Norwegian University
of Life Sciences

Master's Thesis 2016 30 ECTS
Department of Chemistry, Biotechnology and Food Science (IKBM)

Optimizing Transcriptome Analysis using Short-Read RNA-Seq in Atlantic Salmon

Katrine Hånes Kirste
Bioinformatics and Applied Statistics

Acknowledgements

The thesis was performed during spring 2016. Nofima (Norwegian Institute of Food, Fisheries and Aquaculture) has provided the transcriptomic data, which were analysed at the CIGENE (Center for Intergrative Genetics) /NMBU Orion cluster environment.

I want to thank my supervisors, Professor Torgeir R. Hvidsten, NMBU and Dr. Hooman Moghadam, Nofima, for encouraging me to write the thesis and for introducing me to the world of bioinformatics. Thanks to my co-supervisor Dr. Kristian Hovde Liland, NMBU/Nofima for good support in the writing process and programming in R.

I also want the use the opportunity to thank the Norwegian University of Life Science for giving excellent courses and supplying me with lots of interesting knowledge. Thanks to my colleagues at Nofima for being supportive and encouraging me to do the work.

Last, but not least, I want to thank Ole and Tarjei for all the love and support from home – it means a lot to me.

Katrine Hånes Kirste

June 15, 2016

Sammendrag

Målet med studiet var å sammenligne genome-guided assembly og de novo assembly av transcriptomdata, for å finne ut hvilken metode som burde anbefales. De novo assembly er mye mere krevende angående datalagringsplass og tidsbruk, og resultatet av denne metoden bør være signifikant bedre for å rettferdiggjøre bruk av metoden.

Gjennom en rekke dataanalyser og optimaliseringssteg ble metodene sammenlignet på tre ulike vis; read mapping ratio til genom og transkripter, BLAST hits av transkripter til genom og proteinsekvenser, samt antall og ratio av uttrykte gener i genom-guided assembly og BLAST-hits til proteinsekvenser.

De novo assembly utvidet annoteringen signifikant, i vårt tilfelle med mer enn 20,000 transkripter. Genome-guided assembly ga bedre mapping, men metoden gir ingen nye oppdagelser i sekvensene. De novo assembly anbefales for utvidet annotering av gener.

Abstract

The aim of the study was to compare genome-guided and de novo assembly of transcriptomic data, to find out which method to recommend. De novo assembly is much more demanding regarding computational storage and time, and the result of this method should be significantly better to justify using this method.

Through a number of computational analysis and optimization steps, the methods were compared in three different matters; read mapping rate to genome and transcripts, BLAST hits of transcripts to the genome and protein database and number and ratio of expressed genes in genome-guided assembly and BLAST hits to protein database.

De novo assembly extended the annotation significantly, in our case with more than 20,000 transcripts. Genome-guided assembly gave better mapping, but the method does not give novel discoveries. De novo assembly is recommended for an extended annotation of genes.

Innhold

1. Introduction

1.1. Objectives for the study

Aim of the project

The objectives for this study is to compare genome-guided assembly and de novo assembly in transcriptome analysis in Atlanic salmon *(Salmo salar)*. By comparing the genetic information gained from these methods, the aim is to be able to give an answer the question: "Will de novo assembly be useful in future studies?"

De novo assembly is the assembly of sequences without the use of a reference. It can therefore provide new information about the organism we are studying, such as novel genes and isoforms. Genome-guided assembly does not have this opportunity, as it uses the genome as a reference for the mapping of the reads, and only familiar sequences will be assembled. De novo assembly undergoes challenging algorithms and involves several processing steps as compared to genome-guided assembly. It is therefore relatively demanding regarding time and computing resources, and we want to find out if it is worth the extra effort doing de novo assembly.

The salmon genome

Salmon has a highly duplicated genome, and studying the transcriptome of is therefore challenging. The salmon ancestors has gone through at least three rounds of whole-genome duplication some 80 million years ago (Lien et al., 2016). Salmon is considered being pseudo-tetraploid (Davidson et al., 2010), as they can have a quadruple set of chromosomes, and are in the constant process of reverting to a stable diploid state.

Figure 1: Atlantic salmon (Salmo salar) (*http://www.asf.ca/main.html*)

The Atlantic salmon RefSeq assembly; accession GCF_000233375.1 (ICSASG_v2), was used as a reference. The assembly has a total sequence length of 3 Gb, and consists of the 29 chromosomes, mitochondrial DNA (Chr MT; NC_001960.1) and unplaces scaffolds larger than 1,000 bases (http://www.ncbi.nlm.nih.gov/assembly/487001). The RefSeq protein database has 97,738 proteins, while the RefSeq annotation file used for the genome-guided assembly has 81,586 genes (http://www.ncbi.nlm.nih.gov/genome/369?genome_assembly_id=248466).

Figure 2: Illustration of the 29 salmon chromosomes.
(http://www.ncbi.nlm.nih.gov/genome/369?genome_assembly_id=248466)

Samples

Illumina (1) short PE-reads of 16 individual transcriptomes of Atlantic salmon was used for the study. The samples were from different tissues (liver samples or whole fish), life stages (embryo, post smolt or adult fish) and treatments (different diets of fatty acids or pancreas decease (PD) virus infection). In the analysis, all samples were pooled together, as the objectives of the study is to compare expression in assembly methods rather than look at differential expression. It is therefore not put any emphasis on the sample type is this work.

Bioinformatic methods and evaluation

Several bioinformatics tools were used, and programs and program codes are further described in the Methods part. The programs are listed on page 28 and referred to by numbers in brackets.

The raw reads were first quality checked in FastQC (6) and trimmed and filtered, using Trimmomatic (7). Preprocessing is essential due to validation of downstream analysis, and is therefore carefully described in this thesis.

Trinity (8) was used to produce the splice variants in the de novo assembly. The splice variants were grouped into genes by Trinity, based on sequence similarity. These gene clusters are referred to as 'Trinity-genes' in this thesis.

TopHat2 (10), which uses Bowtie2 (9) as an aligner, was used for mapping of the reads to the RefSeq genome and to the splice variants. The alignment rate of these two assemblies was compared and discussed.

The read mapping information was also used for filtering out low expressed splice variants. Bedtools (11) was used for measuring the read depth. The expression, based on fpkm > 1 in at least two samples, was calculated using EdgeR (5). The dataset was further reduced by clustering, using CD-HIT-EST (12) on a 99% similarity threshold.

BLAST (13) algorithms was used for alignment of the splice variants to the salmon genome and protein database. The results were compared and evaluated, and the issue regarding new discoveries is commented on in the Discussion part.

Genome-guided assembly was done by assembling the RefSeq annotation file with the alignment files produced in the read mapping to the genome. Cufflinks2 (14), which was used for the assembly, outputs fpkm-files for genes and isoforms. The genes and isoforms with expression > 1 fpkm in two or more samples were reported as expressed genes and isoforms.

Figure 3 describes an overview of the study.

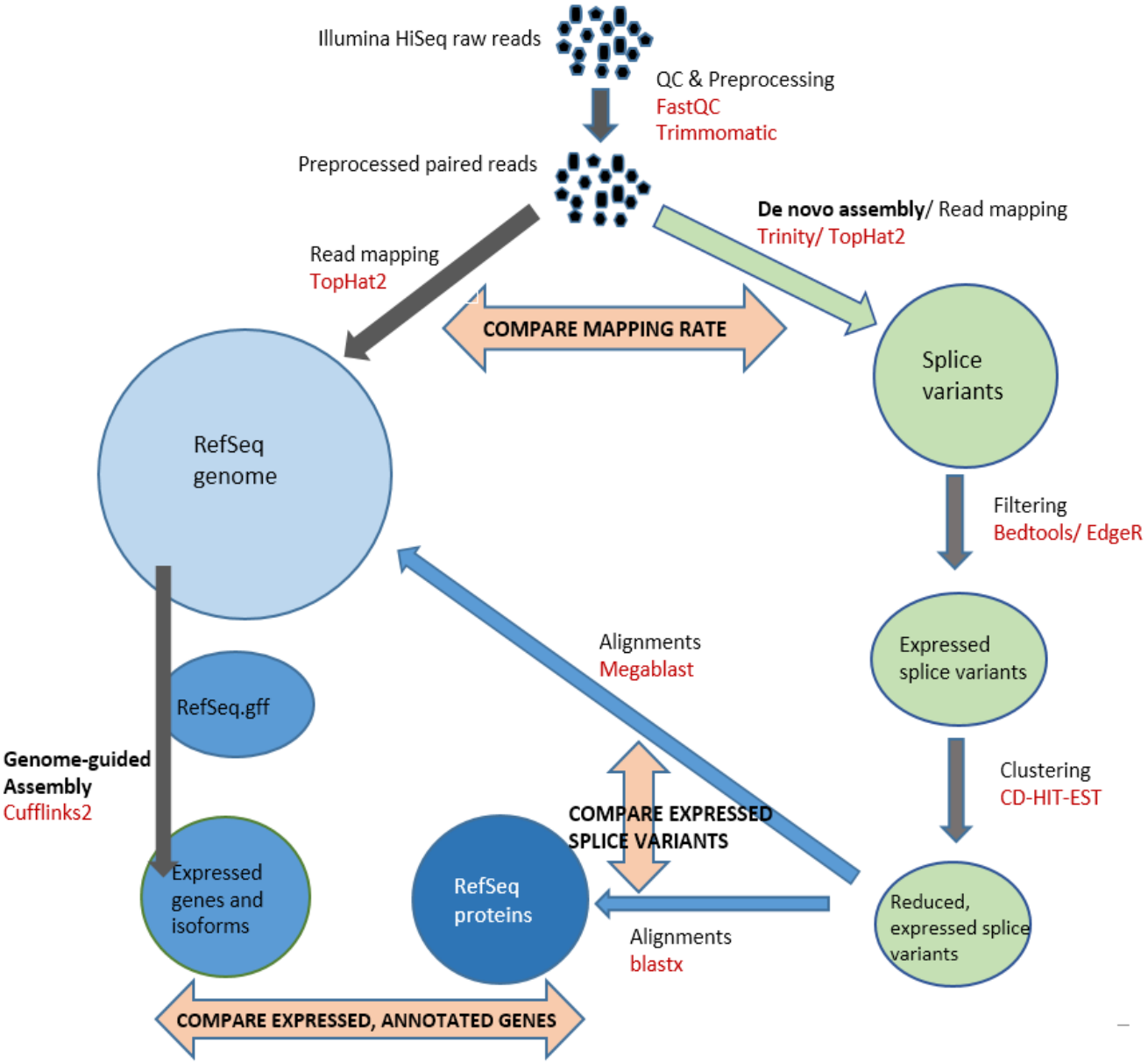


Figure 3: The different analysis steps in the comparison of genome-guided assembly and de novo assembly. The different datasets or databases are shown as circles, and the analysis steps are shown as arrows. The analysis steps or data are described in black typing, and the programs used, are in red typing.

