripts, library normalization may be considered. The most common cDNA library normalization techniques use duplex-specific nuclease (DSN) (Ekblom *et al.*, 2012). Their main idea is to denature double-stranded cDNA and let it partially re-anneal in the presence of DSN. Because most abundant cDNA re-anneals the fastest, it gets digested leading to a decrease of its relative abundance (Zhulidov *et al.*, 2004). That way, less abundant transcripts should have a higher chance of being captured during sequencing (Ekblom *et al.*, 2012). With that being said, studies advising both for and against library normalization were published (Ekblom *et al.*, 2012; Vijay *et al.*, 2013; Hoang *et al.*, 2019).

For PacBio sequencing libraries, it is recommended to perform size selection. Longer transcripts are less abundant and thus difficult to detect. Size selection increases their relative abundance (Stark, Grzelak and Hadfield, 2019). This, however, prevents their use for quantification and differential expression (Amarasinghe *et al.*, 2020).

2.3. Sequencing depth

Sequencing depth is a term used to express how many times a nucleotide in a transcript has been sequenced. Usually there is a relationship between the size of a genome and the complexity of a transcriptome. The larger the genome is, the more complex the transcriptome is. Thus, greater sequencing depth is needed (Wang, Gerstein and Snyder, 2009). The greater the sequencing depth, the more accurate the assembled transcriptome can be. In practice, however, budgets for sequencing are not unlimited and compromises have to be made.

Francis et al. state that for a whole-body animal transcriptome, 30 million Illumina-based reads are sufficient. For an individual organ transcriptome, 20 million reads are sufficient. With more than 60 million reads, not many additional lowly

expressed transcripts are discovered, whereas more sequencing errors accumulate and lead to assembly errors (Francis *et al.*, 2013).

When considering a tradeoff between the sequencing depth and the number of replicates (either technical or biological) in case of studying differential expression, it has been shown that increasing the number of replicates brings more power than increasing the sequencing depth (Rapaport *et al.*, 2013).

3. Quality control and preprocessing of RNA-Seq data

Before using the raw data for transcriptome assembly, it is a good practice to inspect its quality and preprocess it. With poor quality data one has to be cautious with the interpretation of subsequent results. The quality of the input data has a major impact on the quality of the resulting assembly (Smith-Unna *et al.*, 2016).

3.1. Quality control

FastQC (Andrews, 2010) is a tool for basic quality control (QC) steps. FastQC generates quality control statistics based on the sequence composition and Phred quality scores of the read sequences. MultiQC (Ewels *et al.*, 2016) can merge FastQC statistics reports of individual read files (samples) into a single report. Examples of FastQC QC modules include: per base sequence quality, per sequence quality scores, per base sequence content, per sequence GC (guanosine and cytosine) content, overrepresented sequences and sequence duplication levels.

Per base sequence quality module gives information about distribution of quality scores across bases positions. With good quality data it is expected to obtain a tight distribution of high qualities across all base positions. However, it is very common to see slightly lower quality at the starting positions 1 to 6 in Illumina-based reads (which likely is due to sequencer calibration) and general decrease of quality towards the ends of the reads. In case of Illumina sequencing, the fluorescent signal decays with each cycle due to fluorophore degradation over time and due to the phasing problem. As the sequencing process continues, the clusters get out of phase, the signal starts to blur and the confidence of a correct base call decreases. If the quality is too low at the end of the reads, it is possible to trim them. The example output is illustrated in Figure 4.

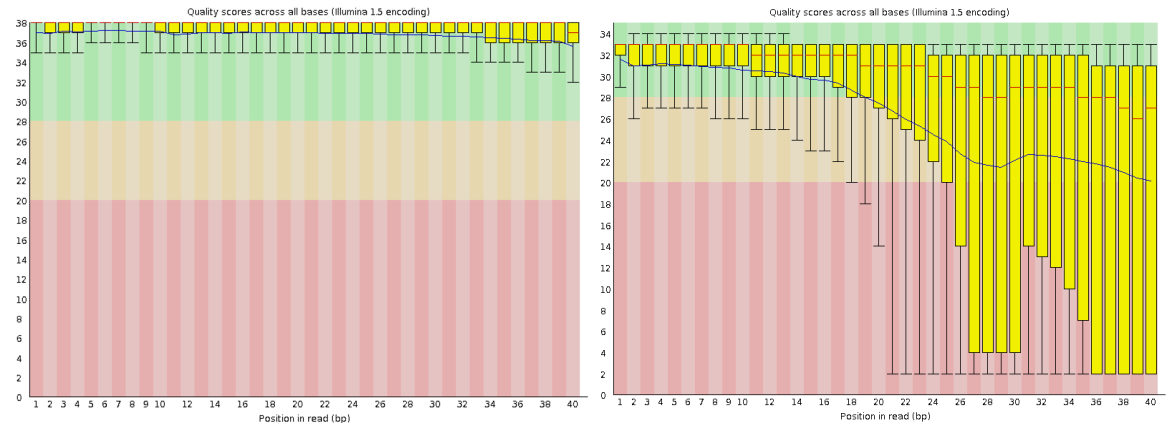

Figure 4. Quality score distributions at individual positions for high-quality sequencing library (on the left) and low-quality sequencing library (on the right). X axis = position of a nucleotide in the read (bp), Y axis = predicted Phred quality score. Phred quality interval over 28 is in green (good), between 20 and 28 is in orange (warning) and below 20 is in red (bad). For high-quality data, there is a tight distribution of high qualities on every base position. For bad quality data, the distribution gets looser and the mean Phred quality value decreases towards the end of the reads. Adapted from (Andrews, 2010).

FastQC also calculates the mean quality score for every sequence and plots this distribution. Example outputs for data of both good and bad quality are shown in Figure 5.

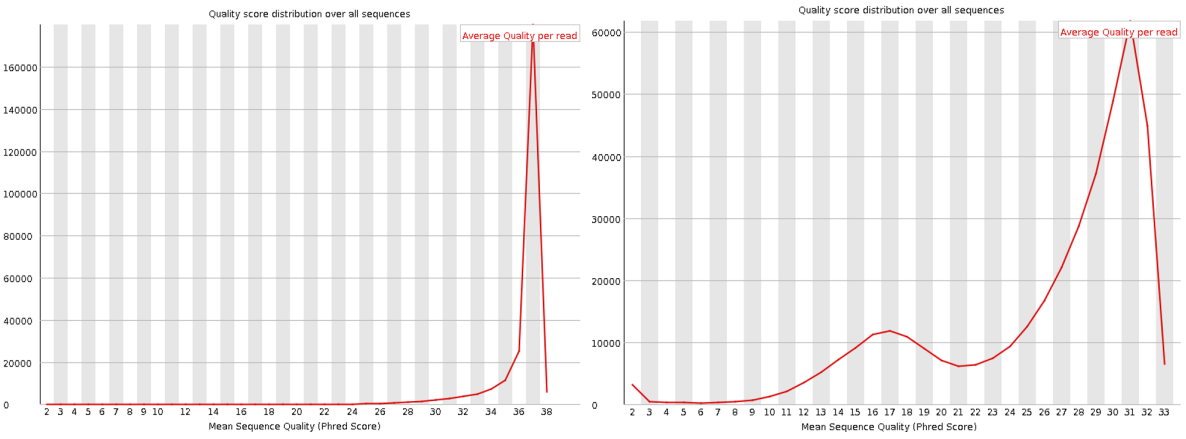

Figure 5. Quality score distribution for data of high quality (on the left) and low quality (on the right). X axis = mean sequence quality (Phred score), Y axis = number of sequences. With good quality data, in the distribution there is one tight peak at a high quality score (37). With bad quality data, there is a loose peak at quality score 30 and a secondary peak at quality score 17. Reads corresponding to the secondary peak may be filtered. Adapted from (Andrews, 2010).

In the per base content module, distributions of the four bases which do not change with the base position are expected. However, with RNA-Seq data it is sometimes common to see spikes at the starting positions (first 10 to 12 bases), illustrated in Figure 6 on the right. The reason for that is the way the sequencing library is generated. In the process there are random hexamer primers being used and they bias the beginning positions in the reads (Hansen, Brenner and Dudoit, 2010).

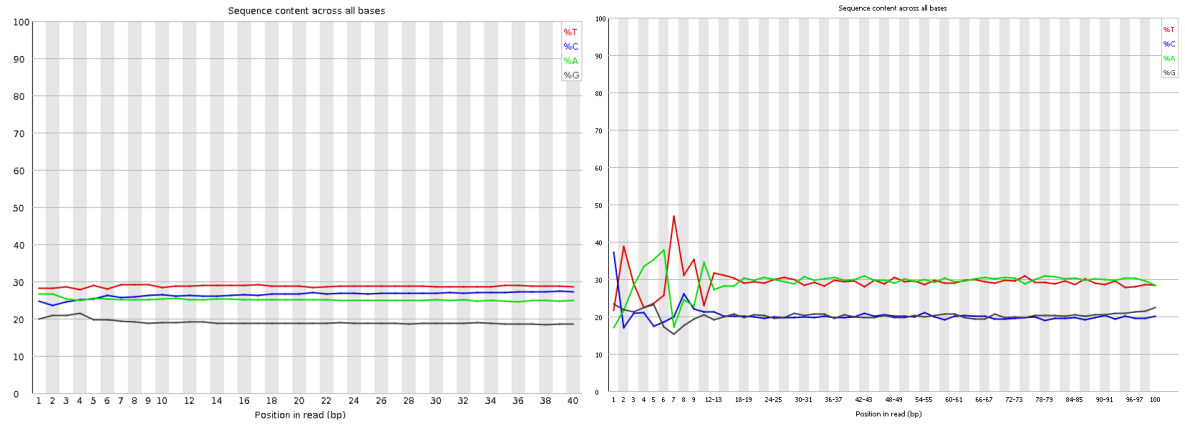

Figure 6. Distribution of bases in a high-quality sequencing library (on the left) and with bias at the beginning of the reads, possibly caused by random hexamer priming (on the right). X axis = position in read (bp), Y axis = percentage. Left Figure is adapted from (Andrews, 2010).

Another measure is the GC content of every read and the resulting distribution of these values. One should see a normal distribution (with the same mean and standard deviation as in the data). A secondary peak in the distribution can reveal possible contamination. Examples of both situations are shown in Figure 7. GC content differs amongst organisms, hence when having more reads with different GC content (creating a peak), it is possible they are resulting from a different organism such as a virus infecting a host of interest. Similarly, in the overrepresented sequences module there can be an overrepresented sequence originating from the viral genome. Alternatively, overrepresented sequences could be primers or adapters.

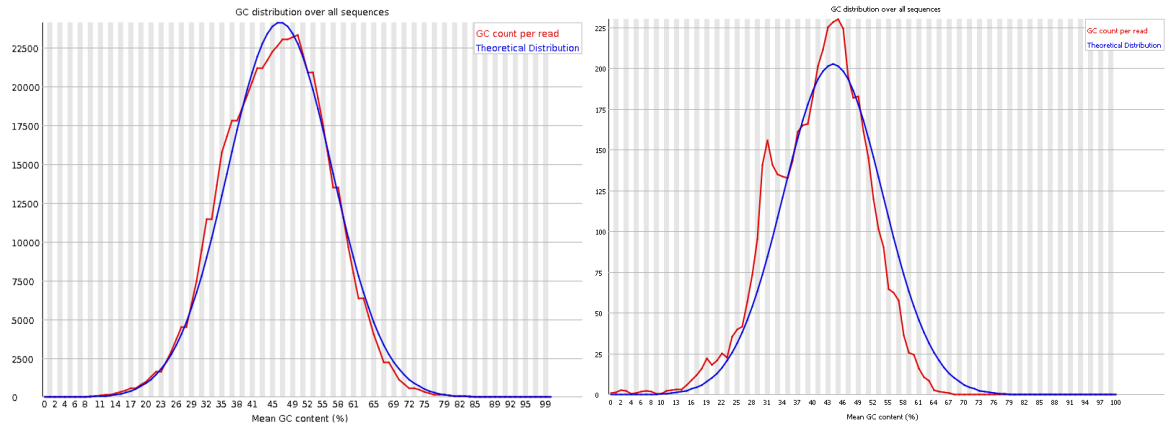

Figure 7. Distribution of GC content in reads with good-quality data (on the left) and distribution with a secondary peak which indicates possible contamination (on the right). X axis = mean GC content (%), Y axis = number of sequences. Blue curve = normal distribution, red curve = actual distribution. Left figure is adapted from (Andrews, 2010).

