Julius-Maximilians-Universität
Würzburg

Faculty of Biology

Department of Bioinformatics

MASTER THESIS

Extending the transcriptome analysis pipeline READemption

March 21, 2017

Author:

Diarmaid Tobin

Matr.Nr.: 1755928

Supervisors:

1st: Dr. Konrad Förstner

2nd: Prof. Dr. Thomas Dandekar

Abstract

The introduction of Sanger DNA sequencing (DNA-Seq) in the 1970s has greatly contributed to resolve formerly unanswered questions. Since, a number of automated platforms using DNA-Seq have been launched, especially being improved in terms of throughput and cost per base. Due to this and the range of application possibilities, DNA-Seq gains ever-increasing popularity within different scientific areas. By converting RNA to cDNA and then applying DNA-Seq, also sequencing of RNA (RNA-Seq) is possible. Dependend on the biological questioning, different techniques have been established, using RNA-Seq as a core functionality. However, heavy usage of distinct redefined RNA-Seq methods leads to the generation of huge amounts of data. This in turn creates the need for computational methods in order to process and analyse the generated output. There is a broad selection of bioinformatical tools that are designed to make sense of this kind of data. Also a number of pipelines have been established, combining different tools or functionalities with intermediate file-handling, in order to perform consecutive procession steps. READemption represents one of those pipelines and was initially designed for processing data derived from differential RNA-Seq experiments. Its functionality has been extended since, offering read procession and alignment, nucleotide specific coverage calculation, read per feature quantification as well as differential gene expression analysis. In the course of this thesis, READemption has been further extended and improved with respect to different aspects. Hereby the level of automation regarding input files has been increased. The output spectrum has been increased by providing more information in form of plain data and additional visualizations. In addition to publication-quality figures, interactive visualizations are generated with bokeh, enabling a sophisticated data exploration. Alternatively to a self implemented solution, the tool cutadapt was integrated, offerering a greater choice of quality trimming and adapter clipping options of raw reads. As a further option for read alignment, the tool STAR was implemented. This measure increases runtime and reduces memory footprint compared to the already integrated read aligner segemehl. The usage of pytest reduces boilerplate code and improves error reporting of the software testing. Code efficiency could have been increased by using the pandas library for data extraction. Furthermore, a toolkit functionality has been integrated, which allows to perform some of READemption's funtions in a standalone manner. The central controller module of the pipeline has been split into five, which increases the overall readability. Consequently it simplifies the integration of further tools or functionalities in the future, whereas considerations regarding further extensions of READemption are also included in this thesis.

Contents

List of Figures

1 INTRODUCTION

1.1 DNA sequencing

DNA sequencing describes a process, by which the nucleotide sequence of DNA molecules is determined. Although resolving the three-dimensional structure of DNA in 1953 significantly contributed to the understanding of DNA replication and encoding of proteins, scientists lacked the ability to sequence DNA or RNA. It took more than 20 years and several different approaches until Maxam-Gilbert sequencing became the first widely adopted method for the analysis of DNA sequences [1]. For this technique radiolabelled DNA fragments are aliquoted into four pools, each getting treated with a chemical specifically removing one base from the fragment. After gelectrophoresis in polyacrylamide gel the band sizes are compared whereby the nucleotide sequence can be concluded [2]. In 1977, Sanger's 'plus and minus' method led to the first fully sequenced genome of the bacteriophage ϕX174 with $\sim$ 5000 bases [3, 4]. By combining elements of Maxam-Gilbert sequencing and the 'plus and minus' method, Sanger introduced in the same year the dideoxy chain-termination method, which is also referred to as Sanger sequencing [5]. This method became the gold standard for years to come and was gradually improved by incorporating new technologies such as shotgun DNA sequencing, fluorescent labeling or capillary based electrophoresis [6, 7, 8]. Subsequently, this led to the introduction of the first commercial DNA sequencing platform [9]. Driven by those advantages and initiatives like the 'Human Genome Project', the human genome with $\sim$ 3 billion bases was sequenced in 2001 using Sanger sequencing [10]. Although other approaches existed, nowadays Sanger sequencing is put at a par with the term 'first-generation sequencing' due to its domination of the field of DNA sequencing for nearly 30 years. Nevertheless the method has limitations, especially in terms of throughput [11]. This resulted in the introduction of other commercially available high-throughput DNA sequencing platforms between 2005 and 2010, commonly referred to as 'second-generation sequencing' systems. Compared to the previous capillary-based methods, those systems enable massively parallel sequencing of fragment libraries, which also drops sequencing costs [12]. A disadvantage of those systems are rather short read sequences obtained, which led to the development of paired-end sequencing [13]. In 2010, further systems were introduced. Those especially stand out with dramatically increased read lengths and are subsequently termed as 'third-generation sequencing' platforms. Longer read sequences improve the quality of downstream analysis, such as de novo genome assemblies [14]. Speaking of both, second and third generation, the expressions next-generation sequencing (NGS) or high-throuput sequencing (HTS) are commonly used. In general, DNA sequencing technologies differ in terms of output read length, runtime, throughput, cost per sequenced base and error rate which should be taken into consideration for choosing a platform [15]. In the following, the modes of operation of the most relevant DNA sequencing platforms are described.

1.1.1 First generation sequencing

Sanger sequencing. Since the early 1990s and after years of improvement, the Sanger sequencing method has commonly been performed in a capillary-based and semi-automated manner to achieve preferably high throughput. As the biochemical principle involves DNA polymerase activity, it is considered as a sequencing-by-synthesis (SBS) method [16]. DNA templates for the sequencing process may be either gathered by a shotgun de novo sequencing approach, or using PCR amplification for targeted sequencing. The templates are aliquoted into four different reaction approaches and mixed with primers, DNA polymerase and the four types of deoxynucleosidetriphosphates (dNTPs). In each approach one kind of fluorescent labeled 2',3' dideoxynucleotides (ddNTPs) is added. ddNTPs are synthetic analogues of the natural dNTPs lacking the 3'-OH which is required to form phosphodiester bonds between two nucleotides. Once a DNA polymerase adds a ddNTP instead of dNTP into an growing nucleotide chain, a so called 'chain termination' occurs as no further dNTP can be incorporated [17, 18]. The sample preparation is followed by a 'cycle sequencing' reaction, consisting of several cycles of template denaturation, primer annealing and primer extension. The concentration of ddNTPs being $\sim$ 100 fold lower than the one of dNTPs ensures chain termination at every possible position of the newly synthesized strands. In capillary-based polymer gels the extension products get separated using high-resolution electrophoresis. Once the DNA fragments exit the capillary ordered by size, laser excitation of the ddNTP specific fluorescent labels enables the identification of the last base. Based on the signals, a software can determine the sequence of the whole template. Such a determined sequence is generally referred to as 'read'. While detecting fluorescent signals, the software additionally determines the error probability per base-call. This kind of improved approach enables read-lengths of up to $\sim$ 1000 bp and an accuracy of 99.999 percent [19].