1.2. Transcriptome analysis

The transcriptome

Implementation of next generation sequencing (NGS) technologies has increased the possibilities for studying the genome and its functions, with data throughput increasing more rapidly than in any other science field.

The transcriptome a collection of RNA molecules produced in a cell. Most commonly they are produced in the process of gene expression, like messenger RNAs (mRNA), which are transcripts of the coding part of the gene (in eukaryots; the exons), and functions as a template in the production of a protein or peptide. It can also be other RNA molecules, like transfer RNA (tRNA) (http://www.ncbi.nlm.nih.gov/genome/annotation_euk/Salmo_salar/100/), which serves as a physical link between the mRNA and the amino acid sequence of proteins, (https://en.wikipedia.org/wiki/Transfer_RNA), or long non-coding RNA sequences (lncRNAs).

The transcriptome gives real-time information on the processes going on in the organism. We can for example study the expression of genes in one tissue versus another, variation of gene expression over time or during decease or treatment. In these studies, differential expression is the measure normally used. A gene is differentially expressed when expression of the gene is significantly higher or lower.

RNA sequencing (RNA-Seq)

RNA-Seq is reading the order of the four building bricks of ribonucleic acids (RNA); adenosine (A), uracil (U), guanine (G) and cytosine (C), and using this information to determine the identity and abundance of the sequences, using experimental and computational methods (Korpelainen et al., 2015). The transcripts are converted to cDNA libraries before sequencing, and every platform has its own library preparation method. RNA-seq is much more sensitive and specific than the traditional microarray method for measuring gene expression, as microarray has a lower detection range due to noise and a saturation problem when expression is high. RNA-seq also has the possibility to discover new genes and isoforms, as the reads can be assembled without the use of a template reference in de novo assembly.

Sanger sequencing was the dominating sequencing method for decades, before the Next Generation Sequencing (NGS) methods were introduced, around 2005. It is quite accurate, but has low throughput compared to NGS. It uses FASTA-format, which is a text format consisting of two lines per sequence, one header which identifies the sequence and one with the sequence.

The second generation sequencing methods, or NGS, uses sequencing by synthesis, and short fragments of 30-150 bases are being amplified in massive parallel format. The expression ‘deep sequencing’ refers to the large overlap of reads that is produced during alignment. Popular second generation platforms are Roche 454 (water-oil emulsion PCR amplification), Illumina (bridge-amplification), SOLID (sequences by ligation) and Ion Torrent (measures change in pH as a result of change in electric current when incorporating a new base). Longer fragments are produced by the third generation sequencing platforms, which also differs from second generation in the sequencing and detection chemistry (Korpelainen et al., 2015). The methods are able to detect one single molecule rather than amplified clusters (single-molecule sequencing by synthesis).

The third generation sequencing platforms are Pacific Biosciences PacBio (SMRT – single molecule real time) and Nanopore Technologies (differences in electric current measured as a cDNA molecule is being passed through a membrane).

1.3. Quality control

Base quality score

The NGS platforms introduced the FASTQ format, which is a file format based on FASTA format (which is described in the last section), but with two extra lines. The third line is an optional line for header, starting with +, and often not used. The last line shows the base quality score values in ASCII characters, one for each nucleotide in the sequence (https://en.wikipedia.org/wiki/FASTQ_format).

The quality of sequence data is often measured using the Phred scale, which is a scale that measures the probability of a base being wrong (http://drive5.com/usearch/manual/quality_score.html).

Each nucleotide in an alignment gets a Phred score, denoted Q:

$$Q = (-10)\ log_{10} P$$

,where P is the estimated probability of a nucleotide being wrong, i.e. probability of error. For instance:

- If Q = 10, then P = 0.1 and estimated correctness is 90%
- If Q = 20, then P = 0.01 and estimated correctness is 99%
- If Q = 30, then P = 0.001 and estimated correctness is 99,9%

The Phred scale is used differently for the different sequencing platforms. Illumina version 1.9 uses Phred+33 and the quality scale goes from 0.2 to 41. The quality values for Illumina 1.9 are:

!"#$%&'()*+,-./0123456789:;<=>?@ABCDEFGHIJ,

where ! shows the lowest Q score and J shows highest possible Q score.

Preprocessing and quality control

Quality control of the reads sequences prior to further analysis is essential, as poor sequences will otherwise pass on the problems in the downstream analysis and end up giving the wrong conclusions. Often, the sequences contain errors, and preprocessing is needed. Preprocessing of reads can involve trimming, i.e. cutting off the part of the read with poor quality, or it can involve filtering, which is throwing away the whole read(Korpelainen et al., 2015).

Sequencing bias

Poor sequence quality is often caused by base incorporation problems, i.e. the wrong base has been added to the sequence during the sequencing reaction (Korpelainen et al., 2015) . Many NGS platforms, like Illumina, sequences clusters of fragments which origin from the same template. Theoretically, one cluster should give the same sequence. In practice, the sequencer makes mistakes, especially at the 5' end. These mistakes can be discovered by aligning all sequences within a cluster, and, based on this information, calculate the probability of a base being wrong. This probability is background for calculation the Q score and the selection of the best possible sequence. For very poor sequences, the probability of each nucleotide is approximately the same. The sequence will therefore be impossible to decide, and these ambiguous bases are by most platforms denoted N. The poor sequences are usually found at one or both of the ends of the read, and should be removed, as they will infer with the alignment later in the process.

Adapter sequence contamination

Sequences that does not origin from the transcripts but from some other contaminate templates should be removed. These sequences could be adapters or primers used by the sequencing platform. Sequenced adapter occurs when the fragment is shorter than the pre-defined read length (7).

Contamination of rRNA

Other unwanted sequences are fragments of ribosomal RNA (rRNA), which is the most redundant type of RNA molecule present in the cell. Contamination of rRNA can occur for instance when the washing step during library preparations has not been performed optimal.

GC-content

When checking the quality of the sequences, the ratio of Guanine and Cytocine should be calculated and compared with the expected ratios. GC content is generally higher for transcriptomes than for genomes, as non-coding DNA often has a larger ratio of A and T. For instance, the salmon genome has an estimated GC content of 43.9 % (http://www.ncbi.nlm.nih.gov/genome/?term=atlantic%20salmon), which is significantly less than the GC content for the salmon transcripts in this study, which is around 48% in the raw data and 47% in the splice variants produced in the de novo assembly. The GC content can be used as a quality indicator, e.g. rRNA has usually has more GC than mRNA, and therefore high GC content can indicate pollution of rRNA in the sample.

Unprocessed mRNA (pre-mRNA)

The extraction of total RNA might also pick up some unprocessed mRNA, which, in eukaryots, is mRNA still containing introns. This is difficult to control for, as there is apparently nothing wrong with the sequence when going through the regular QC steps. Library preparation methods typically use the polyA-tail, which is present in all mRNA molecules, as ligation sequence to extract mRNA from the other RNAs. Poly-A tails are also present in pre-mRNAs, so this step does not remove these sequences. Introns might have a lower GC content, and it might be possible to get an idea of the state of the sequencing sample by measuring GC. Still, the ratio of pre-mRNA versus spliced mRNA is assumingly too low to detect this.

Trimming and filtering of reads

A desired mean quality score (Q) for the sequence is commonly set as a threshold in the preprocessing. If the entire sequence is of such poor quality that that it does not reach the desired threshold, the whole sequence will be automatically thrown away. Sequences with mean Q fulfilling threshold will remain, but each nucleotide will in addition be tested for Q, and part of the sequence below this threshold will be trimmed off.

Trimming involves cutting off the poor part of the sequence, either it be from the 3’ end or from the 5’ end. Poor base distribution is a typical bias which gives reason for trimming. Often, a random nucleotide hexamer is used as primer for the cDNA synthesis. Theoretically, this should give random binding and synthesis should give an even distribution of the bases over the sequence. In practice, the randomness is not perfect and this will appear as poor base distribution in the 3’end of the sequence. In FastQC this is shown in the diagram “Sequence content across all bases”, and the number of bases for trimming can easily be extracted from the diagram. Ambigous bases can appear, especially in the 5’ end of the sequence, and should be removed. The diagram “Per base N content” in the FastQC report shows how many bases to remove.