In the sequence duplication levels module, a warning may be triggered since some transcripts are hugely enriched. In RNA experiments some duplication is expected.

3.2. Read trimming and filtering

Preprocessing of the sequencing data is very important for *de novo* assembly. *De novo* assembly is based on finding overlaps and the presence of technical sequences such as adapters (due to the length of the sequenced fragment being shorter than the read length, it happens that the adapter gets sequenced too) would lead to incorrect overlaps and wrong assembly. Hence it is advisable to trim the adapters from the reads.

It can also be appropriate to trim the ends of the reads where the predicted Phred quality is below a certain threshold. However, it should be noted that short reads are less informative than long reads so in some situations it is not recommended to trim the reads too aggressively.

Reads with overall poor quality or a length, which is too short, can be filtered out.

Some of the most popular tools for trimming and filtering include AdapterRemoval v2 (Schubert, Lindgreen and Orlando, 2016), Cutadapt (Martin, 2011) and Trimmomatic (Bolger, Lohse and Usadel, 2014).

3.3. rRNA sequences removal

Despite using an RNA extraction protocol, which captures mRNA or depletes rRNA, it might be still possible that rRNA is present in the sequencing data. Prior to the assembly, filtering out these rRNA reads can be done. Tools for rRNA filtering include SortMeRNA (Kopylova, Noé and Touzet, 2012), rRNASelector (Lee, Yi and Chun, 2011), riboPicker (Schmieder, Lim and Edwards, 2012) and Meta-RNA 3 (Huang, Gilna and Li, 2009).

3.4. Correcting errors in long reads

Because long read sequencing has a rather high error rate, there is a need for correction of these errors. Although both ONT and PacBio produce long reads, their technology is different and therefore, the source and type of errors is different. For ONT long reads, the major types of errors are substitutions and deletions (including deletions of homopolymer stretches) but insertion errors are also not negligible. Whereas for PacBio CCS long reads, insertions are almost eliminated by the consensus step (Weirather *et al.*, 2017; Lima *et al.*, 2020).

There are two approaches for error correction based on the availability of short reads. If short reads are available, their mapping to the long reads can be used to correct the errors in them (referred to as a hybrid correction approach). Latter approach uses only long reads alone (referred to as a self-correction approach). Generally, hybrid error correction tools provide better results than self-correction ones (Zhang, Jain and Aluru, 2019; Lima *et al.*, 2020).

In error correction of specifically RNA-Seq long reads, very little work has been done and represents an area for future improvement. To this day and to my best knowledge, there is only one platform-independent tool for self-correction of long reads - Racon (Vaser *et al.*, 2017). This tool was originally developed for genomic long reads. When used in practise by Prjibelski et al., Racon removed a significant amount of data, which reduced the number of assembled transcripts (Prjibelski *et al.*, 2020).

PacBio long read self-correction tools include Iso-Seq 3 from the PacBio pipeline (*IsoSeq*, 2020), HGAP (originally developed for genomic long reads) (Chin *et al.*, 2013), ToFU (Gordon *et al.*, 2015) and IsoCon (Sahlin *et al.*, 2018). When short reads are available, LSC (Au *et al.*, 2012) hybrid correction can be used .

For ONT transcriptomic long reads, only a single method for self-correction has been published so far — isONcorrect (Sahlin and Medvedev, 2021). Many genomic ONT long read error correction tools have been developed, however their use on transcriptomic data has shown undesirable effects in resulting transcriptomes such as the high amount of discarded reads, a decrease in number of detected genes, a bias towards correction to the major isoform and loss of lowly expressed isoforms. Their comparison, effect on the transcriptomic data and recommendations for specific use cases is discussed in (Lima *et al.*, 2020).

4. *De novo* transcriptome assembly

Second generation sequencing cannot capture full-length transcripts. Its data comes in the form of short reads of original transcripts. In order to reconstruct these transcripts, reads are "glued" together based on their overlaps.

When a reference genome is available, this information is also used for the assembly. Reads are first mapped onto the genome and then neighboring overlapping reads get "glued".

For non-model species, the reference genome is typically unavailable so the overlapping reads have to be "glued" without a reference *de novo*, using approaches originating from graph theory.

4.1. Challenges of *de novo* transcriptome assembly

The task of reconstruction of a set of transcripts from small pieces (reads) of unknown original locations is itself very algorithmically challenging. Moreover, the number of reads is large which makes the task of assembly computationally challenging. Also, although short read sequencing is rather accurate, reads are not perfect and contain errors.

Yet another challenge that is specific to transcriptome assembly is the highly variable sequencing coverage (i.e. number of reads that align to the sequence) of different transcripts. The coverage can differ by 5 orders of magnitude between highly and lowly expressed transcripts (Martin and Wang, 2011). Taken together, reads with sequencing errors from highly expressed transcripts can be more abundant than error-free reads from lowly expressed transcripts (Grabherr *et al.*, 2011).

Another transcriptome-specific challenge is the presence of alternative splicing. One genomic locus can be a template of more isoforms which all need to be reconstructed (Grabherr *et al.*, 2011).

Transcripts can also have repetitive regions. These are difficult to reconstruct due to the short length of sequencing reads. Resolving repeats is one of the biggest challenges in genome assembly (Hölzer and Marz, 2019). Transcripts tend to be less repetitive compared to genomes but still they might contain repetitive regions difficult to reconstruct (Paszkiewicz and Studholme, 2010).

4.2. Algorithmic approaches to *de novo* transcriptome assembly

Unless using long read technologies, data obtained by sequencing represent short stretches of the original transcripts. The goal is to reconstruct these original transcripts. This problem of sequence reconstruction is called an assembly.

Approaches to solve the assembly problem are based on a field of mathematics called graph theory and employ structures called graphs.

Basic terminology from graph theory needed for characterizing assembly includes:

> **Graph** G is a pair (V, E) where V is a set of vertices and E is a set of edges between the vertices such that $E \subseteq \{(u,v) \mid u, v \in V\}$.
>
> **Directed graph** is a graph in which edges have orientation.
>
> **Path** in a graph is a sequence of edges which joins a sequence of distinct vertices.
>
> **Eulerian path** is a path which visits every edge of the graph exactly once.
>
> **Hamiltonian path** is a path which visits all vertices of the graph exactly once.

Historically, the first approach attempting to solve the assembly problem was using a special type of directed graph called an overlap graph.

> **Overlap graph** is a graph with vertices which correspond to sequences of symbols and edges between those vertices which overlap.

In the context of the assembly problem, each sequencing read is represented as a vertex and reads are connected if they overlap by some minimum criterion.

Such a criterion is usually determined by pairwise sequence alignment. Example of an overlap graph is shown in Figure 8.

Given such an overlap graph (vertices correspond to reads and edges to overlaps), the assembly problem of reconstructing sequence from which the reads originate can be formulated as finding a Hamiltonian path. The Hamiltonian path problem was proved to be NP-complete (Pevzner, Tang and Waterman, 2001).

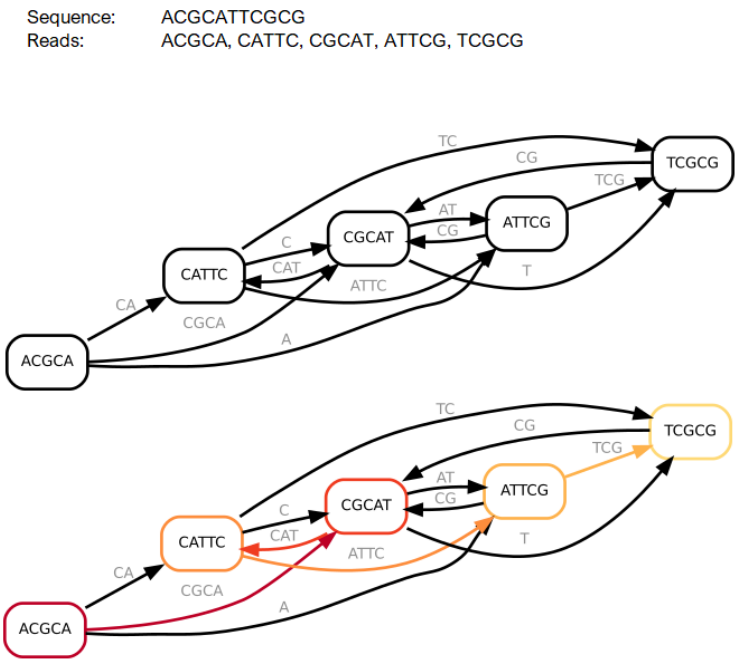

Figure 8. Example of an overlap graph corresponding to a given set of reads. In this example, the overlap criterion for connecting two sequences is exact overlap by at least one base. Highlighted path in the bottom graph corresponds to the original sequence (color gradient determines the order of vertices).

Later, the overlap graph approach was abandoned and superseded with a concept of using smaller parts of reads rather than reads alone (Pevzner, Tang and Waterman, 2001).

For each read, all subsequences of length k, called k-mers, are generated. The idea of using smaller k-mers (1) reduces required memory, (2) helps to reformulate the assembly problem and (3) better represents the composition of the original sequence.

(3) As explained by Compeau et al., given a set of reads of length 100 (100-mers), these represent a small portion of all possible 100-mers in the original sequence. However, if these reads are broken into smaller k-mers, nearly all k-mers from the original sequence get represented for small enough k (Compeau, Pevzner and Tesler, 2011).

Special type of graph using k-mers is called De Bruijn graph (in honour of its discoverer Nicolaas de Bruijn). There are essentially two formulations of the De Bruijn graph based on whether k-mer is represented as a node (see I further below) or as an edge (see II further below) (Miller, Koren and Sutton, 2010). The difference is illustrated in Figure 9.

> (I) **De Bruijn graph** is a directed graph where vertices correspond to k-mers. If two k-mers overlap by k-1 bases, they are connected with an edge.

(1) Since every k-mer is represented (stored) only once, this greatly reduces memory compared to storing every read in case of overlap graphs with enormous counts of sequencing reads produced by SGS.

> (II) **De Bruijn graph** is a directed graph where vertices correspond to (k-1)-mers, two (k-1)-mers are connected with an edge if there is a k-mer corresponding to concatenation of the first (k-1)-mer with last character of the other (k-1)-mer. In other words, for every k-mer vertices corresponding to its (k-1) bases long prefix and suffix are created and connected with an edge.

(2) The latter definition enables to formulate the assembly problem as an Eulerian path problem which can be solved in linear time. Although definition I requires to find a Hamiltonian path (NP-complete problem) and definition II requires to find a Eulerian path (solvable in linear time), both versions are used in practise.

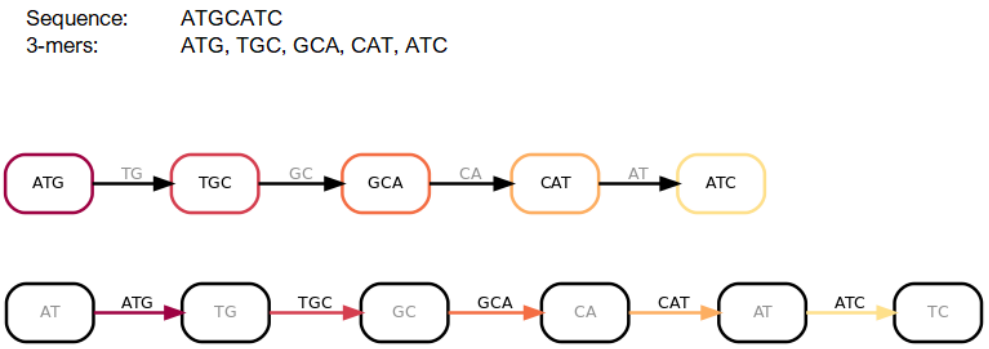

Figure 9. Example of graphs using different definitions of De Bruijn graph on the same sequence. In the top graph, k-mers correspond to vertices (definition I), whereas in the bottom graph, k-mers correspond to edges (definition II).