1.1.2 Second generation sequencing

454 Pyrosequencing. In 2005, 454 Life Sciences introduced the first commercial available NGS platform that combines pyrosequencing with emulsion PCR [20, 21, 22]. In 2007, Roche Applied Science acquired 454 Life sciences and launched its own version. It utilizes the same core biochemistry, but the flow cell with is replaced by a 'picotiter well' plate made up from a fused fiber-optic bundle. For this system the library of template DNA is fragmented by either nebulization or sonication.

The DNA fragments are denatured to single-stranded DNA (ssDNA) and ligated with a specific adapter capable of binding to a bead, which results in beads carrying one fragment each. For fragment amplification emulsion PCR is carried out, whereas beads and PCR reagents are located within water droplets immersed in oil. This amplification is important for a sufficient light signal in the following SBS procedure by pyrosequencing and results in beads with one kind of amplified ssDNA fragment each. Together with all required reagents, the individual beads are deposited into picoliter plate wells. For the pyrosequencing template DNA, DNA polymerase, ATP sulfurylase, luciferin, luciferase and apyrase is required. The four dNTPs for the sequencing reaction are added successively. The incorporation of a dNTP polymerase into the nucleotide chain by the polymerase causes the release of pyrophosphate. ATP sulfurylase converts pyrophosphate to ATP which acts as a catalyst for luciferase-mediated conversion of luciferin to oxyluciferin. This conversion generates light, whereas the light intensity depends on how many individual dNTPs got incorporated in a row. Apyrase degrades dNTPs prior to the addition of another kind of dNTP [23]. The light signal is transmitted through the fiber-optic plate and captured by a specific camera system. A software determines quality criteria per base and creates the output read. Within one run 1 million reads with a read length of up to 1000 bp can be generated. Biggest concern of the 454 system is the lacking accuracy of detecting homopolymers greater than four bases [24].

Illumina bridge amplification. In 2006, Solexa launched its SBS based sequencing system. In 2007, the company was acquired by Illumina, whereas the technique is mostly referred to as 'Illumina sequencing'. The preparation step includes the library creation of DNA fragments of the desired size. Both ends of the fragments are attached with adapters, indices and with regions, that are complementary to oligonucleotides binding on a flow cell surface. Indices, also known as barcodes, are library specific and allow the sequencing of more than one DNA library at the same time, which is called multiplexing. Adapters may be used as primer binding sites or used for paired end approaches. After binding of the fragments to the oligonucleotides attached to the flow cell, bridge amplification is performed to create distinguishable clusters of the same fragment [25, 26]. Those clusters of ~1000 copies are important to generate a detectable light signal in the following sequencing step. The sequencing reaction requires a DNA polymerase, primers, and four reversible terminator nucleotides, labeled with an individual dye each [27]. The incorporation of a nucleotide is detected by a fluorescent signal, whereas the dye identifies the nucleotide and the terminator prevents the addition of further ones. The dye and terminator at the 3'-terminus then are removed, followed by as many sequencing cycles until the full read can be obtained [28]. Illumina offers a big range of sequencing platforms and is currently laeader within the HTS sector. Its systems offer a high throughput and very low per base cost with read lengths of up to 300 bp [29].

SOLiD ligase-mediated sequencing. In 2007, the SOLiD (Sequencing by Oligonucleotide Ligation and Detection) system was launched by Applied Biosystems, though Thermo Fisher has meanwhile acquired the technique. Preparing the library involves DNA fragmentation via nebulization, sonication or enzymatically digestion. Every fragment is attached with an universal adapter termed as P1. For the following emulsion PCR step every fragment is attached to one bead bearing sequences complementary to the P1 adapter on its surface. This can be granted through a much higher concentration of beads than fragments. After the PCR step every bead contains millions of copies of the same template DNA, whereas beads lacking fragments are removed. Next, the beads are attached covalently to a glass slide, where a ligase-mediated SBS process takes place that requires a primer complementary to the P1 adapter, ligase, a cleavage agent and probes made up of 8 nucleotides. The first two nucleotides of those probes are the ones being actually sequenced, the last six nucleotides are universally binding and reveal a cleavage site between the fifth and sixth nucleotide. The eighth nucleotide is attached with one out of four fluorescent labels that corresponds to each dinucleotide permutation of the first two bases. This provides 16 possible dinucleotide permuatations in total. Once primer binding took place, the ligase incorporates the matching probe and excitation of the fluorescent label, including detection of the signal, takes place. Then the probe gets cleaved, leaving a free end for another cycle of ligation, whereas the cycles are repeated until the template is covered. After the first round of ligation cycles another four are performed, in each of them the primer being offset by one base. By comparing the results of all five rounds, the sequence of the template can be concluded resulting in reads of up to $\sim$ 75 bp. As every base being read twice, the accuracy of SOLiD systems is claimed to be very high [30, 31].

Ion Torrent. Ion Torrent's semiconductor sequencing technology was made commercially available in 2010 by Life Technologies. Similar to the 454 pyrosequencing technology library preparation takes place, followed by emulsion PCR resulting in beads carrying amplified DNA fragments including specific adapters. The beads are placed into microwells within microwell plates located upon a semiconductor chip, where a SBS reaction takes place. The microwells successively get added one kind of dNTP which then may be incorporated by a DNA polymerase. Does the incorporation occur, a positively hydrogen ion is released by the reaction which causes a pH change within the microwell. The pH change can be measured and transformed into a voltage signal by the chip beneath the wells. Hereby the voltage signal increases proportional to incorporated dNTPs. As this correlation does not scale perfectly, homopolymers greater than six bases decrease the accuracy. Depending on the platform used, reads up to a length of $\sim$ 400 bp can be obtained [32, 33].

1.1.3 Third generation sequencing

Single molecule real time (SMRT) sequencing. In 2010, Pacific Biosciences released a SMRT sequencing platform that does not require a clonal amplification step of templates [34]. It utilizes zero-mode waveguides (ZMWs) enabling single molecule real time sequencing. ZMWs are 50 nm wide and 100 nm long aluminium wells that are attached on a silica cover slip. Below the silica layer light illumination takes place, whereas the width causes a light attenuation at 25 nm [35]. For the sequencing reaction, a DNA polymerase together with the DNA template are immobilized at the bottom of the ZMW. The dNTPs being used reveal fluorescent phospholinked labels, whereas an individual dye is used for one of the four dNTP types. Due to the light attenuation, only fluorophores being incorporated by the DNA polymerase are excited, which enables the detection of a specific nucleotide incorporation in real time. As the incorporation leads to cleavage of the phosphate chain carrying the fluorescent dye, the fluorophore is removed and the process repeats until the whole read is obtained [36]. This technique allows reads greater than 14.000 bp, whereas $\sim$ 50.000 reads can be obtained in a four hour run. Nevertheless, error rates may leave space for improvement [37, 38].