There can be different reasons for filtering, i.e. removing the entire sequence. Short sequences should be filtered, as they are more challenging to align correctly and errors might occur. This is even more applicable for species with duplicated genomes, like for instance salmon. Other unwanted, short RNA sequences might be present in the sample, which is another argument for only keeping sequences of a certain length. Sequence with low complexity, for instance numbers of repetitive base structures, should be trimmed or filtered, as these might easily align to the wrong sequence.

1.4. Sequence alignment

Homology

It is important to distinguish between sequence similarity and sequence homology. Homology means shared ancestry (https://en.wikipedia.org/wiki/Homology_(biology)), and homologue sequences are sequences which origin from the same genus or species. Orthologues are homologues caused by species divergence, i.e. the same sequence is found in different, but related species. Orthologue genes have therefore the same function, but in different species. Paralogues are homologues that are created by a duplication event within the genome. It is usually within the same species, but can be in different species if the duplication event has happened before the species divergence. It is common that paralogue genes have different functions, as changes in the genome might have happened after the duplication. Ohnologues are paralogues caused by whole-genome duplication, which is the case for Atlantic salmon (Lien et al., 2016). When studying the genome and transcriptome, looking for homology is of great interest, as it can give information about the evolution of the species and changes of the genome over time, such as duplications, inversions and recombination. Homology is a matter of quality and cannot be measured, as sequences are either homologues or not. Homology is rather the conclusion made after observing sequence similarity based on alignment, as significant similarity is often strong evidence of homology (wiki).

Sequence similarity

Sequence identity, the simplest form for similarity measure; is the number of matching nucleotides in a part of the sequences (sub-sequence) given an alignment. It can typically be explained as percent identity, i.e. the ratio of nucleotide matches over the sequences.

Similarity score is another, frequently used measure for sequence similarity. The sequence nucleotides are compared and scored according to a scoring matrix. Each match or mismatch gets a score, and the sum of scores gives a scoring value which identifies the similarity of the sequences. Gaps, which indicates indels or sequencing errors, always get negative scores, while mismatch can be given a positive score, depending on the scoring rules

The log likelihood ratio score measures the probability of homology in a sequence pair as compared with the probability of homology for any independent sequence. When probability of homology between two sequences is larger the overall probability of homology in the independent sequences, the ratio is larger than one, which will give a positive log likelihood ratio score.

Log likelihood ratio score:

$$score(seqA, seqB) = log\,10\frac{P(seqA, seqB \mid homology)}{P(any\ pair \mid independence)}$$

2. Materials and Methods

2.1. Computing resources

Sequence data analyses was performed using resources at the Orion Computing Cluster at CIGENE-NMBU (Center of Intergrative Genetics, Norwegian University of Life Science). All bioinformatics tools were open source programs available on the cluster.

Text editor GNU nano (2) version 2.0.9 was used for scripting.

R version (3) 3.2.4 Revised (2016-03-16 r70336) was used as data editor and analysis tool on local computer, and RStudio (4) version 0.99.489 https://www.rstudio.com/ was used as integrated development environment (IDE). The R-packages micropan and data.table were used for handling FASTA files and very large files, respectively.

MS Office 2013 was used as a tool for writing the thesis.

2.2. Sequence data

16 individual Atlantic salmon *(Salmo salar)* transcriptome sequences, provided by Nofima, was used for the comparative study. The samples were from different tissues (liver samples or whole fish), life stages (embryo, post smolt or adult fish) and treatments (different diets of fatty acids or PD (pancreas decease) virus infection), and were pooled together for the analysis.

Llumina (1) HiSeq version 1.9 was used to produce the 100-101 nucleotides paired end (PE) reads. The size of the FASTQ files were in the range of 0.9 – 3.4 GB, with a total of 52.3 GB. The number of raw reads per sample was 19.1 ± 5.6 million, with a total of 305,535,951 reads for all samples. All sequences were of generally good quality, with GC content of 48 ± 1 %.

2.3. Template files

Illumina (1) HiSeq adapter sequences were supported by Trimmomatic (7), and were available on the cluster. The adapter.fa file is given in Appendix A.

The Atlantic salmon ICSASG_v2 RefSeq assembly and protein database, accession GCF_000233375.1, were downloaded from ftp://ftp.ncbi.nlm.nih.gov/genomes/all/GCF_000233375.1_ICSASG_v2.

The RefSeq assembly has a total sequence length of approximately 3 Gb, and consists of the 29 chromosomes, chromosome MT (mitochondrial DNA) and all unplaced scaffolds of 1000 nucleotides or larger. The unplaced scaffolds are genes not yet mapped to a chromosome, and therefore does not yet have a chromosome number. The RefSeq protein database has 97,738 proteins.

2.4. Quality control and preprocessing

Quality control - FastQC

FastQC (6) version 0.11.3. was used for quality control of the sequences prior and post preprocessing. Based on the quality information in the FASTQ files, FastQC produces graphical reports in html format. The reports show basic statistics, such as total number of sequences, number of poor quality sequences, sequence length and GC content. It visualizes sequence quality per base and per tile, and shows the distribution of Q scores, nucleotides, GC content and sequence length. Sequence ambiguity, k-mers, adapter sequences and duplications are reported, and overrepresented sequences are being listed when exceeding 0.1 % of the total number of sequences. If a sequence is overrepresented, FastQC will make a search towards a built-in database containing possible sources, such as Illumina adapters and primers sequences.

Run FastQC:

```
fastqc –t 8 –o Sample1.fastq.gz
```

Based on the raw read reports, each sequence file was evaluated for further improvement. The reports are useful when deciding parameters for preprocessing, e.g. threshold Q value or weather the sequences need trimming or not. The information on overrepresented sequences gives the opportunity to investigate these prior to preprocessing, and remove them if they appear to be unwanted sequences, like e.g. Illumina adapters, ribosomal RNA or other bias to the library preparation or sequencing method. Comparing reports of the processed reads contra raw reads reveals weather the preprocessing was successful or not.

Preprocessing - Trimmomatic

Trimmomatic (7) version 0.33 was used for the preprocessing of the reads. The program is fast and flexible, and uses quality score as a correction parameters (Bolger et al., 2014). It can crop off nucleotides at one or both ends of the read, and filter the reads based on read length. It also has the opportunity to remove predefined, unwanted sequences, like e.g. Illumina adapters. In PE reads, it uses an algorithm called palindrome mode to remove the whole or part of the adapter sequences. This is done by using the information from one pairing read (e.g. forward) to match the corresponding information in the other pairing read (reverse of the same read) to decide what should be removed. Poor reads has less impact on the filtering, as the alignment score is relative to the base quality.

The preprocessing step was optimized on a few representative samples to fit all samples. FastQC reports were used as background for the optimization, with a special emphasize on the parameters “Per base sequence quality”, “Per tile sequence quality”, “Per base sequence content”, “Per base N content” and “Overrepresented sequences”. Overrepresented sequences with no hits in the raw data were investigated by running NCBI-BLAST (13) with default settings. The sequences were first assembled to Atlantic salmon, and if no significant hit was gained it was assemble to the non-redundant database. If sequence had significant hit for ribosomal RNA in any organism, the sequence was added to the adapter file in Appendix A

Run Trimmomatic:

```
trimmomatic PE -phred33 input_seq_R1.fastq.gz input_seq_R2.fastq.gz
output_seq_1P.fastq.gz output_seq_1U.fastq.gz output_seq_2P.fastq.gz
output_seq_2U.fastq.gz ILLUMINACLIP: adapters.fa:2:30:10:3 SLIDINGWINDOW:4:20
MAXINFO:70:0.7 *CROP:95 HEADCROP:15 MINLEN:70
```

*CROP was used for sequences with ambiguous bases at the 5’ end only.

2.5. De novo assembly

Trinity's In silico Normalization

A normalization step in order to validate and reduce the number of reads is supplied in the Trinity package, and normalized prior to de novo assembly is recommended for large datasets (https://github.com/trinityrnaseq/trinityrnaseq/wiki/Trinity-Insilico-Normalization). In silico read normalization works as a pre-filter to Trinity (http://dx.doi.org/10.6084/m9.figshare.98198). The program converts the sequences to FASTA format and makes nucleotide k-mers of 25 bases. It involves a program called Jellyfish, which counts the k-mers and stores the count data in intermediate statistical files. The stat files are background for choosing the reads within a minimum coverage threshold of 20 reads. Low coverage reads are often created by sequencing errors, and the normalization step gives the opportunity to reduce data and get rid of sequencing errors.

The script 'insilico_read_normalization.pl', which is part of Trinity package (8) version 2.0.2, was used for normalization of the preprocessed, paired reads. The script can handle both FASTA and FASTQ format, and outputs the same format as the input file. For our data, the normalization was done in two steps due to limited storage on the server.

In the first run, all the reads were entered into the code:

```
insilico_read_normalization.pl --seqType fq --JM 10G --max_cov 30
--left Sample_01_1P.fastq.gz, (etc., listing up all 1P files)
--right Sample_01_2P.fastq.gz, (etc., listing up all 2P files)
pairs_together–output Trinity_normalize–CPU 8 --PARALLEL_STATS
```

The program run until the intermediate FASTA files, left.fa (all forward reads) and right.fa (all revers reads), were produced.