4.2.1. De Bruijn graphs

The process of reconstructing a set of transcripts usually starts with simplification of the De Bruijn graph and by iterative error removal steps followed by transcript reconstruction itself by traversing the graph.

De Bruijn graph is usually simplified by collapsing paths on which all nodes have only one incoming and outgoing edge. This type of De Bruijn graph is often called the compacted De Bruijn Graph. Example of this step is shown in Figure 10.

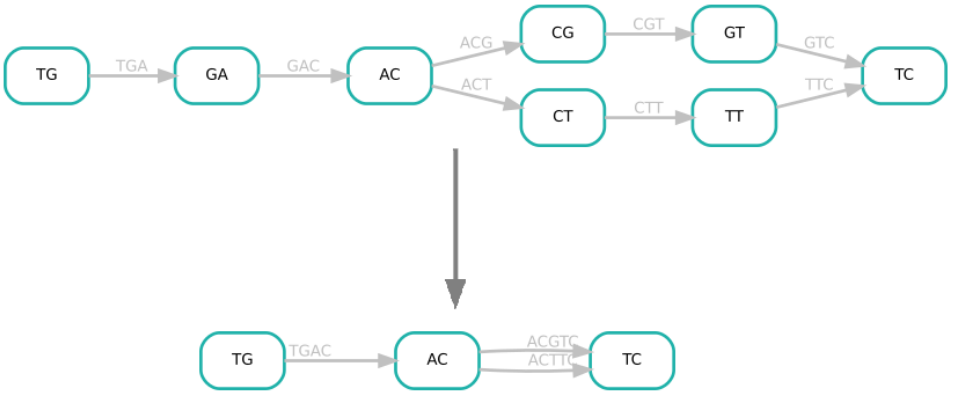

Figure 10. A compacted De Bruijn Graph constructed by collapsing paths.

Sequencing data is not perfect and contains errors. These errors can be detected by analyzing the topology of the De Bruijn graph.

Genome assemblers, on top of which many transcriptome assemblers have been built, also use the coverage information. Meaning that they use a premise saying that read sampling from the genome should be approximately uniform. Therefore, when encountering alternative edges or paths with lower coverage, the ones with lower coverage are usually discarded as erroneous. This assumption does not hold for transcriptomes where the abundance of an erroneous k-mer from a highly expressed transcript can be higher than the abundance of an error-free k-mer from a lowly expressed transcript (Peng *et al.*, 2013). Thus, coverage information has to be used carefully

Common structures indicating errors in the assembly are usually called tips, bulges and chimeric connections (See Figure 11 and Figure 12) (Zerbino and Birney, 2008; Bushmanova *et al.*, 2019).

Tips are edges starting or ending at a vertex without other adjacent edges. They can originate from errors near the ends of transcripts or from alternative isoforms. When there are two tips which do not significantly differ, the one with lower coverage might be trimmed. Also short tips with low coverage are often trimmed.

Bulges are structures where there are two alternative paths having the same starting and ending vertex. The sources of bulges are sequencing errors, allele differences, repeats and alternative splicing. When sequencing error causes the bulge, the alternative paths usually represent sequences which are very similar, are both short but have different coverage from each other. The path with a lower coverage can be removed. If the source of a bulge is allele difference, then the alternative paths usually have similar sequences, lengths and coverage. Both variants should be kept. Paths in the bulge resulting from alternative splicing usually have different lengths (due to inclusion/exclusion of an exon). Both variants should be kept.

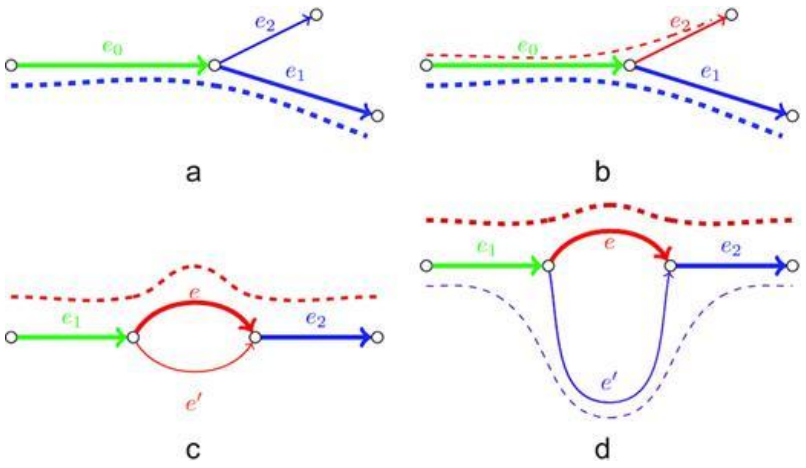

Figure 11. Detecting errors in topology. Higher coverage is represented by a thicker line. (a) There are two tips (in blue) with similar sequence but different coverage (e1 vs. e2), only the tip with higher coverage (e1) is kept. (b) Two tips with different sequences (red and blue), both of them are kept. (c) Bulge resulting from a sequencing error, only the edge with higher coverage (e) is kept. (d) Bulge corresponding to two splice variants resulting from alternative splicing, the edges have different lengths and similar coverage, both are kept. Adapted from (Bushmanova et al., 2019), with permission.

Chimeric connections result from incorrect concatenation of sequences. In the case of genome De Bruijn graphs such artifacts are possible to detect using coverage information (which should be mostly even). In the case of transcriptome de Bruijn graphs, this information cannot be used. Two common chimeric structures were reported by Bushmanova et al. (see Figure 12): single-strand chimeric loops and double-stranded hairpins (Bushmanova *et al.*, 2019).

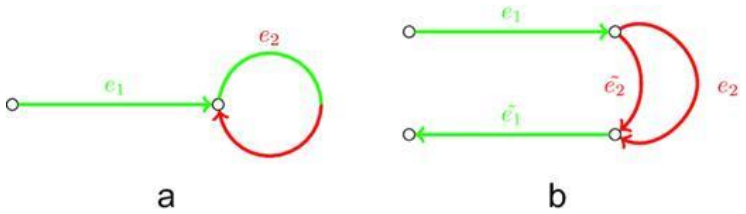

Figure 12. Chimeric connections. (a) Single-stranded chimeric loop created by connecting the end of a transcript with the transcript itself, (b) Double-stranded hairpin created by connecting a correct edge with its reverse-complement copy. Adapted from (Bushmanova et al., 2019), with permission.

Finally, after the graph has been simplified and errors corrected, the transcripts are derived from it. The process is done by constructing paths in the graph which

correspond to individual transcripts. There is not a unified strategy for path construction but usually reads (or read pairs if available) are mapped back to the graph and paths are derived based on connections of vertices highlighted by them as illustrated in Figure 13.

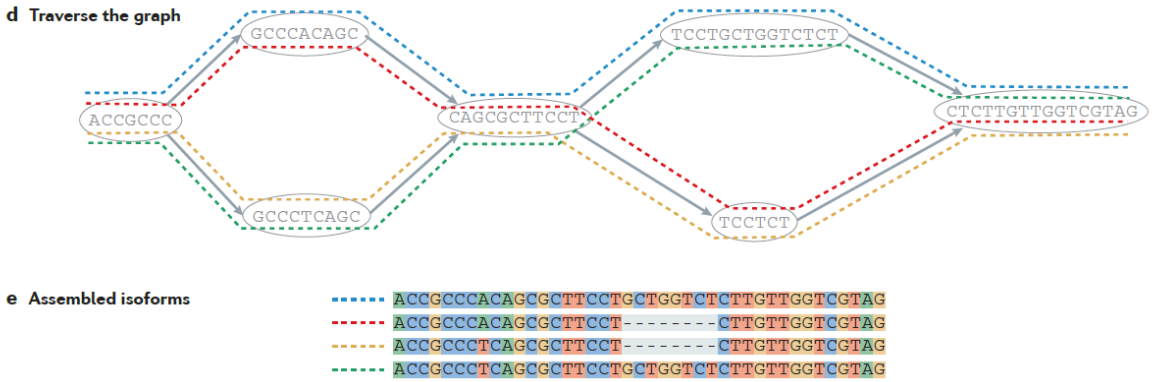

Figure 13. Mapping reads to the collapsed De Bruijn graph (colored dashed lines) and reconstructing the transcripts (assembled isoforms). Adapted from (Martin and Wang, 2011), with permission.

On a different note, an important parameter when constructing the De Bruijn graph is the size of *k*. Longer k-mers produce more contiguous assembly while decreasing transcript numbers (Robertson *et al.*, 2010). Small k-mer sizes generate more transcripts but can lead to false junctions and misassemblies (Zerbino and Birney, 2008). Furthermore, k-mer size is related to expression levels. Lowly expressed genes are better captured with shorter k-mers, whereas the opposite is true for highly expressed genes (Robertson *et al.*, 2010; Schulz *et al.*, 2012).

A way to avoid choosing a single k value and prioritize is to choose multiple k values, build a De Bruijn graph for each of them and then merge these assemblies. This approach has been shown to be successful with increasing the contiguity and ability to capture transcripts with different expression levels at the cost of introducing a higher level of redundancy, longer running time (Robertson *et al.*, 2010; Surget-Groba and Montoya-Burgos, 2010; Schulz *et al.*, 2012; Hölzer and Marz, 2019) and cumulatively inheriting all the misassemblies. Another approach is an iterative construction of the De Bruijn graph with increasing size of k. In this way, in every iteration original sequencing reads and transcripts assembled in a previous iteration are used for constructing new

De Bruijn graph with higher value of k (Peng *et al.*, 2010; Bankevich *et al.*, 2012; Bushmanova *et al.*, 2019).

4.3. *de novo* transcriptome assembly from long reads

Since long reads should represent full-length transcripts, this overcomes the challenging assembly problem. Subsequent steps after error correction include read clustering from which individual transcripts are deduced. (Gordon *et al.*, 2015; de la Rubia *et al.*, 2021)

4.4. *de novo* transcriptome assembly from hybrid sequencing

Currently there are two approaches incorporating both short and long reads into an assembly. One approach aligns long reads to contigs assembled from short reads, whereas the second approach aligns the long reads to an assembly (De Bruijn) graph constructed from the short reads.

In the first approach, short reads are independently assembled into contigs using standard short read assembly. Then the long reads are aligned to these contigs and they possibly help to extend these contigs. Long reads which are not possible to align are clustered together and their consensus sequence is deduced. (Fu *et al.*, 2018)

The latter approach uses short reads to construct an assembly graph. Long reads are then aligned to this assembly graph and these alignments are used to find paths in the graph corresponding to transcripts (Grabherr *et al.*, 2011; Prjibelski *et al.*, 2020).

4.5. Comparison of existing software tools

Many *de novo* transcriptome assemblers have been developed but there is no best one. Their performance differs on datasets from different kingdoms, also depending on sequencing depth, on the type and incidence of repeat regions, read length and other sequencing library parameters (Hölzer and Marz, 2019). A brief listing of existing tools together with a number of their citations reported by Google Scholar and year of publishing are in Table 1.

Most *de novo* transcriptome assemblers use De Bruijn graphs. They differ in strategies for error correction and for transcript path construction. They can either use a single k-mer size or multiple k-mer sizes. Therefore, if they are applied on the same input dataset, they return non-identical results

Oases (Schulz *et al.*, 2012) and Trans-ABySS (Robertson *et al.*, 2010) are *de novo* transcriptome assemblers which utilize previously developed genome assemblers (Zerbino and Birney, 2008; Simpson *et al.*, 2009) and multiple k-mer sizes. For every k, the genome assembler creates a De Bruijn graph and reconstructs the transcripts. These transcripts assembled at different k-mer sizes are then clustered and merged. Thanks to usage of multiple k-mer sizes, they can better capture transcripts which are either lowly or highly transcribed (Wang and Gribskov, 2017). On the other hand, usage of multiple k-mer sizes leads to higher redundancy and longer runtime (Hölzer and Marz, 2019).