Nanopore sequencing. In 2014, Oxford Nanopore Technologies launched the portable MinION device which performs real-time sequencing using nanopore technology [39]. The nanopores are located within a synthetic membrane on which a potential is applied, causing a current flow through the nanopore apperature. A DNA polymerase attached to the DNA molecule to be sequenced is placed on top of the pore. The applied electric field causes one strand of the DNA moving through the pore. Passing a specific spot within the pore, a single nucleotide causes a characteristic disruption of the current, which allows identifying the nucleotide composition of the whole strand in real-time [40]. Adding hairpin structures to the end of the double stranded DNA allows the sequencing of the forward and reverse strand in one go, an amplification of the DNA is not necessary, though. Within 18 hours the device is capable of producing up to $\sim$ 16.000 reads with $\sim$ 60.000 bases in length. Although nanopore sequencing represents a very promising approach, there is not much data published actually evaluating its performance [41, 42].

1.2 RNA sequencing

The quantity and type of transcripts within a cell at a given time is called transcriptome, whereas its characterization is referred to as transcriptomics. Since the 1990s, transcriptomics have been mainly performed using microarray hybridization methods, leading to important advances in this field. However, the technology reveals several limitations, such as lacking accuracy of expression levels or the limited choice of transcription products that may be analysed. Many of those limitations have been overcome by the application of RNA sequencing (RNA-Seq), whereby RNAs are converted to complementary DNAs (cDNAs) and then sequenced using HTS technologies [43, 44, 45]. RNA-Seq has furthermore increased the understanding of complex post-transcriptional processes by realising the significance of different kinds of non-coding RNA species or proteins [46, 47, 48]. Different transcriptome related questionings have led to the development of redefined methods using RNA-Seq, whereas most of them deal with different expression levels of certain genes [49, 50]. In accordance with the experimental design and HTS technology being used, also the library preparation differs. Usually the first step is depletion of ribosomal RNA (rRNA) within the RNA pool to be sequenced. As eukaryotic mRNAs and some of its non-coding RNAs are polyadenylated, they may be extracted using oligo-dT beads. Alternatively and for prokaryotic transcriptomes, RNAs carrying 5' monophosphates can be degraded by exonucleases which depletes rRNAs. The next steps typically include fragmentation of the RNAs and conversion to cDNA, but the order and technical execution differ among the protocols used. Using barcode tags further enables multiplexing different cDNA libraries [51]. Most commonly, the result of the preparation are libraries of adapter- and barcode-ligated cDNAs [52]. Some promising RNA-Seq based methods that are used in transcriptomics today are described in the following sections.

1.2.1 Differential RNA-Seq

Originally designed to describe the primary transcriptome of *Helicobacter pylori*, the differential RNA-Seq (dRNA-Seq) technique was introduced in 2010. Primary transcripts reveal a 5' tri-phosphate group (5'PPP), whereas processed transcripts carry a 5' monophosphate (5'P) or in some cases a 5' hydroxyl (5'OH) group. For a dRNA-Seq approach the RNA sample is split into two aliquots, both containing primary and processed transcripts. One aliquot then gets treated with 5'P-dependent terminator exonuclease (TEX), which degrades RNAs carrying a 5'P group, consequently enriching primary transcripts. Both aliquots are then converted into a cDNA library and sequenced. By quantitative comparison of the sequenced libraries, either transcription starting sites (TSS) or procession starting sites can be detected [53].

This rather novel technique is a promising approach which already led to new insights into the transcriptomes of pathogens like *H. pylori* or *Vibrio cholerae*, but also other bacteria like *Haloferax volcanii.* Thereby dRNA-Seq has been majorly used to create TSS maps within bacterial genomes, which contributed to the detection and encryption of operon structures [54, 55].

1.2.2 Dual RNA-Seq

The infection of mammalian cells through any kind of pathogen leads to an altered gene expression in both, the host, as well as the infective agent. The transcriptomes of pathogens and host cells during infection have been traditionally studied separately and put in context. However, extracting RNAs from host cell and the pathogen from one sample using methods like microarrays is complicated as a matter of cost and technical limitations. Due to the increased sensitivity and depth, as well as reduced cost, RNA-Seq has shown to be effective for analysing such approaches with the method being called dual RNA sequencing (dual RNA-Seq) [56]. So far, dual RNA-Seq has contributed to the understanding of host-pathogen interaction on the transcriptome level, especially by identifiying virulence-associated small noncoding RNAs (sRNAs) [57].

1.2.3 Cross-linking immunoprecipitation sequencing

Beside various RNA species, also RNA-binding proteins (RBPs) play a major role in the post-transcriptional regulation. Rising interest in RBP-binding motifs and the emergence of HTS technologigies has led to the establishment of new methods. One of them combines high-throughput RNA-sequencing with cross-linking immunoprecipitation (CLIP), inter alia termed as CLIP-Seq [58]. For the CLIP method RBPs and RNAs are covalently linked by UV-irradiation *in situ* and isolated by immunoprecipitation. Then the protein gets removed, only leaving the formerly bound RNA [59]. By applying RNA-Seq on RBP-RNA complexes, including downstream analysis steps, CLIP-Seq enables the identification of RBP-binding sites [60].

1.3 Bioinformatical analysis of RNA-Seq data

Increasing applications and dropping costs of HTS technologies make RNA-Seq a very popular method. The heterogeneity and sheer amount of data generated by RNA-Seq experiments increases proportional to its popularity. There is a huge variety of standalone tools and toolkits addressing different analysis steps for RNA-Seq data, often released under an Open Source license [61]. In order to perform consecutive analysis steps in an automated manner, a pipeline can be created. A pipeline combines several tools or procession steps and manages the data handling of input and output files resulting thereof. The output of one process may serve as an input for the following one, whereby output files often need to be transformed. Transformation of the output may also be self implemented in order to preserve information about a certain procession step. The construction of a pipeline generally involves a range of considerations, including usability, runtime, programming paradigm or language and robustness. Usability reflects how difficult it is to perform tasks with this pipeline, for instance a graphical user interface (GUI) may be easier to handle than a command line executed software application. Factors like the choice of a third-party tool or parallelization of procession steps have a major impact on the runtime. Coding by convention normally results in reduced code and less choices for the user to take. However, this may come to the cost of reduced flexibility of the workflow compared to a configuration-based framework. Another factor to assess a pipeline is its robustness. Single processes of a pipeline are easily interrupted for any technical reason, software bug or data, that do not match expectations. To increase robustness, options to continue the process where it failed, called reentry, may be implemented [62]. A possible way to prevent failures upfront are software tests, which are modules that screen the code upon functionality. Tests are executable scripts that simulate processes of the pipeline and evaluate its functionality, e.g. by running dummy data and check if the result matches the expectation [63]. Although other programming languages are widely used, Python is probably one of the most favorable ones for Biosciences. Python is freely available and has an active scientific community. This has a decisive impact on the development of freely available third-party libraries. Those libraries are often designed rather specifically and therefore quite powerful for a smaller range of tasks, e.g. the effective exploration of huge datatables. Additionally, the Python standard library provides quite a range of of built-in functionalities. Also the syntax is comparable clear, a plus especially for beginners [64]. The actual analysis steps of such a pipeline are dependent on the experimental setup. For this thesis relevant procedures are described in the following sections.