Running the script for the second time, the FASTA files produced in the first round were entered into the code, and the script was run until finishing successfully:

```
insilico_read_normalization.pl --seqType fa --JM 10G --max_cov 30
--left left.fa --right right.fa --pairs_together
--output Trinity_normalize_fa --CPU 8 --PARALLEL_STATS
```

De novo assembly - Trinity

Trinity (8) version 2.0.2 was used for de novo assembly. Trinity consists three programs modules; Inchworm, Chrysalis and Butterfly (Haas et al., 2013). First, Inchworm assembles the reads to longer contigs. It first decomposes the reads into k-mers of 25 nucleotides to make a k-mer dictionary. It starts with the most abundant and complex k-mer as a seed to form the first contig, and extends the sequence on the 3'end based on the coverage of overlapping k-mers. It iterates to form new contigs until all k-mers has been processed. The Inchworm algorithm is greedy and efficient, and makes the splice variants, which are passed on to the next program. Chrysalis groups the contigs together via overlapping k-1 mers to forms de Bruijn graphs. One de Bruijn graph represents a gene with all its isoforms. At last, Butterfly compacts the de Bruijn graph to the most probable path, and then compacts the graph with the reads to reconstruct the isoforms. Unbranched structures are pruned to avoid sequencing errors. The final product is a reconstruction of the alternatively spliced isoforms, presented as sequences in a FASTA file. The Trinity algorithm can take several days to run, but is efficient regarding its sensitivity.

The normalized fasta files of left and right sequences produced the normalization step were entered into the Trinity code:

```
Trinity --seqType fa --max_memory 10G
--left left.fa.normalized_K25_C30_pctSD200.fa
--right right.fa.normalized_K25_C30_pctSD200.fa
--CPU 8 --no_bowtie
```

The Trinity output file, Trinity.fasta, was imported into R for further processing. The number of splice variants, gene clusters ('Trinity-genes', with suffix _G in the header), nucleotides, GC content and sequence length distribution was extracted from the data and reported.

2.6. Read mapping

Version 2.0.12 of TopHat (10) was used for mapping of the reads to the genome and the splice variants.

TopHat (Trapnell et al., 2012) uses Bowtie as an alignment engine, and version 2 can handle gaps (Korpelainen et al., 2015). It is therefore useful for assembly to eucaroyte genomes, as these contains exons and an aligner that can handle gaps is necessary.

TopHat-index of the databases had to be build prior to the assembly. The bowtie2-build function in Bowtie2 version 2.2.3 was used:

```
bowtie2-build -f database.fa database
```

For the TopHat assembly, this code was used:

```
tophat -p 8 -o TopHat_Sample1 database_index/database.fa Sample1_1P.fastq
  Sample1_2P.fastq
(etc, for all samples)
```

2.7. Filtering of splice variants

Filtering was done to deduce low expressed genes and possible errors from the dataset. The Trinity output file carries no information on the expression of the splice variants. Read mapping to the transcripts will provide this information. By using an expression value based on read depth, transcripts with low expression will be filtered out.

Assembly of the reads to the transcriptome – TopHat2

The first step in this process is aligning the reads to the transcriptome. TopHat2 (10) was used as an aligner, and the method for the transcriptome-guided assembly is described in Chapter 2.6.

Computation of read coverage - Bedtools

The coverageBed option in Bedtools (11) version 2.23.0 was used for computation of the read coverage. The program computes the depth and breadth of coverage of features in the alignment files.

Each samples was processes separately. The bam-format alignment files from the assembly and the Trinity result file, which was converted to bed-format, were entered into the code:

```
coverageBed -abam TopHat_denovo_Sample1/accepted_hits.bam
-b R_Trinity.bed > cov_sample1.txt
```

The program produces text files with coverage information for all splice variants, e.g. number of reads and nucleotides covering each splice variant, the sequence length and the fraction of nucleotides covered in each splice variant.

Filtering - EdgeR

EdgeR (5) is a Bioconductor software package for empirical analysis of gene expression in R (3). It is designed to work for actual read counts, and was used for the evaluation and filtering of low expressed splice variants from the dataset.

The files produced in the previous step were imported into R. A matrix consisting of the read counts was made, rows representing the contigs and columns representing the number of reads in each sample.

The matrix was used as input in the DEGList function in R:

mcount <- DGEList(counts = matrix_reads)

The DGEList output consists of a matrix of the counts (similar to the input matrix), and a table describing the library size for each sample.

The DEGList was used as input in the rpkm function, which calculates the expression values. Rpkm gives the same information as fpkm in PE reads, and is a frequently used as measuring unit for gene expression. An index vector for selection of the expressed contigs was made. The threshold for expression was set to fpkm > 1 in at least two samples:

idx <- rowSums(rpkm(mcount)>1) >= 2

The reason for selecting 1 fpkm as minimum expression, is because it is a commonly used threshold, and, in most cases, represents a reasonable expression level (Hooman M, personal communication). Expression in at least two samples was set as a minimum to reduce bias.

The index vector was used to make a new dataset. The dataset was matched with the Trinity data to make a new dataset. This dataset, consisting of Trinity headers and sequences representing splice variants, was saved in FASTA format and exported back to the server to be used for further analysis.

Clustering – CD-HIT-EST

To reduce the dataset even more, CD-HIT-EST in the CD-HIT (12) software package version 4.6.1 was used.

CD-HIT-EST clusters nucleotide sequences (Li and Godzik, 2006) by using a greedy incremental algorithm. The sequences are first sorted on length, and the longest sequence forms the first cluster. The remaining sequences are compared to the cluster and being grouped into it if the similarity is above a certain threshold. If it is below the threshold, it forms a new cluster. The program goes through the list of sequences several times until all sequences have been clustered.

A high similarity threshold was desired on our clustering, as we did not want to lose too much information in the data. The settings were set to 99% sequence similarity, and a word length of 10 nucleotides.

```
cd-hit-est -i Trinity_filtered.fa -o denovo99_rpkm.fa -c 0.99 -n 10
```

2.8. Alignment of splice variants to RefSeq databases - BLAST

Blast+ (13) version 2.3.0 was used for comparing the splice variants to the RefSeq genome and the RefSeq protein database. BLAST (Altschul et al., 1990) is a popular tool for comparing nucleotide- and amino acid sequences. It can be used on-line on the NCBI web site or by scripting, which was done in this case. The rapid and heuristic algorithm performs local alignment between sequences by using a maximum segment pair (MSP) score. It also produces an expectation value (e-value), which estimates how many matches would have occurred at a given score by chance. E-value is useful for filtering out low confidence hits, and is often used as a threshold parameter in the alignments. BLAST has different tools for different use, depe type of sequence that is being compared (nucleic acids or amino acids), or the aim of the assembly. In alignment search within the same species, high similarity is required to give significant hits, while searching for orthologues will require less similarity to give hits of interest. (http://www.ncbi.nlm.nih.gov/books/NBK153387/)

Indexing of database prior to BLAST

The RefSeq databases (genome and proteins) were indexed prior to the BLAST assembly, using the makeblastdb tool in Blast+:

```
makeblastdb -in RefSeq_database.fa -parse_seqids -dbtype nucl
```

Assembly of reference genome - Megablast

Megablast was selected for alignment of the splice variants to the genome, as this algorithm is more sensitive to sequence alignments in sequences with high similarity than regular BLAST. For better specificity of the BLAST, a maximum e-value of 10^{-6}, which is a commonly accepted threshold.

```
blastn -task megablast -db RefSeq_genome.fa -query Trinity_filtered.fa
-out results_megablast_genome -outfmt 6 -evalue 0.000001
```

Assembly to protein sequences – blastx

The blastx algorithm compares nucleotide sequences to amino acid sequences. To be able to compare sequences of different formats, it translates the nucleotide sequences to amino acids in all six reading frames. The blastx is therefore more time consuming than the blastn algorithms. The same threshold e-value as in Megablast was used for the alignment:

```
blastx -db RefSeq_proteins.fa -query Trinity_filtered.fa
-out results_blastx_proteins -outfmt 6 -evalue 0.000001 -num_threads 16
```

2.9. Genome-guided assembly

Version 2.2.1 of Cufflinks (14) was used for the assembly of hits from the read mapping to the RefSeq annotation file.

Run Cufflinks:

```
cufflinks -p 8 -G GCF_000233375.1_ICSASG_v2_genomic.gff
-o Cufflinks_Sample1 TopHat_Sample1/accepted_hits.bam
```

3. Results

3.1. Preprocessing

Descriptive Statistics

The total number of reads before and after preprocessing is given in Table 1. All raw and preprocessed reads were flagged as good quality sequences in the FastQC reports.

Table 1. Total number of reads before and after preprocessing with Trimmomatic, including the survival rates (or death rate, for the dropped reads).

Reads	*Total number*	*Survival (or death) rate*
Raw	305,535,951	
Total Surviving	280,628,529	91,8 %
Paired Surviving	226,987,777	74,3 %
Forward Only Surviving	38,345,699	12,6 %
Reverse Only Surviving	15,295,053	5,0 %
Dropped	42,741,420	14,0 %

Sequence length

The read lengths were reduced, from 100-101 nucleotides in the raw reads to 70-86 nucleotides in the preprocessed reads. The majority of the reads were at the length maximum, as shown in Figure 1.

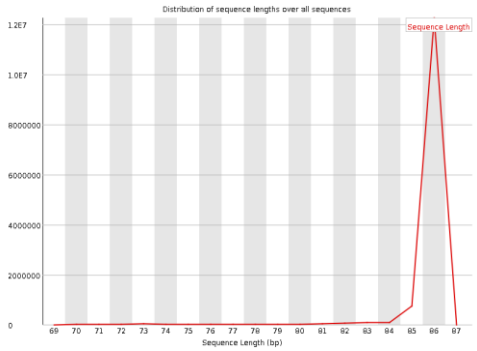


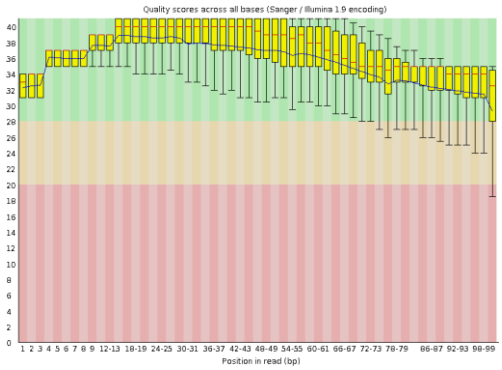
Figure 4. The FastQC diagram shows the sequence length distribution for a representative sample after preprocessing.