Trinity (Grabherr *et al.*, 2011) was the first assembler created specifically for the *de novo* transcriptome assembly task. Disadvantage of Trinity is a fixed k-mer size to 25. Nevertheless, with 5 191 citations of its protocol (Haas *et al.*, 2013) on Google Scholar in August 2021, it is one of the most popular assemblers. Trinity can also perform hybrid assembly using both Illumina short reads and PacBio CCS reads.

Another assembler which uses a previously developed genome assembler (Luo *et al.*, 2012) for the De Bruijn graph construction is SOAPdenovo-Trans (Xie *et al.*, 2014). It uses error removal steps from Trinity and a graph traversal procedure for transcript reconstruction from Oases. SOAPdenovo-Trans uses only a single k-mer size and has a rather short runtime in comparison with other assemblers (Hölzer and Marz, 2019).

RnaSPAdes (Bushmanova *et al.*, 2019) is a *de novo* transcriptome assembler which uses 2 k-mer sizes and builds the De Bruijn graph iteratively. The k-mer sizes can either be provided by the user or their ideal size can be estimated from the length distribution of reads. HybridSPAdes (Prjibelski *et al.*, 2020) uses rnaSPAdes to construct the De Bruijn graph from short reads. It then aligns long reads to the graph and reconstructs the transcripts,

Another hybrid assembler is IDP-denovo (Fu *et al.*, 2018). It assembles transcripts from the short reads and possibly extends them using long reads. Unused long reads are clustered and used to create consensus transcripts. Utilization of long reads which are not covered by short read data in IDP-denovo is the main difference to other hybrid methods.

For long read technologies, tools for error correction described in chapter 3.4 usually also perform the clustering and final transcript reconstruction (Gordon *et al.*, 2015; Sahlin *et al.*, 2018; *IsoSeq*, 2020; Sahlin and Medvedev, 2021). Their performance comparison has yet to be evaluated.

RnaQUAST (Bushmanova *et al.*, 2016) is a tool which can evaluate transcriptome assemblies by using reference genomes. It can be used to compare newly developed assemblers to already existing ones on the same dataset. Comprehensive comparison of assemblers on 9 different datasets using rnaQUAST was done by (Hölzer and Marz, 2019).

Assembler	**Number of citations**	**Year of publishing**
Trans-ABySS	971	2010
Trinity	1206[2]	2011
Oases	1461	2012
SOAPdenovo-Trans	756	2014
rnaSPAdes	134	2019
hybrid SPAdes	284	2016
IDP-denovo	24	2018

Table 1. Number of citations on Google Scholar as of August 2021 and the year of publishing of individual assemblers.

[2] Number of citations for the main paper (Grabherr *et al.*, 2011), Trinity protocol (Haas *et al.*, 2013) has 5 191 citations

4.6. Combining multiple assemblies

There is no single best assembler and although most of them are built on similar principles, they produce different sets of transcripts. To combine strengths of individual assemblers, it is possible to merge their assemblies (Hölzer and Marz, 2019). Combining transcripts produced by different assemblers can increase overall completeness (Smith-Unna *et al.*, 2016). For this purpose the Oyster River Protocol can be used (MacManes, 2018).

When correctness is the priority, it may be a good approach to keep the intersection of transcripts produced by different assemblers. Shared set of transcripts is likely to be correctly assembled by multiple assemblers (Voshall *et al.*, 2021).

5. Transcriptome evaluation and quality assessment

It is difficult to determine if the resulting assembly is good or bad. There is no single metric which would reflect the quality. However, tools such as TransRate (Smith-Unna *et al.*, 2016) and RSEM-EVAL (Li *et al.*, 2014) combining several metrics and using support from the reads, try to assess the assembly quality. BUSCO is a biological based metric assessing the completeness of gene content (Simão *et al.*, 2015).

TransRate tries to evaluate the quality of assembly with respect to artifacts such as family collapse, chimerism, unsupported insertion, incompleteness, fragmentation, local misassembly and redundancy as illustrated in Figure 14. It provides a quality score to each transcript and an overall assembly score (Smith-Unna *et al.*, 2016).

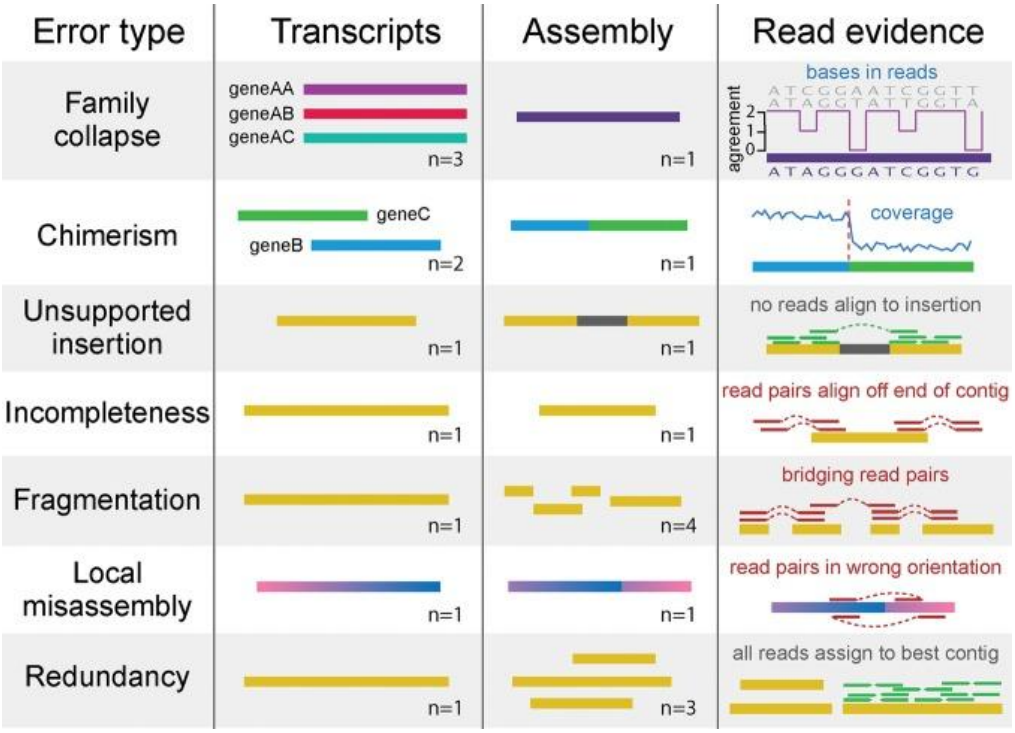

Figure 14. Error types and their detection by using reads: family collapse (transcripts from distinct but highly similar genes get assembled into a single transcript), chimerism (incorrect merge of 2 transcripts which was detected by different coverage of reads in corresponding parts of the chimeric transcript), unsupported insertion (no support of reads of the insertion), incompleteness (missing part of the transcript), fragmentation of a transcript, local misassembly (inverted part of a transcript) and redundancy (more assembled transcripts corresponding to one original). Adapted from (Smith-Unna et al., 2016), with permission.

RSEM-EVAL evaluates how well the assembly is explained by the RNA-Seq reads and evaluates the assembly compactness. It produces an overall score for the assembly as well as scores for individual contigs which reflects how well the contig is supported by the reads. Therefore, reads with low scores may be filtered (Li *et al.*, 2014).

BUSCO (Benchmarking Universal Single-Copy orthologs) can be used to assess completeness of a transcriptome in terms of gene content. It is a set containing single-copy orthologs found in more than 90% of species in sets of phylogenetic clades. The analysis of the transcriptome then provides numbers of complete, duplicated, fragmented, and missed genes. For species highly derived from the assessment clade, analysis may result in more missing genes only due to longer evolutionary history and not due to incomplete assembly (Simão *et al.*, 2015).

These tools can be used to tune assemblers' parameters or to compare assemblies obtained with various assemblers from the same data.

6. Downstream analyses

One goal of transcriptomics is to catalogue all the transcripts in a species. This information can then be further used to quantify the transcripts and find those which are differentially expressed amongst different tissues, conditions or developmental stages. Differentially expressed transcripts are often biologically relevant. Functional annotation then enables us to gain more insight into the functions of transcripts.

6.1. Expression quantification

RNA-Seq became a widely used technique for measuring gene expression which also is the first step in detection of differential expression among samples (Patro, Mount and Kingsford, 2014).

The basic idea of transcript quantification with RNA-Seq is to align sequencing reads to the transcriptome and count the number of reads aligned to each transcript. The higher the number of reads aligned to a transcript (relative to its length), the higher the abundance of the transcript.

One of the challenges of quantification is the presence of reads which map to multiple transcripts (these reads are often referred to as “multireads”). Simple approach addressing presence of multireads is discarding the multireads and performing the quantification solely on the rest of the reads. This approach wastes data and can induce bias (Li and Dewey, 2011). In the case of transcripts derived from paralogous genes, many reads would not align uniquely and thus would be discarded for both of these transcripts (Mortazavi *et al.*, 2008). Therefore, other methods addressing this were developed.

These quantification methods can be divided into two classes: alignment-based and alignment-free. Alignment-based methods are more computationally expensive (due to the alignment). Example of such a method is RSEM (Li and Dewey, 2011). Faster and less computationally expensive are alignment-free methods such as Sailfish (Patro, Mount and Kingsford, 2014), kallisto (Bray *et al.*, 2016) and Salmon (Patro *et al.*, 2017). Alignment-free methods speed up the quantification by avoiding the alignment step;

they instead use variations of exact matching, which is fast. Sailfish uses exact matching of read k-mers to transcripts, kallisto uses exact matching of read k-mers to De Bruijn graph constructed from the transcripts and Salmon tries to find a chain of super maximal exact matches. All aforementioned methods use the Expectation-Maximization (EM) algorithm[3] to estimate the abundances.

In case of long read sequencing, every long read should correspond to a single transcript. Quantification is then just straight forward counting of number of reads (Wyman *et al.*, 2019). However, the lower sequencing depth in long read sequencing makes the quantification less accurate (Dong *et al.*, 2021).

6.2. Differential expression analysis

Many transcriptomic studies aim to find a set of genes which have different expression levels between samples. Different tissues, conditions or development stages can be compared. Genes which are differentially expressed are often biologically relevant.

Analysis of differential expression typically proceeds in 3 steps: read count normalization, model parameters estimation and testing for differential expression (Rapaport *et al.*, 2013).

The goal of count normalization is to adjust for differences in library sizes and to adjust for differences in library compositions so that samples can be compared.

Looking at sequencing as a random sampling of reads from a fixed pool of genes naturally leads to a model of Poisson distribution. In Poisson distribution, mean and variance are the same. However, in practise the variance of gene expression across samples is greater than mean expression values, a phenomenon called overdispersion. Therefore, negative binomial distribution is usually used instead. Negative binomial distribution takes overdispersion into account because it can adjust variance independently from the mean (Rapaport *et al.*, 2013).

[3] For the first publication explaining the EM algorithm see (Dempster, Laird and Rubin, 1977).

Some of the most popular and accurate (Sahraeian *et al.*, 2017) differential expression analysis tools are DESeq (Anders and Huber, 2010) (or newer version DESeq2 (Love, Huber and Anders, 2014)) and edgeR (Robinson, McCarthy and Smyth, 2010).

Both edgeR and DESeq use negative binomial distribution and are based on a hypothesis that most genes are not differentially expressed. They differ in how counts are normalized, how the overdispersion parameter is calculated and in hypothesis testing (although both use variation of Fisher exact test) (Rapaport *et al.*, 2013).

As mentioned in chapter 2.3, the number of replicates should be prioritized over sequencing depth when the main goal is to study differential expression (Rapaport *et al.*, 2013).

Results of differential expression analysis are often visualized as volcano plots. The Y axis on the volcano plot measures negative logarithm of p value, the x axis measures logarithm of fold change difference. The most significantly differentially expressed transcripts are in the plot on the top left or top right positions. Figure 15 shows an example of a volcano plot.