1.3.1 Quality control and adapter clipping of raw reads

Raw reads derived from RNA-Seq experiments are stored within FASTA or FASTQ file format. In general, either nucleotide or amino acid sequences are stored within FASTA files, whereas every sequence reveals a preceding header line withholding an unique identifier of the sequence starting with the character '>' [65]. In order to combine quality scores with every nucleotide of a sequence, the Wellcome Trust Sanger Institute introduced the FASTQ file format. Similar to FASTA, the sequence is represented by a header line containing the identifier that starts with the character '@'. The next line contains the sequence. A third line can optionally contain a description. In the fourth line every nucleotide's quality score is represented by a single ASCII character and derived from the sequencing platform, whereas the phred quality score is the most common. This phred quality score (Q) is defined as an estimated probability of a wrong called base (P) and is expressed by formula 1 [66].

$$Q = -10 \cdot log_{10}(P) \tag{1}$$

Based on the phred score, or similar quality score estimations by other platforms, a quality trimming of the reads can be performed by removing nucleotides below a certain quality score cutoff [67]. If present, also adapters or poly(A) tails are usually removed from the raw reads. Another part of the quality control is the evaluation of GC content levels, which should be homogeneous within samples from the same experiment. Duplicated and too short reads are normally also removed, as they might lead to wrong conclusions in downstream analysis. Several tools exist which perform the quality evaluation of raw read data and perform trimming steps. The aim is to gather reads representing the actual transcriptome and excluding errors, caused by the technical steps of the respective RNA-Seq approach [68].

1.3.2 Read alignment

In theory the reads of a RNA-Seq experiment purely represent RNA sequences found within an extracted transcriptome, but do not withhold any information about its origins or functionalities. Additionally, the most applied HTS technologies create comparable short reads, whereas a single sequenced RNA may be represented by several reads. In order to make sense of those reads, an essential computational step, called mapping or alignment, is required. To perform alignment on RNA-Seq data, a reference genome or reference transcriptome is necessary. There are different strategies to create reference sequences, often they are stored within FASTA files and made publicly available [69, 70]. Task of the alignment process is the correct allocation of reads to the respective reference. This is a computational expensive and quite complicated step, as the read not necessarily represents its reference sequence due to biological and technical reasons. One biological reason that makes the correct alignment of reads difficult, are splicing events. Those events result in junction reads spanning two or more exons, which requires the read being split and the resulting fragments being aligned correctly. A promising approach to identify splicing events are mapping approaches that use a reference transcriptome [71, 72]. Another problematic aspect are reads containing insertions, deletions or mismatches caused by sequencing errors or genomic variations. Several strategies exist to face this issue, e.g. by involving per base quality scores for accurate mapping [73]. In general, the field of aligning RNA-Seq data is under constant development, offering a wide range of alignment tools. Depending on the biological questioning or experimental setup, some tools might be chosen over another, in order to get as many reads as possible accurately aligned to the reference sequence [74, 75, 76]. The alignment result can be stored in Sequence Alignment/Map format (SAM) which supports all sequence modes, i.e. single- and paired-end reads. Information about how often a sequence got matched to the reference, inter alia, is encoded in a CIGAR string. This human-readable format can be further converted to the compressed, non human-readable Binary Alignment/Map (BAM) format, making further analysis steps computational less expensive [77].

1.3.3 Quantitative analysis of aligned read data

One quantitative analysis approach of RNA-Seq data is coverage calculation. Coverage denotes the number of reads covering each base in the reference sequence, an information which can be derived from SAM/BAM files. For further usage, the coverage of chromosome positions are stored in files, often within uncompressed text formats like wiggle [78]. A more specific quantification approach is the gene expression quantification by using an annotation file. Often in Gene Feature Format (GFF3), annotation files contain information about what genomic position is coding for, i.e. which RNA species. By comparing the number of reads at a position from the SAM/BAM files with the annotated position from the GFF3 file, a quantitative assertion about the expression of certain genes can be made. As sequencing depth of libraries usually differ, normalization of the coverages and gene expression quantification data have to be performed in order to compare different libraries. The choice of normalization methods is subject of discussion and also depends on the experimental approach, but inevitable for comparing sequenced libraries [79]. Another, often used technique is the comparison of expression alterations within the same organism under different conditions, called differential gene expression analysis. There is a growing number of algorithms which combine normalization methods and differential expression detection to answer different biological questionings, such as gene expression changes of a pathogens during infection [80].

1.3.4 Visualization

The visualization of large amounts of complex data is inevitable in order to classify this information in a short period of time. It is also helpful for 'sanity checks' of a pipeline's performance or the data being processed. Furthermore it aids the recognition of patterns or relations within biological data, especially when comparing different datasets [81, 82].For example, this is the case with differential gene expression analysis, by which quantitative changes in expression levels of different experimental groups are determined. Hereby the base mean, p-value and fold change serve as a basis for visualization. The base mean describes the expression level of a transcript. The fold change denotes a measure of difference between two expression levels of a transcript from different libraries. The p-value is used as a measure for the fold change being significant and is based on the expression difference and variance of the sample [83]. In order to decrease the risk of Type I errors in this kind of multiple comparison, usually an adjustment of the p-values (padj values) is performed, whereas different adjustment methods may be applied [84]. By plotting the log2 of the fold change (log2 fold change) against the base mean in a scatter plot, one obtains a MA plot. Coloring significant and non-significant data spots individually is a convenient way to get a quick overview of how many transcripts of two compared libraries are expressed significantly. Another way to visualize this kind of data is plotting the log2 fold change against the negative log10 of the padj values, which results in a volcano plot. Hereby the low padj values, which indicate high significance, are plotted on top and the low padj values at the bottom of the coordinate system. Using the log2 fold change instead of the fold change for visualizations has two reasons. Firstly to make the data in the plot appear more dense to a manageable size. Additionally, a decreased expression results in a negative, but an increased expression level in a positive log2 fold change value. By plotting the log2 fold change on the y-axis, the viewer gets an immediate feeling for what proportion of genes are expressed significantly higher or lower in comparison. MA plots and volcano plots are widely used to represent RNA-Seq data, but there are many other ways to display this kind of data [85].