GC content

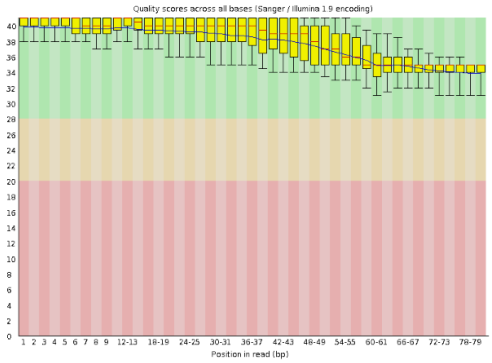
The mean GC content was 47.3 ± 1.0 % for the paired reads and 48.7 ± 1.7 % for the unpaired reads.

Sequence quality

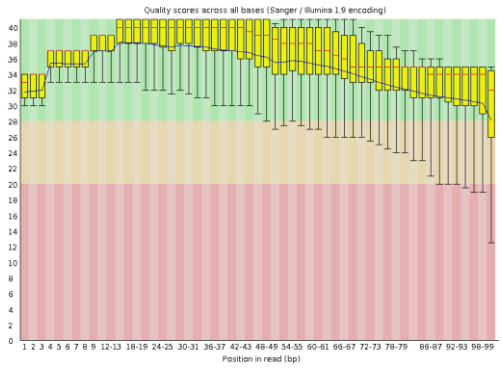
The raw sequences were generally of good quality, with a median Q score larger than 30 for all samples. Still, some reads had bases with Q score between 2 and 20, with the poor quality mainly located at the 5'end. After filtering and trimming, the minimum Q score was no lower than 28. Some examples of per base sequence quality are presented in Figure 2.



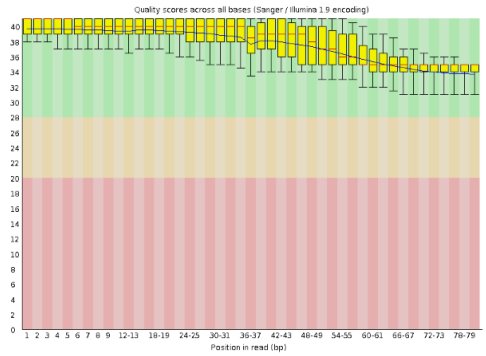
A. R1 raw sequences



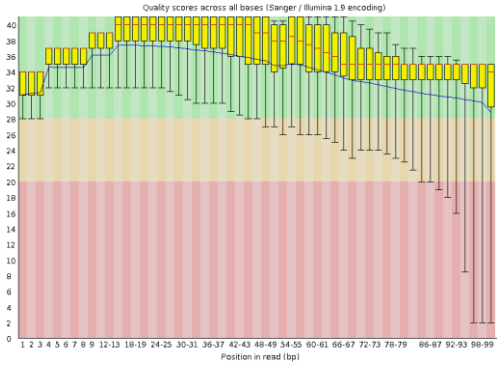
B. Sequences in A preprocessed



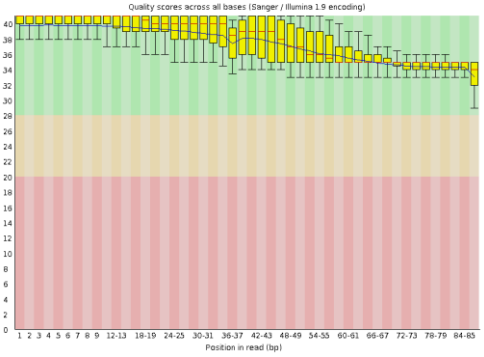
C. R2 raw sequences



D. Sequences in C preprocessed



E. R2 raw sequences



F. Sequences in E preprocessed

Figure 5. The FastQC diagrams shows per base sequence quality, represented by Q values (Phred score). The diagrams show quartiles (yellow boxplots), mean (blue graph) and median (red graph).

A – D shows per base sequence quality from one representative sample. A and C are the raw sequences, where R1 is the forward sequence and R2 is the reverse. B and D shows per base sequence quality after preprocessing in the same samples as in A and C, respectively. E and F shows a sample with poorer quality than the previous one.

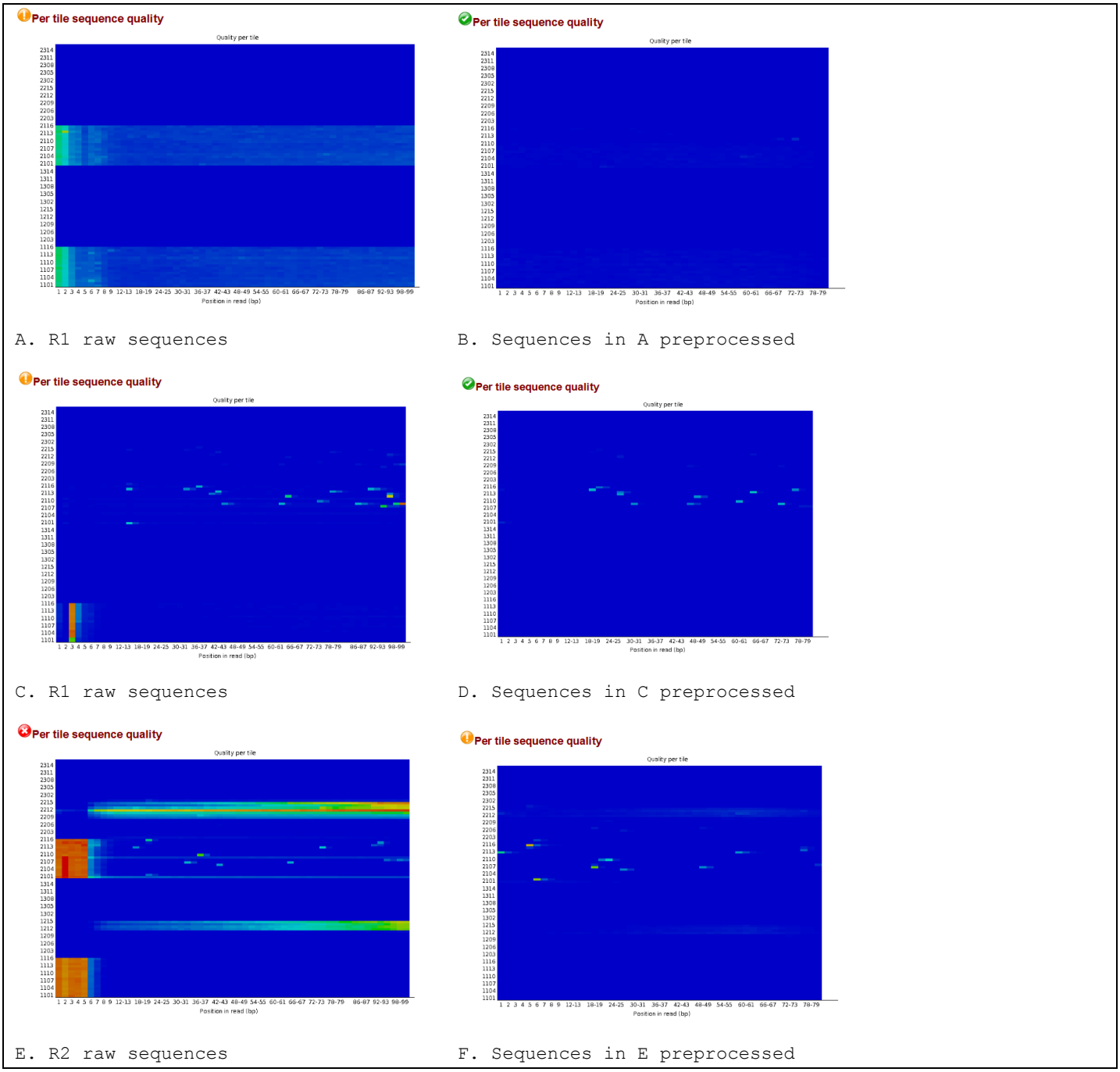
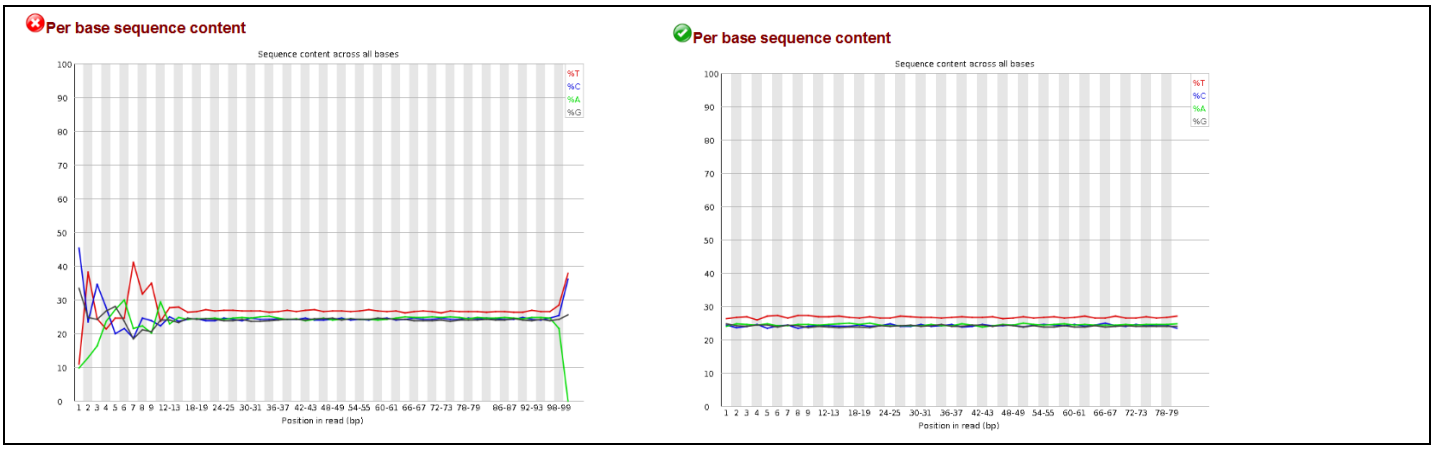


Figure 6. The FastQC diagrams show the sequence quality per tiles for each position of the reads. Blue color indicates good sequence quality, while other colors indicates a decline in quality. For example, red color can indicate an air bubble being present on the chip (ref manual).