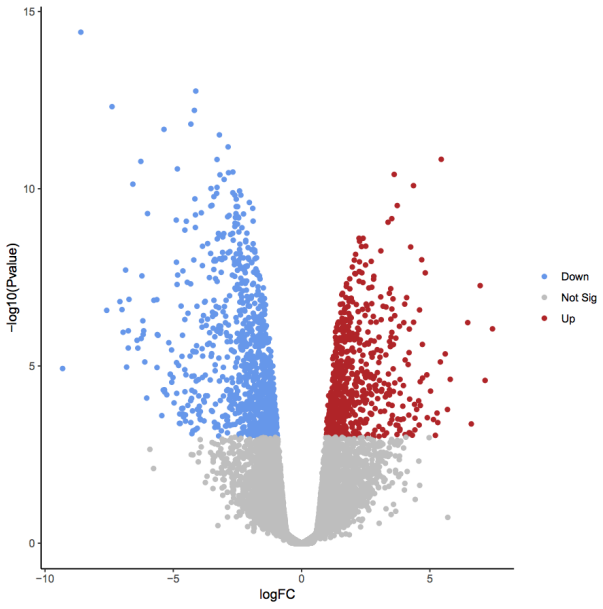

Figure 15. Volcano plot showing differential expression analysis results. Most significant transcripts are on the top left and top right where logFC and -logPval are the most away from zero. (Galaxy Training: Visualization of RNA-Seq results with Volcano Plot, no date)

6.3. Functional annotation

Collecting information about the individual transcripts helps to gain more insight into the overall transcriptome. Information can be acquired thanks to public databases with annotated data using similarity search or by using predictions. Functional annotation can be used to mine interesting transcripts.

Most assembled transcripts are expected to encode proteins (since mRNA capture or rRNA depletion was performed before sequencing). TransDecoder (Haas, 2018) is a tool which predicts proteins encoded by transcripts based on nucleotide composition, open reading frame length and Pfam domain content (Haas *et al.*, 2013).

Both transcripts and predicted proteins can be used to query a protein database such as the manually annotated, non-redundant SwissProt database with BLAST (Altschul *et al.*, 1990). Transcripts can also be used to query the Rfam database of non-coding RNA families and proteins to query the Pfam database of protein families.

For protein sequences there exist tools which predict the presence of targeting peptides (Jose Juan Almagro Armenteros *et al.*, 2019; José Juan Almagro Armenteros *et al.*, 2019), subcellular localization (Almagro Armenteros *et al.*, 2017) or topology of transmembrane proteins (Krogh *et al.*, 2001; Reeb *et al.*, 2015). These tools can be further used to annotate the protein sequences.

7. Practical section

7.1. The *transXpress* pipeline

transXpress is a *de novo* transcriptomics pipeline which is open source and freely available for use. It is a collaborative project of several labs, including my supervisor's lab.

By providing only sequencing reads and a few additional parameters, *transXpress* goes through all steps of the *de novo* transcriptomics experiment workflow described in chapter 1.3. Quality control of input short reads is done using FastQC (Andrews, 2010); the outputs for each dataset (sample) are merged into a single report with MultiQC (Ewels *et al.*, 2016). Reads are trimmed using Trimmomatic (Bolger, Lohse and Usadel, 2014) and assembled into transcripts with either rnaSPAdes (Bushmanova *et al.*, 2019) or Trinity (Grabherr *et al.*, 2011). Transcripts are quantified with kallisto (Bray *et al.*, 2016) and differential expression is analyzed with edgeR (Robinson, McCarthy and Smyth, 2010). TransDecoder (Haas, 2018) predicts protein products from the assembled transcriptome. For protein sequences, presence of a targeting peptide and subcellular localization are predicted using TargetP (Jose Juan Almagro Armenteros *et al.*, 2019) and Deeploc (Almagro Armenteros *et al.*, 2017), respectively. TMHMM (Krogh *et al.*, 2001; Søndergaard, 2019) is used to predict the secondary structure of membrane proteins. Transcripts and protein sequences are annotated with the best BLAST (Altschul *et al.*, 1990) hit found in the SwissProt database. Graphical visualization of the pipeline is shown in Figure 16.

transXpress is written in Snakemake (Köster and Rahmann, 2012) and uses the Anaconda (Anaconda Inc., 2020) package management system. Snakemake workflow engine divides the workflow into a set of rules whose mutual dependencies are inferred from the defined format of input and output files of each rule. These dependencies form a directed acyclic graph of the rules. Disjoint paths in the graph can be executed independently in parallel; Snakemake supports execution on a computational cluster. Snakemake only executes a rule if its output files are not present or if the modification

time of the input files is newer, it avoids duplicate work (Köster and Rahmann, 2012). The *transXpress* pipeline takes advantage of Snakemake's features and is highly parallelized. Because *transXpress* uses many software tools, the Anaconda package management system represents a very useful method for their handling. It ensures that all tools are compatible with each other.

transXpress is executed as a command line program and works on Linux based systems. It has been tested on LSF, PBS and SLURM high-performance computational clusters and cloud execution such as Amazon Web Services is also possible.

transXpress is freely available from: https://github.com/transXpress/transXpress-snakemake under the GNU General Public License v3.0 (GPLv3).

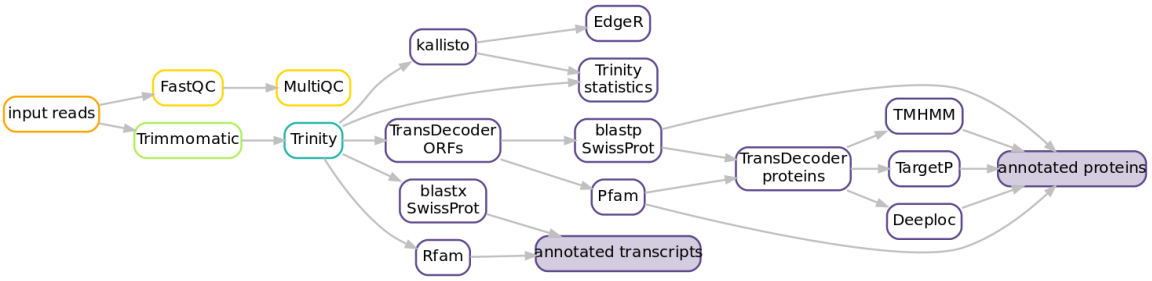

Figure 16. Visualization of steps involved in transXpress pipeline. Quality of input reads (orange) is inspected with FastQC (Andrews, 2010) and MultiQC (Ewels et al., 2016) (yellow), reads are preprocessed with Trimmomatic (Bolger, Lohse and Usadel, 2014) (green) and assembled with either Trinity, as in this case, (Grabherr et al., 2011) or rnaSPAdes (Bushmanova et al., 2019) (blue). On the assembled transcriptome, downstream analyses are performed (purple). These include quantification with kallisto (Bray et al., 2016), differential expression analysis with EdgeR (Robinson, McCarthy and Smyth, 2010) and functional annotation. Functional annotation is performed using similarity search with BLAST (Altschul et al., 1990) against SwissProt database, cmmscan (Nawrocki and Eddy, 2013) against Rfam and hmmscan (HMMER, 2020) against Pfam. Predictions are made using TMHMM (Krogh et al., 2001; Søndergaard, 2019), TargetP (Jose Juan Almagro Armenteros et al., 2019) and Deeploc (Almagro Armenteros et al., 2017) on sequences predicted with TransDecoder (Haas, 2018). Annotated transcriptome and annotated proteins (graph nodes with purple background) are the main output files (in FASTA format with heavily decorated description lines containing e.g. transcript quantification, predicted features, etc., shown later in Figure 24).

7.2. Functions implemented in this thesis

I contributed to the *transXpress* project and implemented workflow steps for short-read quality control, functional annotation and differential expression analysis. I have also optimized package installation.[4]

For input data quality control I selected the FastQC tool (Andrews, 2010) which is run on every input file with sequencing data. MultiQC then merges FastQC reports into a single report (Ewels *et al.*, 2016).

To gain more insight into what the transcriptome consists of, I implemented steps which predict the presence of targeting peptide and secondary structure of transmembrane proteins. Targeting peptide presence prediction is done through a tool called TargetP (Jose Juan Almagro Armenteros *et al.*, 2019). This tool, which uses deep learning, can predict whether a protein sequence contains a signal peptide, mitochondrial transit peptide, chloroplast transit peptide or thylakoid luminal transit peptide. To predict if a protein sequence is likely to be a transmembrane protein and if so, its secondary structure, I used the Python implementation (Søndergaard, 2019) of TMHMM (Krogh *et al.*, 2001).

Differential expression analysis is done through a Trinity (Grabherr *et al.*, 2011; Haas *et al.*, 2013) script which runs edgeR (Robinson, McCarthy and Smyth, 2010). By comparing expression levels of transcripts in individual samples (different tissues, conditions, developmental stages, etc.), it enables us to identify those whose expression varies the most. This highlights their possible biological importance, which has to be evaluated by the user in a case-by-case manner.

7.3. Case study: *Piper longum* transcriptome

Piper longum (also known as long pepper) is a non-model plant whose fruits are used as a seasoning. *Piper* plants have also been used in traditional medicine from ancient times. *Piper longum* produces biochemically interesting compounds such as terpenes, alkaloids

[4] For GitHub commits see:
github.com/transXpress/transXpress-snakemake/commits?author=CalounovaT

and flavonoids (Parmar *et al.*, 1997). Many of these compounds are very difficult to synthesize in the lab and their synthetic mass production for potential drugs would be untenable. Hence it is desirable to discover pathways, and their "protein players" — enzymes, by which plants produce these compounds. Transcriptome as an intermediate between genome and proteome represents a great source of information from which genes for these enzymes could be identified. When genes involved in the metabolic pathway are discovered and cloned, they can be transferred to host organisms which can then use this "know how" to produce such compounds of interest. Furthermore, the pathway may be possible to optimize so it gives higher yields of the compound (Facchini *et al.*, 2012).

To uncover the biochemical and medicinal potential of *Piper longum*, it is needed to gain more insight into its genetic basis. Genome information is not yet available for *Piper longum* but transcriptomic data was recently published (Dantu, Prasad and Ranjan, 2021). I used this data to perform *de novo* transcriptome assembly and functional annotation. I also used this data to compare gene expression amongst different tissues. My results can be used further to identify biologically relevant enzymes by mining this transcriptome.

Sequencing data was acquired from NCBI Sequence Read Archive (SRA) and contained Illumina stranded, paired-end 2x150 bp reads from Piper longum leaf (SRR10362954), spike — the fruit (SRR10362953) and root (SRR10583928) samples. This data contained 16 901 456, 22 900 035 and 27 496 748 reads, for leaf, spike and root respectively.

The *transXpress* pipeline was run on the IOCB computational cluster with Trinity (Grabherr *et al.*, 2011) as the assembler of choice.

Predicted sequencing quality of the input data is rather high as illustrated by Figure 17 which shows quality plots in the MultiQC report (Ewels *et al.*, 2016).

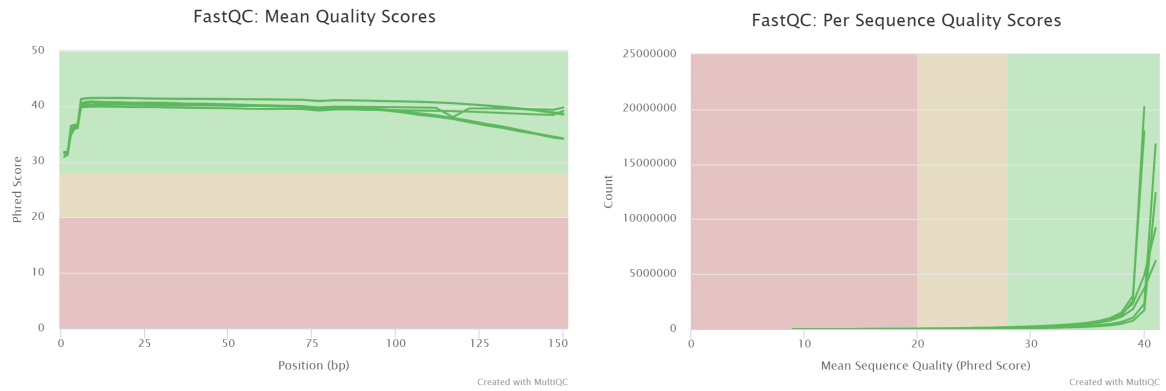

Figure 17. MultiQC report showing plot with mean quality scores (on the left) and per sequence quality scores (on the right). On the left, X axis = bp position, Y axis = predicted Phred score. Mean quality for all reads lies in an interval of Phred score over 28 (green). There is a slight quality "jump" at the first few bases which may originate from Illumina machine calibration. Towards the end, quality slightly decreases again which is very common (for explanation see chapter 3.1). On the right, X axis = mean predicted Phred score, Y axis = count. Mean quality of most sequences is well above predicted Phred quality over 28 (green).