1.4 READemption

READemption is a bioinformatical pipeline, originally designed to process dRNA-Seq data for determining transcriptional starting sites (TSS) in bacterial genomes [86]. Since, it has been refined and successfully applied to the analysis of a wide range of RNA-Seq data derived from prokaryotes, eukaryotes and viruses. READemption consists of several modules, is implemented in Python and includes parallelisation of working steps. The number of decisions, a user needs to take, is minimized by a predefined set of default parameters that can partly be redefined by the user. The linked modules are implemented in a possibly generic manner and file handling is managed by global predefined variables, which both reduces code. Thus, the framework follows the concepts of 'convention over configuration', 'don't repeat yourself' and 'keep it short and simpel'. Additionally, most working steps result in a variety of output files for the data analysis, including data visualization [62]. All possible subcommands with the corresponding arguments are defined within an executable called 'reademption'. A 'controller' module executes, based on the input commands, the specific working tasks by addressing specific modules. The subcommands are executed via a command line interface, whereas eight subcommands exist with one positional argument and one to several optional arguments each (Fig. 1).

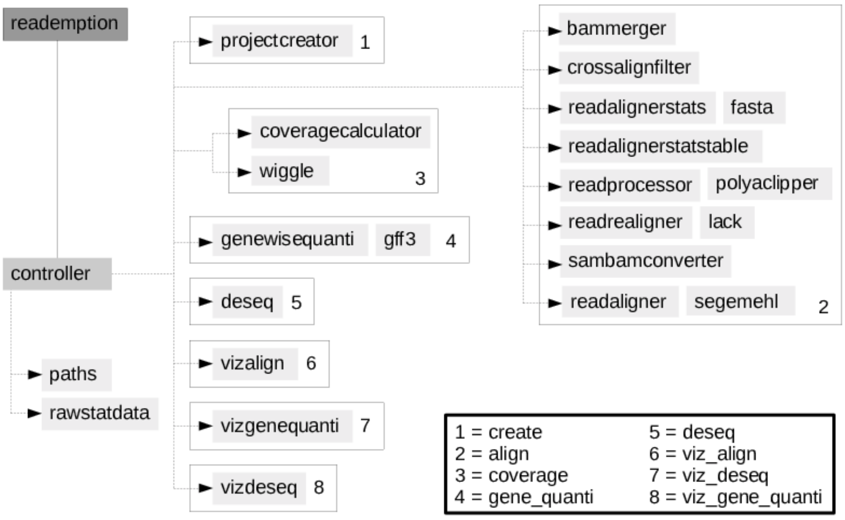


Figure 1: *Module structure of READemption. The arrows indicate, which modules are executed by the controller. The boxes show, which modules are executed upon what subcommand. Some modules are directly positioned side by side. Does the controller execute the one on the left, the right one gets executed too. Note, that not every subcommand necessarily requires all of the modules.*

In the following the different subcommands and its functions are described.

Project creation. The subcommand 'create' creates the folder structure with all input and output folders necessary for the process. The path, in which this folder structure is created, is specified by the positional argument. The required input files need to be stored in the respective folders and from that point on the data handling is performed automatically.

Read alignment. Via the subcommand 'align' read processing and read alignment is executed. By optional arguments poly(A)-clipping, adapter clipping and quality trimming of the reads can be performed. The alignment tool 'segemehl' performs the aligning process, the optional remapping can be performed using lack [87, 88]. Hereby several output files are created. Processed and unaligned reads are stored in FASTA files, whereas the alignments are stored in SAM/BAM format. Also index files created by segemehl are stored. Statistics regarding the mapping process are stored within JavaScript Object Notation (JSON) format and are basis for the visualizations [89].

Nucleotide specific coverage calculation, read per feature quantification & differential gene expression analysis Based on the aligned reads, coverage files in wiggle format are created with the subcommand 'coverage'. Beside raw countings, two different normalization methods are applied on the raw countings and stored in distinct files. In both cases the raw countings are normalized by the total number of aligned reads, but once multiplied by the lowest number of aligned reads of all libraries and once by one million. Several optional arguments allow variable coverage calculation or normalization by the number of uniquely aligned reads. These output files can be viewed by programs like Integrative genome browser (IGB) or used for TSS analysis [90, 91]. Based on annotation files and mapping results, a gene-wise quantification can be performed via the subcommand 'gene_quanti'. Hereby the number of reads, overlapping with each annotation entry, is determined for every library separately and stored in files. The output file additionally contains information about the reference, position on the reference and strand orientation. For a comparative picture, those information are stored within a CSV file, whereas two further differently normalized data sets are created. Once the quantification values are normalized by the tnoar (total number of aligned reads), in the other case by the RPKM (Reads Per Kilobase per Million mapped reads). The RPKM (R) is determined by formula 2 with the number of mappable reads that fall onto the gene (C), the total number of mappable reads (N) and the length of the gene (L) [92].

$$R = \frac{10^9 \cdot C}{N \cdot L} \tag{2}$$

Some optional arguments allow different features, like the definition of considered RNA species to be quantified. The subcommand 'deseq' performs the differential gene analysis of different libraries by DESeq2 [93]. The final output files are extended by the raw read countings of every library, an information that is derived from the 'gene_quanti' step.

Visualization. For visualization of the data, three different subcommands are provided. All of them create PDF files containing different representations of the generated data. 'viz_align' creates two histograms of the read length distribution of the reads before and after read processing. 'viz_gene_quanti' on the one hand provides a bar chart displaying the number of reads per RNA class per library. On the other hand a scatterplot is created, which displays the comparison of the raw gene-wise quantification values for each library pair, including the pearson correlation coefficiant [94]. Via the subcommand 'viz_deseq' a MA-Plot and a volcano plot are created, both combining the data of the different compared libraries. A description of the READemption pipeline, including an example analysis and all subcommands can be viewed online [95]. Testing for READemption includes the evaluation of central procession steps, like creation of the folder structure, the correct file-handling and poly(A)-clipping of the raw reads. Modules, responsible for parsing FASTA and GFF3 files are also tested. The alignment by segemehl and the self implemented gene-wise quantification and coverage calculation are evaluated, just as the central controller.

2 MATERIALS & METHODS

All changes made and the integration of third-party programs were performed using Python 3.5 [96] and different Python libraries on a system using Linux Ubuntu 15.04.