Figure 3A represents a typical sample in the data set, and B is the same sample after preprocessing. C – F represents another sample, with the lowest per tile sequence quality in the data set. C and E are the raw sequences, R1 is forward and R2 is reverse sequence, and D and F are the same samples as in C and E after preprocessing, respectively.

All raw sequences had poor distribution in the first approximately 15 nucleotides from the 3'end. Seven of the samples had poor distribution in the last four nucleotides from the 5'end. This problem was identified in the "Per base sequence content" diagrams in the FastQC reports, as shown in Figure 4. The "Per base N content" diagrams in the same reports revealed that the poor distribution at the 5'end was due to base ambiguity. After the preprocessing, all diagrams showed good quality.



A. Raw sequences B. Preprocessed sequences

Figure 7. The FastQC diagrams show the distribution of the four nucleotides over the positions in the sequences. Figure 4A shows the raw sequences from a typical sample, while B shows the preprocessed reads from the same sample as in A.

Overrepresented sequences

19 FastQC reports from the raw sequence files showed overrepresented sequences, described as sequences representing more than 0.1% of the total number in the sample. There were 12 unique overrepresented sequences, and the overrepresented sequences in the raw data are presented in Table 3.

Five of the sequences were TruSeq adapters (ref D701-712 adapter in 'Illumina Adapter Sequences Document # 10000000000002694 v00), and one was possibly Illumina PCR primer. The remaining six sequences had no hits in the FastQC reports. From NCIB-BLAST results, three of these had hits to rRNA (two from salmon and one orthologue), and three had hits to coding genes in salmon. For further details about the BLAST hits, see Table 4.

The trimmed and filtered sequences were also controlled for overrepresented sequences. In the paired reads, there were 16 unique overrepresented sequences, as presented in Table 5. In the unpaired reads, there were 105 unique overrepresented sequences. NCBI-BLAST hits showed ribosomal RNA in most of these sequences.

Selection of quality assured reads for downstream analysis

The 226,987,777 preprocessed, paired reads (see table…) were used for any further analysis.

Table 2. Overrepresented sequences (> 0.1% of the total number of sequences in sample) in the raw data.

No	*Sequence*	*Possible source*
1	CCGACATCGAAGGATCAAAAAGCGACGTCGCTATGAACGCTTGGCCGCCA	No Hit
2	CCTCACCCGGCCCGGACACGGAAAGGATTGACAGATTGATAGCTCTTTCT	No Hit
3	CCTGCCAGTAGCATATGCTTGTCTCAAAGATTAAGCCATGCAAGTCTAAG	No Hit
4	CGAGAGTAAAGGTTACCTGCTTCAACAGTGCTTGAACGGCAACCTTCTAC	No Hit
5	CTCACCCGCTCCTAAAAATTGCTAATGACGCACTAGTCGATCTCCCAGCA	No Hit
6	CTCACAACTAGGATTCCAAGACGCGGCCTCCCCTGTAATAGAAGAACTCC	No Hit
7	GATCGGAAGAGCACACGTCTGAACTCCAGTCACACAGTGATCTCGTATGC	TruSeq Adapter, Index 5
8	GATCGGAAGAGCACACGTCTGAACTCCAGTCACATGTCAGAATCTCGTAT	TruSeq Adapter, Index 15
9	GATCGGAAGAGCACACGTCTGAACTCCAGTCACCGATGTATCTCGTATGC	TruSeq Adapter, Index 2
10	GATCGGAAGAGCACACGTCTGAACTCCAGTCACGGCTACATCTCGTATGC	TruSeq Adapter, Index 11
11	GATCGGAAGAGCACACGTCTGAACTCCAGTCACTGACCAATCTCGTATGC	TruSeq Adapter, Index 4
12	GATCGGAAGAGCGTCGTGTAGGGAAAGAGTGTAGATCTCGGTGGTCGCCG	Illumina Single End PCR Primer 1

Table 3. NCBI-BLAST hits of overrepresented sequences (> 0.1 % of the total number of sequences in the sample) in the raw data. The sequence numbers represents the sequences in Table 3.

No	*NCBI-BLAST hits*	*Species*	*Total Score*	*Query Cover*	*E-value*
1	rRNA, 18S	Bandit angelfish	93.5	100 %	2e -16
2	rRNA, 18S	Atlantic salmon	99,6	100 %	1e -20
3	rRNA, 18S	Atlantic salmon	97,6	97,60 %	5e -20
4	Ferritin, heavy subunit	Atlantic salmon	97,6	100 %	1e -20
5	COX2 gene	Atlantic salmon	99,6	100 %	1e -20
6	Cytochrome b	Atlantic salmon	99,6	100 %	1e -20

Table 4. NCBI-BLAST hits in overrepresented sequences (> 0.1 % of total number of sequences in the sample) in preprocessed paired sequences.

NCBI-BLAST hits	*Species*	*Number of unique sequences*
Actin	Atlantic salmon	2
Ferritin, heavy subunit	Atlantic salmon	3
Cytochrome oxidase subunit II	Atlantic salmon	3
Cytochrome b	Sea trout	1
Zink finger protein pseudogene	Rainbow trout	4
rRNA, 18S	Atlantic salmon	1
rRNA, 18S	Lenok	1
rRNA, 18S	Bandit angelfish	1

3.2. De novo assembly

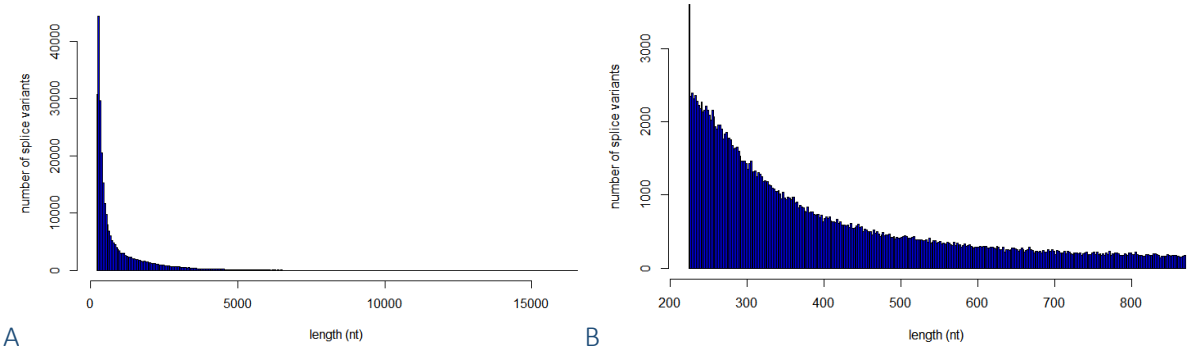
In silico Normalization

Prior to running de novo assembly, Trinity's In silico normalization was performed. This was done in order to validate and reduce the number of reads. The total number of reads post normalization was 22,179,600, which is 9.77 % of the original processed paired reads.

De novo assembly

The total number of splice variants produced in de novo assembly was 279,969. The number of genes defined by Trinity was 195,236. The length of the splice variants were in the range of 224 – 16,549 bases, as illustrated in Figures 8 and 11. The mean length was 870 bases and the median was 449 bases. The GC content was 46.7 %, and the total number of nucleotide was approximately 243 million.

Size distribution of splice variants in de novo assembly



A B

Figure 8. The diagrams shows the legnth distribution of the splice variants produced in de novo assembly. Figure 4A shows the distribution of all splice variants, while B shows the splice variants shorter than the mean product size.

3.3. Read mapping

Both genome-guided assembly (gga) and transcriptome-assembly (tga) was performed, using the 226,987,777 paired reads, which were selected for downstream analysis. The reads were mapped to the RefSeq genome and to the splice variants. The results of the read assembly is presented in Table 5.

Table 5: Mapped and unmapped reads in the assembly to the salmon RefSeq genome and the transcripts from de novo assembly.

	N, mapped reads	N, Unmapped reads	Mapping rate
Genome-guided assembly (gga)	219,116,097	7,871,680	96.5 %
Transcriptome-guided assembly (tga)	205,048,572	21,939,205	90.3 %

3.4. Filtering of splice variants

The read depth information gained in the transcriptome-guided assembly (see last section) was used for filtering out low expressed splice variants. The cut off level for expression was set to fpkm > 1 in at least two samples. After filtering, the total number of splice variants went down to 110,666, and number of 'Trinity-genes' down to 54,507. More statistics on the filtered dataset is located in Table 6, and the distribution of the product size is illustrated in Figures 9 and 11.