The resulting transcriptome consists of 279 145 transcripts corresponding to 133 702 genes. The average transcript length is 947.7bp, the median transcript length is 590. More information about the length distribution of transcripts can be seen in Figure 18.

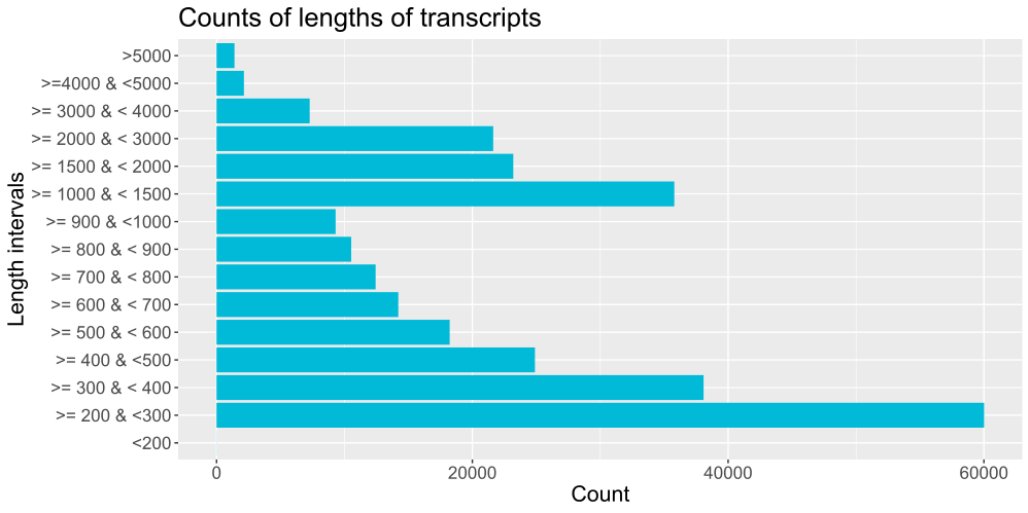

Figure 18. Counts of transcripts with lengths in given intervals (bins). There are no transcripts shorter than 200bp because 200bp was set as a minimum length in Trinity for reporting transcripts. Most transcripts are relatively short. Reconstructing full-length transcripts from short reads is still challenging in de novo transcriptome assembly.

Based on the 279 145 assembled transcripts, putative protein sequences were predicted in *transXpress* using TransDecoder (Haas, 2018). Total number of protein sequences in *Piper longum* transcriptome is 142 310. Average length of protein sequences is 284 amino acids, the median length is 206 amino acids. More information about the length distribution can be seen in Figure 19. Most of the proteins (54.6%) were predicted by Transdecoder to be full-length, followed by 5` partial (24.6%), internal (10.5%) and 3` partial (10.3%).

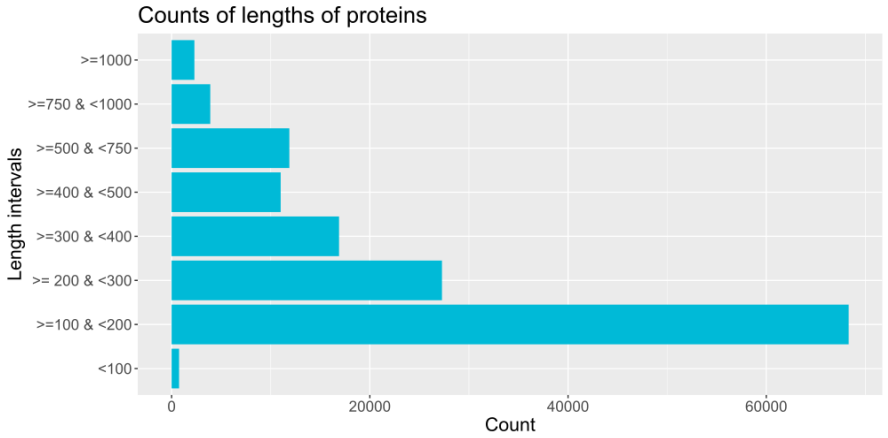

Figure 19. Counts of proteins with lengths in given intervals (bins). Although most proteins are reported to be complete by TransDecoder, they are rather short.

As illustrated in Figure 20, most (86.1%) of the protein sequences were predicted not to contain targeting peptides. Out of the protein sequences predicted to have targeting peptide, the most common was signal peptide followed by a chloroplast transit peptide.

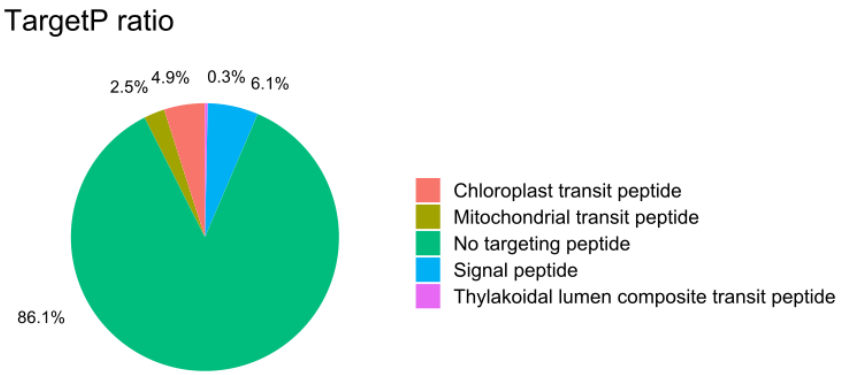

Figure 20. Ratio of proteins with or without targeting peptides as predicted with TargetP.

27 484 protein sequences (representing 19 % of all protein sequences) were predicted to be membrane proteins.

Out of 10 possible subcellular locations, most protein sequences were predicted to be localized in the nucleus (25.13%), cytoplasm (20.81%) and mitochondrion (16.99%). More information can be seen in Figure 21.

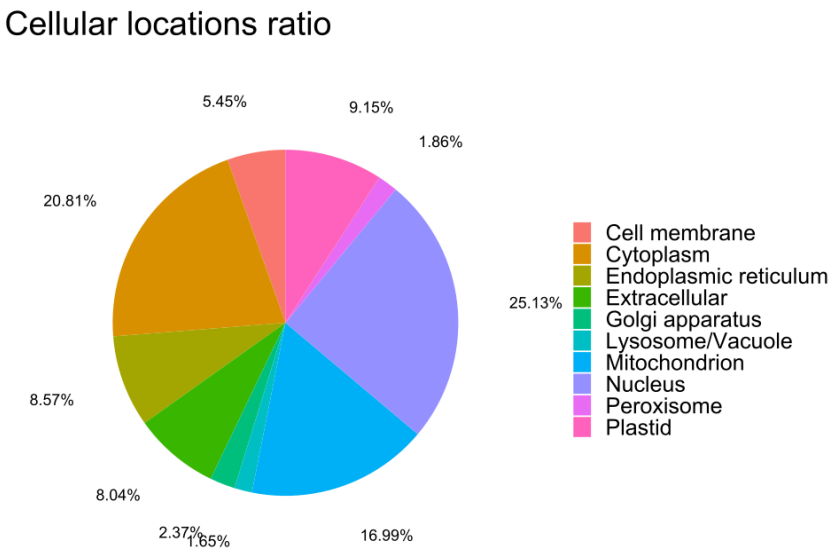

Figure 21. Ratio of predicted subcellular localizations by Deeploc.

TransXpress also ran a BLAST (Altschul *et al.*, 1990) search against the SwissProt protein database. The e-value threshold for this search was 1e-6. Out of 279 145 transcript sequences, 132 640 (48%) had at least one hit with an e-value higher than the specified threshold. Out of 142 310 protein sequences, 103 512 (73%) had at least one hit with an e-value higher than the specified threshold.

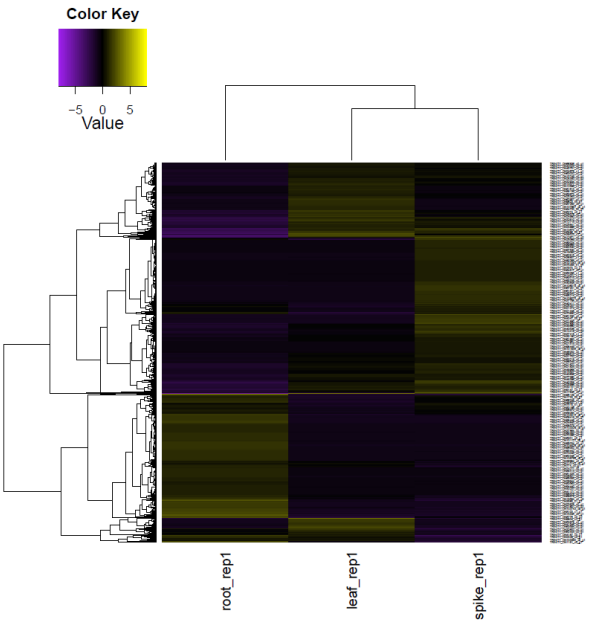

Figure 22. Heatmap showing differential expression between root, leaf and tissue samples.

Figure 22 shows a heatmap demonstrating differential expression between the 3 tissue samples which was created based on analysis performed using edgeR (Robinson et al. 2010). Further investigation of the top differentially expressed transcripts identified by this analysis could lead to discovery of biologically relevant transcripts in *Piper longum*.

The *Piper longum* transcriptome assembly was then subjected to quality evaluation using BUSCO (Simão *et al.*, 2015) and RSEM-EVAL (Li *et al.*, 2014).

BUSCO (Simão *et al.*, 2015) v5.2.2 was run on the transcriptome using the lineage dataset embryophyta_odb10 (eukaryota, 2020-09-10) as the closest available lineage for *Piper longum*. The result of this assessment is the following: C:95.3%[S:11.9%,D:83.4%],F:2.7%,M:2.0%,n:1614 (counts are shown in Figure 23). Meaning that 95.3% of orthologs were found and were complete. Out of them only 11.9% were single-copy, whereas 83.4% were duplicated. This high proportion of duplication is likely due to the presence of many isoforms in the transcriptome. 2.7% of BUSCO orthologs were fragmented and only 2.0% were missing. Overall, this BUSCO assessment gave information that the transcriptome assembly has a high quality in terms of gene completeness.

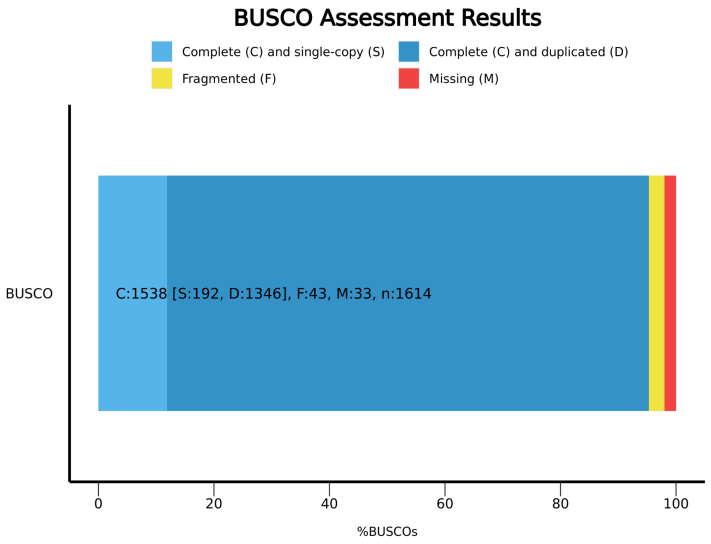

Figure 23. BUSCO results of Piper longum transcriptome assessment.