Structural changes. In order to increase usability, the originally single controller module was split into five. Each newly created controller module addresses one of the steps of project creation, mapping, coverage calculation, gene-wise quantification and differential gene expression analysis. At the same time unused modules and functions, as well as deprecated code synthax, were removed or replaced. Subcommands for the visualizations were removed and visualizations were made standard output for the respective procession step. Similar to the 'reademption' module, a 'reademption_tk' module was implemented, containing the subcommands for a toolkit function. Addtionally, a controller for the toolkit functionality was implemented. With this functionality the visualization of the alignment, gene-wise quantification and differential gene expression, as well as coverage calculation, can be performed in a stand alone manner.

Data extraction and handling. For extracting data from JSON files, Python's json library [97] was used. Data extraction and corruption from CSV files was performed using pandas [98] or the csv library [99]. File handling and information extraction from SAM/BAM files was facilitated using the wrapper pysam [100]. Parsing operations of i.e. FASTQ files were performed using Biopython [101]. For generating alignment statistics, the read lengths are no longer derived from the CIGAR (Compact Idiosyncratic Gapped Alignment Report) string, but the actual sequence of the BAM file using pysam's 'query_length' function. Mathematical operations were performed using numpy [102]. Extraction of .xz files is done using the lzma compression library [103]. For the visualizations matplotlib [104] and the python library bokeh [105] were used. The unittest [106] framework for the pipeline was rewritten using pytest [107] and was adapted to the changed module structure.

Input and output options. At the alignment stage the pipeline now can automatically handle input read files in FASTQ format by identification of the '@' character in the header. The visualization of alignment statistics was extended by a bar chart and a histogram. At the gene quantification step a CSV file containing the read number per RNA classes per library is now created. Differential gene expression analysis by READemption calculates the log2 fold change of two compared libraries. Upon user demand, the CSV file is extended by a column containing the respective fold changes by exponentiating the basis 2 with the log2 fold change. Furthermore the volcano plot was removed and replaced by a manhattan plot.

Integrated third-party programs. As an additional read processing option cutadapt [108] was integrated. This functionality can be chosen at the alignment stage by optional arguments. The alignment tool STAR [109] was included to the pipeline as a further option to perform alignments. Runtime evaluation of the alignment tools was performed using the shell command 'time'. Hereby the 'real' value was considered, as it reflects the time passed from the start to the end of the whole process. Both determined runtimes reflect the average of three distinct runtime evaluations. In both cases the alignment was performed with READemption's read processing step, including poly(A)-clipping, on a server using 24 threads. Alignment to small genomes with STAR requires a parameter and value representing the genome size. The parameter in READemption is abbreviated as '-iN' and the value V is calculated by formula 3 with the genome size G in bases.

$$V = \frac{log_2(G)}{2} - 1 \tag{3}$$

The percentage of aligned reads was determined by dividing the total number of aligned reads by the number of reads being long enough for alignment. Per default, only reads being at least 12 bases in length after read processing are used in the alignment. The percentage of uniquely aligned reads was calculated by dividing the number of uniquely aligned reads by the total number of aligned reads. Both types of calculation were applied on every library of the two different processes. The means of the single percentages calculated library-wise were determined in order to use them for performance comparison.

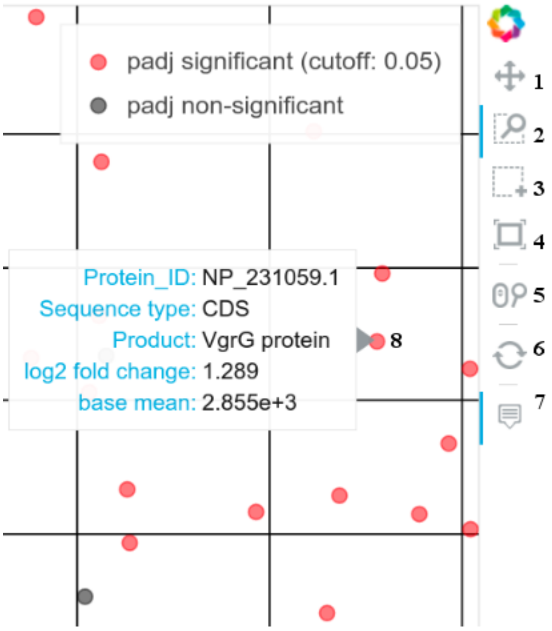
Test data. For evaluation of the made changes, part of a single-end dataset from a dRNA-Seq approach was used. The dataset consists of four libraries from the transcriptome of *V. cholerae* 01 biovar El Tor str. N16961. The libraries are derived from samples harvested at a optical density (OD) of 0.1 and at OD 2.0, with two replicates each. All samples used here are not TEX treated. Sequencing of the libraries was performed with an Illumina platform resulting in reads of up to $\sim$ 100 bases in length.

The extended version of READemption is freely available online under the ICS license [110].

3 RESULTS & DISCUSSION

3.1 Input and output

Having a pipeline combining tools and being able to perform different operations with specific parameters is beneficial. READemption provides a big number of, partly optional, parameters in order to offer a flexible data analysis. However, the more parameters exist, the more a user has to consider what combination to use, which comes to the cost of usability. Therefore parameters that are not essential may be avoided. This has been done for the '–fastq' parameter, as the read input files now are automatically identified as FASTA or FASTQ format. Similarly, .xz compressed input files are now automatically extracted by READemption. Before, it had to be done manually upfront the process. The additional output of the gene-wise quantification and differential gene expression analysis step were both implemented due to user request and increase the output's information spectrum. Integrating other tools increased the number of parameters, but it was tried to keep it at a minimum, while having maximum functionality. An example is the integration of cutadapt, where the user needs to set the '–cutadapt' parameter. All additional parameters are given in the cutadapt-specific form within a list. Especially for those, being familiar with cutadapt, this is an advantage. Visualization output for READemption has been upgraded in terms of more data being visualized and creation of interactive figures. Bokeh and matplotlib were used to visualize the same data in the same manner, whereas matplotlib creates publication-quality figures in PDF format. Bokeh, however, generates interactive plots and charts in HTML format. The HTML files can be opened with any browser in which predefined interactive actions can be performed, for instance zooming options. Firstly, zooming in and out a figure can be done with the mouse wheel (wheel zoom tool), but also by selecting a rectangular area (box zoom tool). A box select tool highlights a chosen rectangular area for visual distinction. In some cases the figures exceed the displayed coordinate systems, whereas the pan tool allows to shift the displayed area of the figure in any direction. Also the size of the coordinate system can be altered via the resize tool. A reset button enables to return to the original presentation of the figures. A hover tool allows to inspect any glyph of a scatter plot or bars in charts and histograms upon information. By moving the cursor on it, a box appears showing predefined information referring to the select object. Figure 2 shows a section of bokeh's html interface with the toolbar. The aligning step of READemption generates two JSON files containing information about the read procession ('read_processing.json') and read alignment ('read_alignments_final.json'). Based on 'read_processing.json', a histogram showing the read length distribution after read procession is created.