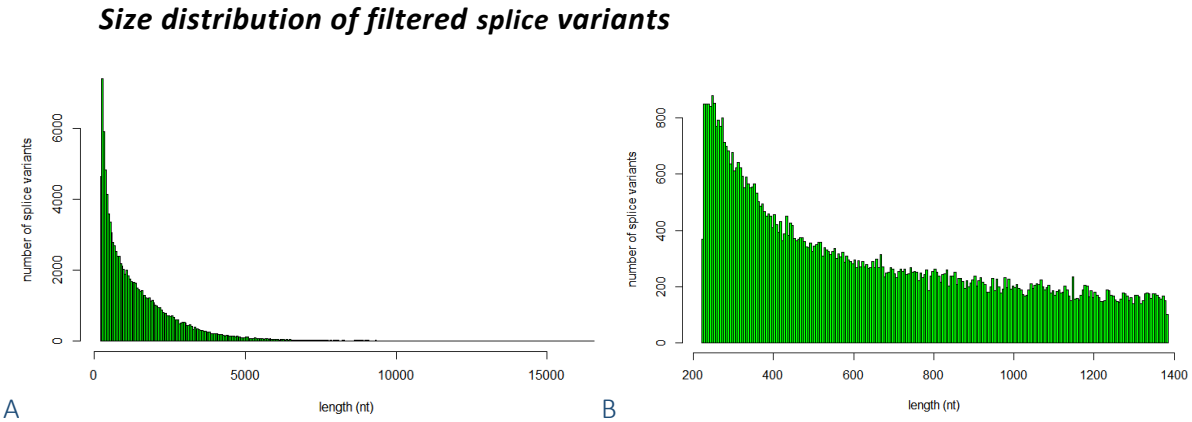


Figure 9. The diagrams shows the size distribution of the filtered splice variants. Figure 5A shows the distribution of all filtered splice variants, while B shows the filtered splice variants shorter than the mean size in the filtered dataset.

Clustering

The expressed spliced variants were clustered, based on a 99% similarity threshold. Post clustering, there were 102,333 splice variants, grouped into 51,056 genes ('Trinity-genes'). This dataset was used for the BLAST assemblies. Mores statistics is located in Table 6, and the distribution of the product size is illustrated in Figures 10 and 11

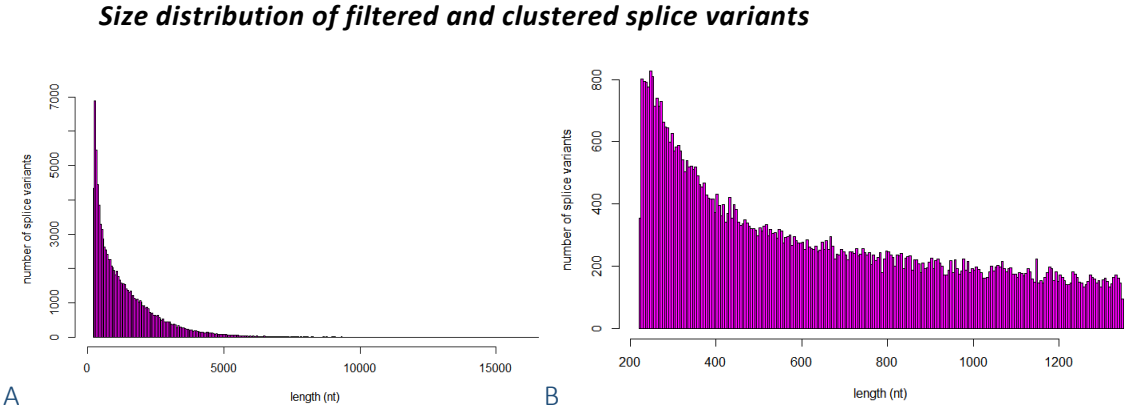


Figure 10. The diagrams shows the size distribution of the filtered and clustered splice variants. Figure 6A shows the distribution of all the splice variants in the clustered dataset, while B shows the splice variants shorter than the mean size product in the dataset.

Table 6. Statistical measures of the splice variants produced in de novo assembly. Denovo represents the unfiltered dataset produced by Trinity. The filtered dataset involves only the splice variants with an expression of > 1 fpkm in two samples or more. The clustered dataset was reduced by clustering after the filtering was done.

	Denovo	Filtered	Clustered
Splice variants	279696	110666	102333
Genes by Trinity	195236	54507	51056
% of de novo	100	40	37
Mean size	870	1384	1349
Median size	449	986	969
Minimum size	224	224	224
Maximum size	16549	16549	16549
% GC	47	47	47
Nucleotides	243368417	153141160	138029831

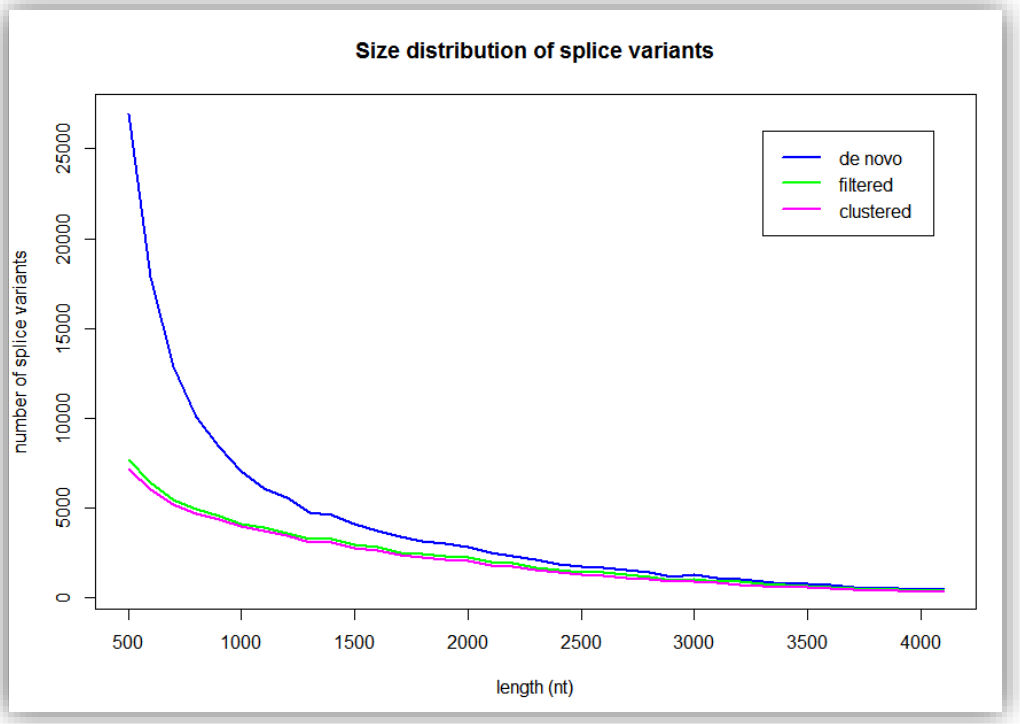


Figure 11. Size distribution of the splice variants up to 4000 nucleotides in the unfiltered (de novo), filtered and clustered datasets.

3.5. BLAST

The dataset containing 102,333 splice variants and 51,056 'Trinity-genes', as described in the previous section under 'Clustering' in Ch. 3.4, was used as query for the BLAST search.

1) This dataset was aligned to the RefSeq genome, using the Megablast algorithm.

   The result was 99,893,169 hits.

   101,974 splice variants and 50,882 'Trinity-genes' were expressed.

   359 splice variants and 174 'Trinity-genes' were not aligned to the salmon reference.

2) The same dataset was also aligned to the RefSeq protein database, using the blastx algorithm. The RefSeq protein database has 97,555 genes.

   The result was 7,734,116 hits.

   64,727 splice variants and 23,183 'Trinity-genes' were expressed.

   37,249 splice variants and 27,669 'Trinity-genes' were expressed in the genome alignment but not in the protein alignment.

To get a better picture of these numbers, the results are given in Table 7.

Table 7: Total number of BLAST hits from alignment of splice variants and 'Trinity-genes' to the salmon RefSeq genome and the protein database.

	N, hits	N, splice variants	N, 'Trinity-genes'
Query data (filtered splice variants)		102,333	51,056
Total hits to the genome	99,893,169	101,974	50,882
No hits to the genome		359	174
Total hits to the proteins	7,734,116	64,727	23,183
Hits to the genome but not to the protein db.		37,249	27,669

3.6. Genome-guided assembly

The RefSeq.gff annotation file had 81,586 genes and 136,264 isoforms.

50,267 genes and 74,860 isoforms were expressed at an expression level of > 1 fpkm in at least two samples.

4. Discussion

4.1. Preprocessing

Most part of the raw sequences had high quality score, as shown in Figure 5. Still, there were some improvements to be done, like getting rid of poor sequences at the ends. By cropping off at both ends of the reads, the whole sequences seemed to be of highly good quality, see Figures 5 and 7.

The 'Per tile sequence quality' diagrams, as shown in Figure 6, showed varying sequence quality, from good to poor, in the raw sequences. The quality was quite consistent for forward and reverse sequences, and for samples within the same project. After preprocessing, the quality improved, from poor (red flag) to acceptable (yellow flag) or from acceptable to good (green flag).

The decrease in sequence length was acceptable, as it was more important to get rid of the poor sequence ends to avoid poor alignments. Distribution of read length post trimming showed that most sequences were at a maximum, and the minimum threshold could have been set even higher, though this was probably not so important, as very few were at a minimum of 70 bases.

BLAST results of overrepresented sequences showed contamination of Illumina adapter and primer sequences. These were efficiently removed in the preprocessing step. Ribosomal RNA was also a source of contamination, and the most abundant rRNA sequences were added in the ILLUMINACLIP FASTA file and filtered in the preprocessing. Trimmomatic outputs paired and unpaired sequences in separate files. By quality checking these files, it appeared that the unpaired sequences had significant amounts of rRNA, these were therefore left out of downstream analysis. The GC content in paired versus unpaired sequences supported the decision to leave these sequences out from the study. When doing BLAST in the overrepresented sequences, some orthologues from related species were discovered, see Table 4.