Another quality assessment was done using RSEM-EVAL (Li *et al.*, 2014). Final assembly score is -7 606 636 138.53.

TransRate (Smith-Unna *et al.*, 2016) quality assessment was not performed due to technical problems in TransRate.

The *Piper longum* transcriptome can be further mined to find interesting transcripts and proteins. User-friendly way to do so is to use SequenceServer (Priyam *et al.*, 2019) which enables performing BLAST (Altschul *et al.*, 1990) searches against a custom database made from the annotated transcriptome.

To demonstrate this, I selected a terpene synthase enzyme (UniProt accession A0A2R4QKX7) from a related plant species *Piper nigrum* (black pepper). This enzyme has been shown to synthesize sesquiterpenes α-copaene and β-cadinene. Sesquiterpenes contribute to the flavor of black pepper (Jin *et al.*, 2018).

I used SequenceServer version 1.0.14 to query this enzyme against the *Piper longum* transcriptome. Using e-value cutoff 1.0e-5, I obtained 134 hits in the transcriptome.

For every hit, SequenceServer shows its alignment to the query and also the functional annotation which was created with transXpress — quantification in different samples, best hit in SwissProt, identified Pfam domains, topology prediction for transmembrane

proteins, subcellular localization and presence of targeting peptide prediction. Example is shown in Figure 24.

Figure 24. Example of a BLAST hit displayed in SequenceServer. In the description line there are the following annotated features: predicted completeness, length, TPM (transcripts per million) values for different samples, annotation of best blastp hit, Pfam domain hit and its annotation, tmhmm topology prediction (in this case the protein is not likely to be transmembrane), Deeploc subcellular localization prediction and prediction of presence of targeting peptide using TargetP.

Conclusions and future prospects

Research efforts over the 20th century have mainly focused on model systems, with tools and genomic resources being specifically developed to support their investigation. Second generation sequencing and *de novo* transcriptomics paved the way for the investigation of non-model organisms through genomic approaches in an ever more cost-effective way.

In this work I described the workflow of *de novo* transcriptomics experiments. In particular, I focused on well-established methods, as well as promising approaches (i.e., third generation sequencing) under active development. Throughout the thesis, I highlighted critical steps and pitfalls which might negatively affect the final results and provided recommendations to ensure high-quality of the transcriptome assembly.

In the “Practical section”, I presented the application of *transXpress*, a highly parallelized pipeline for *de novo* assembly and functional annotation of RNA-Seq data. *TransXpress* allows users to carry out a full *de novo* transcriptomics workflow by only providing sequencing reads as an input. I contributed to its development and I tested it in real-world scenarios. Specifically, I employed the pipeline to mine the recently-published transcriptomic data of *Piper longum* — a non-model plant of great medicinal interest. I showed how differential expression analysis and similarity searching can be performed on its transcriptome to find transcripts potentially encoding biologically-relevant enzymes. This demonstrated how *transXpress* can be effectively implemented to study non-model organisms via *de novo* transcriptomics.

In-depth exploration of the *Piper longum* transcriptome, ultimately leading to identification of candidate genes for enzymes involved in biosynthetic pathways of its secondary metabolites, was beyond the scope of this thesis. It represents the direction of future work. Non-model organisms offer great research opportunities due to their unexplored complexity. With the ever-rising progress in technologies, their complexity will eventually get more and more uncovered. It will be only up to our ideas on how to use this gained knowledge. Understanding and utilizing the chemical "know how" used by plants could lead to breakthroughs in drug research and is an exciting field to study.

References

Almagro Armenteros, J. J. *et al.* (2017) 'DeepLoc: prediction of protein subcellular localization using deep learning', *Bioinformatics* , 33(21), pp. 3387–3395.

Almagro Armenteros, J. J. *et al.* (2019) 'Detecting sequence signals in targeting peptides using deep learning', *Life science alliance*, 2(5). doi: 10.26508/lsa.201900429.

Almagro Armenteros, J. J. *et al.* (2019) 'SignalP 5.0 improves signal peptide predictions using deep neural networks', *Nature biotechnology*, 37(4), pp. 420–423.

Altschul, S. F. *et al.* (1990) 'Basic local alignment search tool', *Journal of molecular biology*, 215(3), pp. 403–410.

Amarasinghe, S. L. *et al.* (2020) 'Opportunities and challenges in long-read sequencing data analysis', *Genome biology*, 21(1), p. 30.

Anaconda Inc. (2020) *Anaconda*, *Software Distribution*. Available at: https://anaconda.com/ (Accessed: 6 August 2021).

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

Andrews, S. (2010) 'FastQC: a quality control tool for high throughput sequence data'. Available at: https://www.bioinformatics.babraham.ac.uk/projects/fastqc/.

Au, K. F. *et al.* (2012) 'Improving PacBio long read accuracy by short read alignment', *PloS one*, 7(10), p. e46679.

Bainbridge, M. N. *et al.* (2006) 'Analysis of the prostate cancer cell line LNCaP transcriptome using a sequencing-by-synthesis approach', *BMC genomics*, 7, p. 246.

Bankevich, A. *et al.* (2012) 'SPAdes: a new genome assembly algorithm and its applications to single-cell sequencing', *Journal of computational biology: a journal of computational molecular cell biology*, 19(5), pp. 455–477.

Bolger, A. M., Lohse, M. and Usadel, B. (2014) 'Trimmomatic: a flexible trimmer for Illumina sequence data', *Bioinformatics* , 30(15), pp. 2114–2120.

Bräutigam, A. and Gowik, U. (2010) 'What can next generation sequencing do for you? Next generation sequencing as a valuable tool in plant research', *Plant biology* , 12(6), pp. 831–841.

Bray, N. L. *et al.* (2016) 'Near-optimal probabilistic RNA-seq quantification', *Nature biotechnology*, 34(5), pp. 525–527.

Bushmanova, E. *et al.* (2016) 'rnaQUAST: a quality assessment tool for de novo transcriptome assemblies', *Bioinformatics* , 32(14), pp. 2210–2212.

Bushmanova, E. *et al.* (2019) 'rnaSPAdes: a de novo transcriptome assembler and its

application to RNA-Seq data', *GigaScience*, 8(9). doi: 10.1093/gigascience/giz100.

Chin, C.-S. *et al.* (2013) 'Nonhybrid, finished microbial genome assemblies from long-read SMRT sequencing data', *Nature methods*, 10(6), pp. 563–569.

Cock, P. J. A. *et al.* (2010) 'The Sanger FASTQ file format for sequences with quality scores, and the Solexa/Illumina FASTQ variants', *Nucleic acids research*, 38(6), pp. 1767–1771.

Compeau, P. E. C., Pevzner, P. A. and Tesler, G. (2011) 'How to apply de Bruijn graphs to genome assembly', *Nature biotechnology*, 29(11), pp. 987–991.

Conesa, A. *et al.* (2016) 'A survey of best practices for RNA-seq data analysis', *Genome biology*, 17, p. 13.

Crick, F. H. (1958) 'On protein synthesis', *Symposia of the Society for Experimental Biology*, 12, pp. 138–163.

Dantu, P. K., Prasad, M. and Ranjan, R. (2021) 'Elucidating biosynthetic pathway of piperine using comparative transcriptome analysis of leaves, root and spike in Piper longum L', *bioRxiv*. doi: 10.1101/2021.01.03.425108.

Dempster, A. P., Laird, N. M. and Rubin, D. B. (1977) 'Maximum Likelihood from Incomplete Data via the EM Algorithm', *Journal of the Royal Statistical Society. Series B, Statistical methodology*, 39(1), pp. 1–38.

Dong, X. *et al.* (2021) 'The long and the short of it: unlocking nanopore long-read RNA sequencing data with short-read differential expression analysis tools', *NAR genomics and bioinformatics*, 3(2), p. lqab028.

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

Ekblom, R. *et al.* (2012) 'Comparison between Normalised and Unnormalised 454-Sequencing Libraries for Small-Scale RNA-Seq Studies', *Comparative and functional genomics*, 2012, p. 281693.

Ekblom, R. and Galindo, J. (2011) 'Applications of next generation sequencing in molecular ecology of non-model organisms', *Heredity*, 107(1), pp. 1–15.

Ewels, P. *et al.* (2016) 'MultiQC: summarize analysis results for multiple tools and samples in a single report', *Bioinformatics* , 32(19), pp. 3047–3048.

Ewing, B. *et al.* (1998) 'Base-calling of automated sequencer traces using phred. I. Accuracy assessment', *Genome research*, 8(3), pp. 175–185.

Ewing, B. and Green, P. (1998) 'Base-calling of automated sequencer traces using phred. II. Error probabilities', *Genome research*, 8(3), pp. 186–194.

Facchini, P. J. *et al.* (2012) 'Synthetic biosystems for the production of high-value plant metabolites', *Trends in biotechnology*, 30(3), pp. 127–131.

Francis, W. R. *et al.* (2013) 'A comparison across non-model animals suggests an optimal sequencing depth for de novo transcriptome assembly', *BMC genomics*, 14, p. 167.

Fu, S. *et al.* (2018) 'IDP-denovo: de novo transcriptome assembly and isoform annotation by hybrid sequencing', *Bioinformatics* , 34(13), pp. 2168–2176.

Garalde, D. R. *et al.* (2018) 'Highly parallel direct RNA sequencing on an array of nanopores', *Nature methods*, 15(3), pp. 201–206.

Gordon, S. P. *et al.* (2015) 'Widespread Polycistronic Transcripts in Fungi Revealed by Single-Molecule mRNA Sequencing', *PloS one*, 10(7), p. e0132628.

Grabherr, M. G. *et al.* (2011) 'Full-length transcriptome assembly from RNA-Seq data without a reference genome', *Nature biotechnology*, 29(7), pp. 644–652.

Haas, B. J. *et al.* (2013) 'De novo transcript sequence reconstruction from RNA-seq using the Trinity platform for reference generation and analysis', *Nature protocols*, 8(8), pp. 1494–1512.

Haas, B. J. (2018) *TransDecoder*. Github. Available at: https://github.com/TransDecoder/TransDecoder (Accessed: 4 August 2021).

Hallberg, R. L. and Bruns, P. J. (1976) 'Ribosome biosynthesis in Tetrahymena pyriformis. Regulation in response to nutritional changes', *The Journal of cell biology*, 71(2), pp. 383–394.

Hansen, K. D., Brenner, S. E. and Dudoit, S. (2010) 'Biases in Illumina transcriptome sequencing caused by random hexamer priming', *Nucleic acids research*, 38(12), p. e131.

Hara, Y. *et al.* (2015) 'Optimizing and benchmarking de novo transcriptome sequencing: from library preparation to assembly evaluation', *BMC genomics*, 16, p. 977.

HMMER (2020). Available at: http://hmmer.org/.

Hoang, N. V. *et al.* (2019) 'The Impact of cDNA Normalization on Long-Read Sequencing of a Complex Transcriptome', *Frontiers in genetics*, 10, p. 654.

Hölzer, M. and Marz, M. (2019) 'De novo transcriptome assembly: A comprehensive cross-species comparison of short-read RNA-Seq assemblers', *GigaScience*, 8(5). doi: 10.1093/gigascience/giz039.

Huang, Y., Gilna, P. and Li, W. (2009) 'Identification of ribosomal RNA genes in metagenomic fragments', *Bioinformatics* , 25(10), pp. 1338–1340.

Hutchins, E. D. *et al.* (2014) 'Transcriptomic analysis of tail regeneration in the lizard Anolis carolinensis reveals activation of conserved vertebrate developmental and repair mechanisms', *PloS one*, 9(8), p. e105004.

IsoSeq (2020). Github. Available at: https://github.com/PacificBiosciences/IsoSeq (Accessed: 7 August 2021).

Jin, Z. *et al.* (2018) 'Molecular cloning and functional characterization of three terpene synthases from unripe fruit of black pepper (Piper nigrum)', *Archives of biochemistry and biophysics*, 638, pp. 35–40.

Kopylova, E., Noé, L. and Touzet, H. (2012) 'SortMeRNA: fast and accurate filtering of ribosomal RNAs in metatranscriptomic data', *Bioinformatics* , 28(24), pp. 3211–3217.