● padj significant (cutoff: 0.05)
● padj non-significant

Protein_ID: NP_231059.1
Sequence type: CDS
Product: VgrG protein
log2 fold change: 1.289
base mean: 2.855e+3

Figure 2: *Toolbar of bokeh. The numbers indicate the buttons for pan tool (1), box zoom (2), box select (3), resizing (4), wheel zoom (5), resetting (6) and the hover tool (7). A blue bar indicates the tool being activated. Here it is zoomed in, the cursor is located on the glyph (8) indicated by the the grey triangle. Due to the hover tool a box appears, showing predefined information about the glyph's data.*

Furthermore, the total number of processed input reads for the alignment are displayed by a bar chart, each library represented by one bar. Those bars are visually subdivided into unmodified, single(A)- and poly(A)-clipped reads (Fig. 3). Based on data derived from 'read_alignments_final.json', a histogram showing the read frequency and length of aligned reads is created. Based on both output files, a histogram per library is generated, each showing the number of total input reads, the number of input reads being long enough for alignment, the number of aligned reads and the number of uniquely aligned reads (Fig. 4). The visualizations give a valuable oversight over the read procession and alignment process. Also they make the whole pipeline handling less error prone and time intensive. For instance, users may forget to set the parameter for poly(A)-clipping. This has impact on the alignment outcome, consequently leading to inaccurate downstream analysis results. This can be immediately detected by viewing the visualization that shows the ratio of poly(A)-clipped reads.

The gene-wise quantification step determines how many of the aligned reads overlap with which annotation entries. Hereby one output CSV file is created, that combines this information and serves as basis for the visualizations. Those are the same as before, but additionally generated by bokeh (Fig. 5 and Fig.6). The bar chart shows read countings of the different annotation classes. Pan tool, both zoom tools and the hover tool are integrated. In this case the zoom tools are useful, as rRNA and tRNA countings cannot be recognized without them. The hover tool shows the number of reads per class, as the exact number is hard to recognize from the bar. This information is additionally stored within a CSV file as standard output. The scatter plot showing the correlation of the read counts is equipped with the same tools. The hover tool displays the NCBI protein ID, the final product and the type of RNA species or coding sequence, predefined by the parameter '–features'. This information might be especialy useful for inspecting outliers.

At the differential gene expression analysis step, just as before, a MA plot is

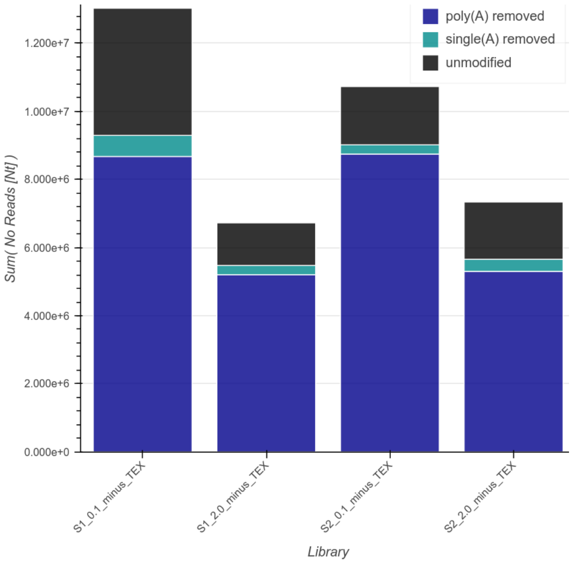


Figure 3: Read processing overview. A bar per library shows the total number of input reads. Each bar is subdivided into the number of poly(A)-clipped, single(A)-clipped and unmodified reads.

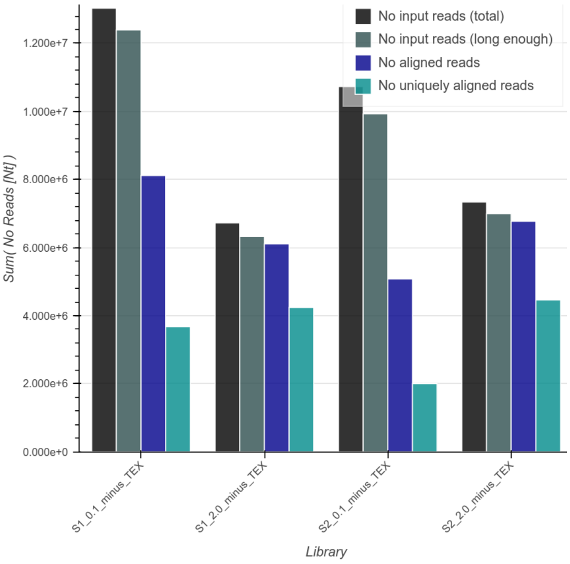


Figure 4: Whole process overview. A histogram per library shows the number of total input reads, reads that are long enough for alignment, aligned reads and uniquely aligned reads.

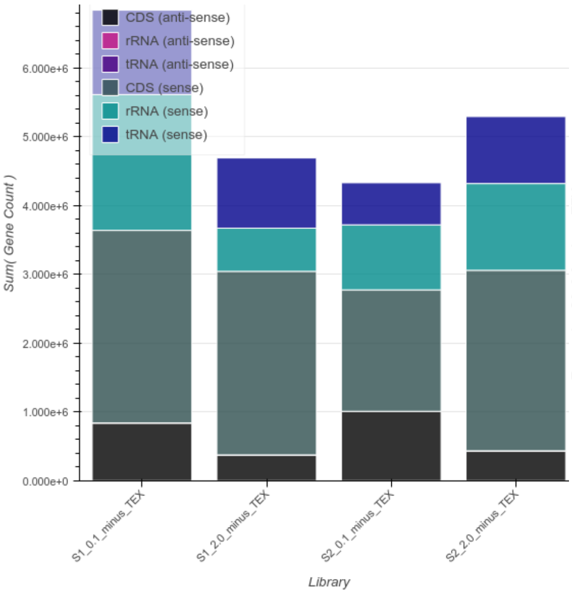


Figure 5: Read number per RNA class. A bar per library represents the counted read number per feature, derived from the annotation. The features to be shown are specified by the parameter '–features'.