4.2. De novo assembly

Normalization of reads prior to the de novo assembly had big effect on the number of reads, and seemed to be very important due to data being very large and difficult to handle. The normalization removes all reads that are on top of the 20 read depth threshold (http://ivory.idyll.org/blog/trinity-in-silico-normalize.html). At this threshold, the number of reads should be sufficient for doing a good assembly.

The de novo assembly output a good number of splice variants. The size distribution showed a large number of short reads, i.e. below 300 bases. Longer reads require larger read depth to guarantee the overlap (Li et al., 2010), therefore the short reads are likely to be low expressed.

Clustering was done mainly to reduce the amount of data. Setting the threshold is a tradeoff between too much data and possibly loosing important data. Because of the high level of duplicated sequences in salmon, the similarity threshold was set quite high, to 99 %.

Appendix B shows that the main reduction of short reads was done in the filtering, while reduction of the long reads was done in the clustering. This makes sense, as short transcripts are expected to have low expression, and long transcripts are expected to be highly expressed genes.

4.3. Mapping rate

The mapping rate of reads to the genome was higher than the mapping rate of reads to the transcripts. Mapping to the genome is less demanding than doing de novo assembly regarding complexity of the algorithms. In Trinity, the transcripts are dependent on a minimum coverage to be assembled, but in mapping to the genome requires less read depth. More unmapped reads in the de novo assembly is therefore expected.

4.4. Expression of splice variants

Most splice variants and 'Trinity-genes' from the de novo assembly dataset were aligned to the genome sequence. This indicates good quality of the transcripts, as there is a good match to the reference of the species. Still, a few 'Trinity-genes' (174) and splice variants (359) did not get hits, and these should be further investigated, e.g. by doing BLAST on the non-redundant database.

A large number of splice variants were expressed in the genome but not to the annotated genes. It is not likely to believe all these are coding genes, and it is expected that some of the transcripts are long non-coding RNA. Still, it is reason to believe that some are new discovered genes. Considered our samples are from different life stages and treatment, there is most probably some genes expressed that are not normally expressed.

4.5. Comparison of the assemblies

There were about twice as many expressed genes in the genome-guided assembly (50,267) than in the BLAST of the splice variants to the protein database (23,183). The number is quite significantly higher, and if we look at ratio of expressed genes in the database, the difference is even higher; 24% of the proteins in the protein database are expressed and 62 % of the proteins in the annotation file used in genome-guided assembly are expressed. This supports the results from the mapping rate comparison, where genome-guided assembly had a higher mapping rate than the mapping of reads to the transcripts.

The mapping in genome-guided assembly is of a highly acceptable level, and can here be recommended as a method for transcriptome analysis if new discoveries are not the main issue, but rather look at expression of annotated genes.

4.6. Future studies

This section mentions things that could have been explored in the data if time had not been a limitation. The 174 'Trinity-genes' and 359 splice variants that were not aligned to the salmon reference are obviously expressed sequences, but not identified in the salmon genome. These are possible orthologues, i.e. new discovered genes or isoforms in salmon, derived from other species. A BLAST search to the non-redundant database, which contains sequences from a number of species, could give an answer to this question.

Duplication has not been considered in this study. The fact that salmon is a highly duplicated species (Lien et al., 2016) gives uncertainty to the results, especially regarding the number of hits and findings in the BLAST searches, as one query sequence could easily match several loci. Reciprocal best hits (RBH) is when two genes codes for one protein. Reciprocal BLAST can be used to find this information (Ward and Moreno-Hagelsieb, 2014).

4.7. Conclusion

By using de novo assembly, we can extended the gene annotation of Atlantic salmon with up to 27,000 genes. There are many uncertainties to the exact number, but the extension is significant when analyzing for treated samples. De novo assembly is recommended for extended results in transcriptome analysis.

Computer programs and analysis platforms used in the thesis:

1) Illumina http://www.illumina.com/
2) GNU nano http://www.nano-editor.org/download.php
3) R https://cran.r-project.org/
4) RStudio https://www.rstudio.com/
5) EdgeR https://bioconductor.org/packages/release/bioc/html/edgeR.html
6) FastQC http://www.bioinformatics.babraham.ac.uk/projects/fastqc/
7) Trimmomatic http://www.usadellab.org/cms/?page=trimmomatic
8) Trinity https://github.com/trinityrnaseq/trinityrnaseq/wiki
9) Bowtie http://bowtie-bio.sourceforge.net/index.shtml
10) TopHat https://ccb.jhu.edu/software/tophat/index.shtml
11) Bedtools http://bedtools.readthedocs.io/en/latest/index.html
12) CD-HIT-EST http://weizhongli-lab.org/cd-hit/
13) BLAST https://blast.ncbi.nlm.nih.gov/Blast.cgi
14) Cufflinks http://cole-trapnell-lab.github.io/cufflinks/

Literature

Altschul, S., Warren Gish, Miller, W., Myers, E.W., and Lipman, D. (1990). Basic Local Alignment Search Tool. J. Mol. Biol. *3*, 403–410.

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

Davidson, W.S., Koop, B.F., Jones, S.J.M., Iturra, P., Vidal, R., Maass, A., Jonassen, I., Lien, S., and Omholt, S.W. (2010). Sequencing the genome of the Atlantic salmon (Salmo salar). GenomeBiology *11*.

Haas, B.J., Papanicolaou, A., Yassour, M., Grabherr, M., Blood, P.D., Bowden, J., Couger, M.B., Eccles, D., Li, B., Lieber, M., et al. (2013). De novo transcript sequence reconstruction from RNA-seq using the Trinity platform for reference generation and analysis. Nat. Protoc. *8*, 1494–1512.

Korpelainen, E., Tuimala, J., Somervuo, P., Huss, M., and Wong, G. (2015). RNA-seq Data Analysis - A Practical Approach (Chapman & Hall/CRC).

Li, W., and Godzik, A. (2006). Cd-hit: a fast program for clustering and comparing large sets of protein or nucleotide sequences. Bioinformatics *22*, 1658–1659.

Li, R., Zhu, H., Ruan, J., Qian, W., Fang, X., Shi, Z., Li, Y., Li, S., Shan, G., Kristiansen, K., et al. (2010). De novo assembly of human genomes with massively parallel short read sequencing. Genome Res. *20*, 265–272.

Lien, S., Koop, B.F., Sandve, S.R., Miller, J.R., Kent, M.P., Nome, T., Hvidsten, T.R., Leong, J.S., Minkley, D.R., Zimin, A., et al. (2016). The Atlantic salmon genome provides insights into rediploidization. Nature.

Trapnell, C., Roberts, A., Goff, L., Pertea, G., Kim, D., Kelley, D.R., Pimentel, H., Salzberg, S.L., Rinn, J.L., and Pachter, L. (2012). Differential gene and transcript expression analysis of RNA-seq experiments with TopHat and Cufflinks. Nat. Protoc. *7*, 562–578.

Ward, N., and Moreno-Hagelsieb, G. (2014). Quickly Finding Orthologs as Reciprocal Best Hits with BLAT, LAST, and UBLAST: How Much Do We Miss? PLoS ONE *9*, e101850.

Appendix A

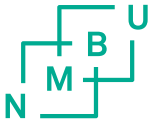
Illumina adapter and primer sequences, used in the ILLUMINACLIP-function in Trimmomatic.

```
>PrefixPE/1
TACACTCTTTCCCTACACGACGCTCTTCCGATCT
>PrefixPE/2
GTGACTGGAGTTCAGACGTGTGCTCTTCCGATCT
>PE1
TACACTCTTTCCCTACACGACGCTCTTCCGATCT
>PE1_rc
AGATCGGAAGAGCGTCGTGTAGGGAAAGAGTGTA
>PE2
GTGACTGGAGTTCAGACGTGTGCTCTTCCGATCT
>PE2_rc
AGATCGGAAGAGCACACGTCTGAACTCCAGTCAC
>SE_PCRprimer
GATCGGAAGAGCGTCGTGTAGGGAAAGAGTGTAGATCTCGGTGGTCGCCG
```

Appendix B

Number of splice variants in the unfiltered (denovo), filtered and clustered datasets. The left column is the maximum product size, down to the product size in the previous row.

size -	denovo -	filtered -	clustered
600	169644	36797	34194
1000	38362	19066	18169
1400	21034	13970	13347
1800	14351	10684	10001
2200	10635	8458	7749
2600	7395	6008	5434
3000	5337	4476	3907
3400	3817	3225	2761
3800	2599	2276	1952
4200	1844	1589	1370
4600	1406	1207	1057
5000	973	837	684
5400	667	596	492
5800	476	438	362
6200	304	271	224
6600	227	203	168
7000	176	153	130
7400	137	120	96
7800	81	71	59
8200	44	39	35
8600	29	29	22
9000	56	54	40
9400	36	36	32
9800	8	8	7
10200	14	13	9
10600	4	4	4
11000	6	4	3
11400	2	2	1
11800	16	16	11
12200	3	3	3
12600	2	2	2
13000	2	2	2
13400	3	3	3
13800	0	0	0
14200	0	0	0
14600	0	0	0
15000	2	2	1
15400	0	0	0
15800	0	0	0
16200	1	1	1
16600	3	3	1



Norges miljø- og biovitenskapelig universitet
Noregs miljø- og biovitskapelege universitet
Norwegian University of Life Sciences

Postboks 5003
NO-1432 Ås
Norway