Köster, J. and Rahmann, S. (2012) 'Snakemake--a scalable bioinformatics workflow engine', *Bioinformatics* , 28(19), pp. 2520–2522.

Krogh, A. *et al.* (2001) 'Predicting transmembrane protein topology with a hidden Markov model: application to complete genomes', *Journal of molecular biology*, 305(3), pp. 567–580.

Kukurba, K. R. and Montgomery, S. B. (2015) 'RNA Sequencing and Analysis', *Cold Spring Harbor protocols*, 2015(11), pp. 951–969.

Lau, W. and Sattely, E. S. (2015) 'Six enzymes from mayapple that complete the biosynthetic pathway to the etoposide aglycone', *Science*, 349(6253), pp. 1224–1228.

Lee, J.-H., Yi, H. and Chun, J. (2011) 'rRNASelector: a computer program for selecting ribosomal RNA encoding sequences from metagenomic and metatranscriptomic shotgun libraries', *Journal of microbiology* , 49(4), pp. 689–691.

Levin, J. Z. *et al.* (2010) 'Comprehensive comparative analysis of strand-specific RNA sequencing methods', *Nature methods*, 7(9), pp. 709–715.

Li, B. *et al.* (2014) 'Evaluation of de novo transcriptome assemblies from RNA-Seq data', *Genome biology*, 15(12), p. 553.

Li, B. and Dewey, C. N. (2011) 'RSEM: accurate transcript quantification from RNA-Seq data with or without a reference genome', *BMC bioinformatics*, 12, p. 323.

Li, C. *et al.* (2016) 'INC-Seq: accurate single molecule reads using nanopore sequencing', *GigaScience*, 5(1), p. 34.

Lima, L. *et al.* (2020) 'Comparative assessment of long-read error correction software applied to Nanopore RNA-sequencing data', *Briefings in bioinformatics*, 21(4), pp. 1164–1181.

Love, M. I., Huber, W. and Anders, S. (2014) 'Moderated estimation of fold change and dispersion for RNA-seq data with DESeq2', *Genome biology*, 15(12), p. 550.

Luo, R. *et al.* (2012) 'SOAPdenovo2: an empirically improved memory-efficient short-read de novo assembler', *GigaScience*, 1(1). doi: 10.1186/2047-217X-1-18.

MacManes, M. D. (2018) 'The Oyster River Protocol: a multi-assembler and kmer approach for de novo transcriptome assembly', *PeerJ*, 6, p. e5428.

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

Martin, M. (2011) 'Cutadapt removes adapter sequences from high-throughput sequencing reads', *EMBnet.journal*, 17(1), pp. 10–12.

McGettigan, P. A. (2013) 'Transcriptomics in the RNA-seq era', *Current opinion in chemical biology*, 17(1), pp. 4–11.

Metzker, M. L. (2010) 'Sequencing technologies - the next generation', *Nature reviews. Genetics*, 11(1), pp. 31–46.

Miller, J. R., Koren, S. and Sutton, G. (2010) 'Assembly algorithms for next-generation sequencing data', *Genomics*, 95(6), pp. 315–327.

Mortazavi, A. *et al.* (2008) 'Mapping and quantifying mammalian transcriptomes by RNA-Seq', *Nature methods*, 5(7), pp. 621–628.

Nawrocki, E. P. and Eddy, S. R. (2013) 'Infernal 1.1: 100-fold faster RNA homology searches', *Bioinformatics* , 29(22), pp. 2933–2935.

Owen, C. *et al.* (2017) 'Harnessing plant metabolic diversity', *Current opinion in chemical biology*, 40, pp. 24–30.

Parkinson, J. and Blaxter, M. (2009) 'Expressed Sequence Tags: An Overview', in Parkinson, J. (ed.) *Expressed Sequence Tags (ESTs): Generation and Analysis*. Totowa, NJ: Humana Press, pp. 1–12.

Parmar, V. S. *et al.* (1997) 'Phytochemistry of the genus Piper', *Phytochemistry*, 46(4), pp. 597–673.

Paszkiewicz, K. and Studholme, D. J. (2010) 'De novo assembly of short sequence reads', *Briefings in bioinformatics*, 11(5), pp. 457–472.

Patro, R. *et al.* (2017) 'Salmon provides fast and bias-aware quantification of transcript expression', *Nature methods*, 14(4), pp. 417–419.

Patro, R., Mount, S. M. and Kingsford, C. (2014) 'Sailfish enables alignment-free isoform quantification from RNA-seq reads using lightweight algorithms', *Nature biotechnology*, 32(5), pp. 462–464.

Peng, Y. *et al.* (2010) 'IDBA – A Practical Iterative de Bruijn Graph De Novo Assembler', in *Research in Computational Molecular Biology*. Springer Berlin Heidelberg, pp. 426–440.

Peng, Y. *et al.* (2013) 'IDBA-tran: a more robust de novo de Bruijn graph assembler for transcriptomes with uneven expression levels', *Bioinformatics* , 29(13), pp. i326–34.

Pevzner, P. A., Tang, H. and Waterman, M. S. (2001) 'An Eulerian path approach to DNA fragment assembly', *Proceedings of the National Academy of Sciences of the United States of America*, 98(17), pp. 9748–9753.

Piétu, G. *et al.* (1999) 'The Genexpress IMAGE knowledge base of the human brain transcriptome: a prototype integrated resource for functional and computational genomics', *Genome research*, 9(2), pp. 195–209.

Priyam, A. *et al.* (2019) 'Sequenceserver: A Modern Graphical User Interface for Custom BLAST Databases', *Molecular biology and evolution*, 36(12), pp. 2922–2924.

Prjibelski, A. D. *et al.* (2020) 'Extending rnaSPAdes functionality for hybrid transcriptome assembly', *BMC bioinformatics*, 21(Suppl 12), p. 302.

Rapaport, F. *et al.* (2013) 'Comprehensive evaluation of differential gene expression analysis methods for RNA-seq data', *Genome biology*, 14(9), p. R95.

Reeb, J. *et al.* (2015) 'Evaluation of transmembrane helix predictions in 2014', *Proteins*, 83(3), pp. 473–484.

Rhoads, A. and Au, K. F. (2015) 'PacBio Sequencing and Its Applications', *Genomics, proteomics & bioinformatics*, 13(5), pp. 278–289.

Robertson, G. *et al.* (2010) 'De novo assembly and analysis of RNA-seq data', *Nature methods*, 7(11), pp. 909–912.

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

Rokyta, D. R. *et al.* (2012) 'The venom-gland transcriptome of the eastern diamondback rattlesnake (Crotalus adamanteus)', *BMC genomics*, 13, p. 312.

de la Rubia, I. *et al.* (2021) 'Reference-free reconstruction and quantification of transcriptomes from Nanopore long-read sequencing', *bioRxiv*. doi: 10.1101/2020.02.08.939942.

Russell, J. J. *et al.* (2017) 'Non-model model organisms', *BMC biology*, 15(1), p. 55.

Sahlin, K. *et al.* (2018) 'Deciphering highly similar multigene family transcripts from Iso-Seq data with IsoCon', *Nature communications*, 9(1), p. 4601.

Sahlin, K. and Medvedev, P. (2021) 'Error correction enables use of Oxford Nanopore technology for reference-free transcriptome analysis', *Nature communications*, 12(1), p. 2.

Sahraeian, S. M. E. *et al.* (2017) 'Gaining comprehensive biological insight into the transcriptome by performing a broad-spectrum RNA-seq analysis', *Nature communications*, 8(1), p. 59.

Sanger, F., Nicklen, S. and Coulson, A. R. (1977) 'DNA sequencing with chain-terminating inhibitors', *Proceedings of the National Academy of Sciences of the United States of America*, 74(12), pp. 5463–5467.

Santosh, B., Varshney, A. and Yadava, P. K. (2015) 'Non-coding RNAs: biological functions and applications', *Cell biochemistry and function*, 33(1), pp. 14–22.

Schmieder, R., Lim, Y. W. and Edwards, R. (2012) 'Identification and removal of ribosomal RNA sequences from metatranscriptomes', *Bioinformatics* , 28(3), pp. 433–435.

Schubert, M., Lindgreen, S. and Orlando, L. (2016) 'AdapterRemoval v2: rapid adapter trimming, identification, and read merging', *BMC research notes*, 9, p. 88.

Schulz, M. H. *et al.* (2012) 'Oases: robust de novo RNA-seq assembly across the dynamic range of expression levels', *Bioinformatics* , 28(8), pp. 1086–1092.

Simão, F. A. *et al.* (2015) 'BUSCO: assessing genome assembly and annotation completeness with single-copy orthologs', *Bioinformatics* , 31(19), pp. 3210–3212.

Sim, G. K. *et al.* (1979) 'Use of a cDNA library for studies on evolution and developmental expression of the chorion multigene families', *Cell*, 18(4), pp. 1303–1316.

Simpson, J. T. *et al.* (2009) 'ABySS: a parallel assembler for short read sequence data', *Genome research*, 19(6), pp. 1117–1123.

Smith-Unna, R. *et al.* (2016) 'TransRate: reference-free quality assessment of de novo transcriptome assemblies', *Genome research*, 26(8), pp. 1134–1144.

Søndergaard, D. (2019) *tmhmm.py: A transmembrane helix finder in Python* 3. Github. Available at: https://github.com/dansondergaard/tmhmm.py (Accessed: 5 August 2021).

Stark, R., Grzelak, M. and Hadfield, J. (2019) 'RNA sequencing: the teenage years', *Nature reviews. Genetics*, 20(11), pp. 631–656.

Sullivan, W. (2015) 'The Institute for the Study of Non–Model Organisms and other fantasies', *Molecular Biology of the Cell*, 26(3), pp. 387–389.

Surget-Groba, Y. and Montoya-Burgos, J. I. (2010) 'Optimization of de novo transcriptome assembly from next-generation sequencing data', *Genome research*, 20(10), pp. 1432–1440.

Sweetlove, L. (2011) 'Number of species on Earth tagged at 8.7 million', *Nature*. doi: 10.1038/news.2011.498.

Toren, D. *et al.* (2020) 'Gray whale transcriptome reveals longevity adaptations associated with DNA repair and ubiquitination', *Aging cell*, 19(7), p. e13158.

Vaser, R. *et al.* (2017) 'Fast and accurate de novo genome assembly from long uncorrected reads', *Genome research*, 27(5), pp. 737–746.

Vijay, N. *et al.* (2013) 'Challenges and strategies in transcriptome assembly and differential gene expression quantification. A comprehensive in silico assessment of RNA-seq experiments', *Molecular ecology*, 22(3), pp. 620–634.

Voshall, A. *et al.* (2021) 'A consensus-based ensemble approach to improve transcriptome assembly', *bioRxiv*. doi: 10.1101/2020.06.08.139964.

Wang, S. and Gribskov, M. (2017) 'Comprehensive evaluation of de novo transcriptome assembly programs and their effects on differential gene expression analysis', *Bioinformatics* , 33(3), pp. 327–333.

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

Weirather, J. L. *et al.* (2017) 'Comprehensive comparison of Pacific Biosciences and Oxford Nanopore Technologies and their applications to transcriptome analysis', *F1000Research*, 6, p. 100.

Wyman, D. *et al.* (2019) 'A technology-agnostic long-read analysis pipeline for transcriptome discovery and quantification', *bioRxiv*. doi: 10.1101/672931.

Xie, Y. *et al.* (2014) 'SOAPdenovo-Trans: de novo transcriptome assembly with short RNA-Seq reads', *Bioinformatics* , 30(12), pp. 1660–1666.

Zerbino, D. R. and Birney, E. (2008) 'Velvet: algorithms for de novo short read assembly using de Bruijn graphs', *Genome research*, 18(5), pp. 821–829.

Zhang, H., Jain, C. and Aluru, S. (2019) 'A comprehensive evaluation of long read error correction methods', *bioRxiv*. doi: 10.1101/519330.

Zhulidov, P. A. *et al.* (2004) 'Simple cDNA normalization using kamchatka crab duplex-specific nuclease', *Nucleic acids research*, 32(3), p. e37.

List of figures

List of tables