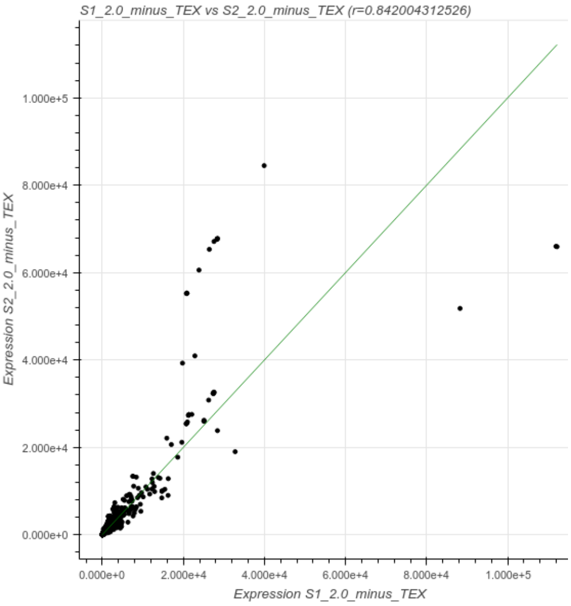


Figure 6: Correlation plot. A scatter plot comparing the gene-wise countings calculated for every library. In addition the pearson correlation coefficient (r) is displayed.

generated. The plot displays the log2 fold change and the log10 of the base mean. The log10 of the base mean is used making the plot appear more dense to a manageable size. Hereby the user gets an immediate feeling for what proportion of genes are higher or lower expressed compared to one another. Additional information can be derived from the interactive plots by using bokeh tools. Pan tool, the zoom tools, box select tool, resize tool and the reset button allow navigation throughout the plot. The hover tool displays protein ID, type and product of the transcript, the log2 fold change and the base mean. The MA plot generated using bokeh is shown in figure 7.

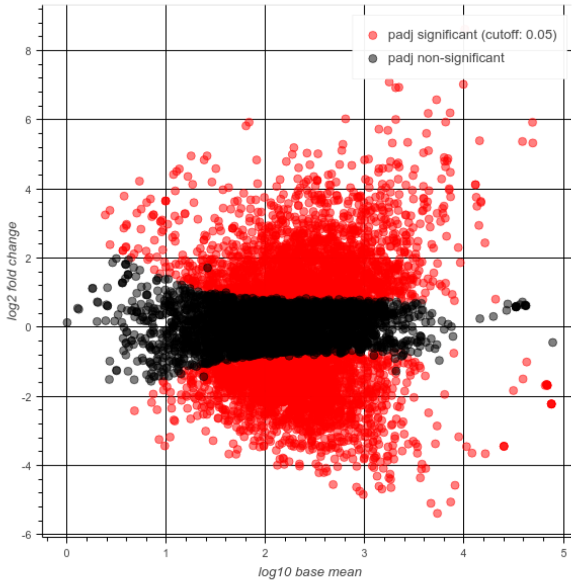


Figure 7: *MA plot. A scatterplot with the log2 fold change on the x-axis and the log10 of the base mean on the y-axis. The log10 of the base mean is used for a more dense representation. The padj cutoff being set can be derived from the legend. For every library comparison one MA plot is generated.*

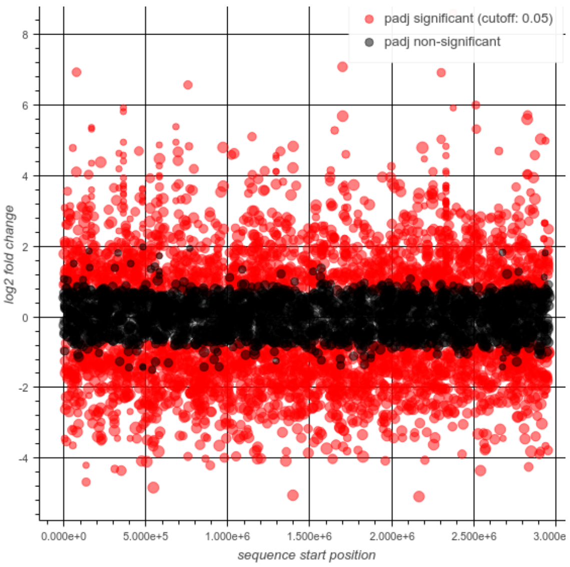


Figure 8: *Manhattan plot. A scatterplot with the log2 fold change on the x-axis and the genome start position on the y-axis. The padj cutoff is shown in the legend. The glyph sizes per default refelect the log2 of the sequence lengths. A manhattan plot is generated per chromosome for every compared library.*

Instead of additionally generating a volcano plot, now a manhattan plot is created. This is achieved by plotting the log2 fold change on the x-axis and the sequence start position of the annotated read on the y-axis. Additionally, the size of the glyphs in the plot per default represent the log2 of the annotated sequence size (Fig. 8). Manhattan plots are generated per chromosome for every compared library. This is quite an instructive visualization, giving the user a feeling for what gene expression alteration in which part of the reference sequence takes place, including the transcript's size. The bokeh tools being used correspond to the ones of the MA plot. For both, the MA and manhattan plot, glyphs revealing a padj below the default cutoff of 0.05 are red and glyphss with a padj higher than 0.05 are colored black. This allows the discrimination of the fold change's significance. The CSV file, that serves as a basis for these visualizations, now additionally contains the fold change of every annotated read. To gain the data for visualization in the first place, different modules are used depending on the input file type. JSON files created by the alignment step contain multi-level libraries which are accessed by the json module and extracted using the python standard library. However, the CSV input files are now transformed to 2-dimensional labeled data structures by pandas, called 'DataFrames'. As an example, the deseq step results in CSV files harboring the compared and annotated reads row-wise. The rows are subdivided into several columns, bearing information regarding i.e. log2 fold change or adjusted p-values (padj) of the compared read libraries. For a differentiated coloring of the glyphs based on the respective padj values, the creation of two different DataFrames is required. Information derived from those two DataFrames are plotted by bokeh in two different processes, but are then displayed within one graph. To split the DataFrame, panda's splicing can be used. As a result, one DataFrame contains all padj-columns below a certain cutoff including the respective rows. The other DataFrame contains the rows in which the padj-columns reveal values higher than the cutoff. This procedure is computational less expensive than other methods, e.g. using for-loops, to extract data.

3.2 Module structure and toolkit function

The usage of one central controller made it a rather unclear module of nearly 1000 lines of code. Hence, the controller was split into five modules. In this course, a 'helper' module was created, containing functions which are used by all of the controllers. At the same time the visualizations were made standard output for the respective analysis step, which made the subcommands for visualizations unneccessary. This leaves five subcommands for the project creation, read processing and mapping process, coverage calculation, gene-wise quantification and differential gene expression analysis with DESeq2. Each of those central procession steps now is managed by one controller, executing the necessary modules. Additionally, the unused 'fastq' module and the module 'readaligner', which was designed as an abstraction layer for read aligners, were removed. Those measures increase the developer's oversight and makes the integration of further functionalities more efficient. Making the visualizations standard output also decreases the user's decisions to take and increases the default output spectrum. Nevertheless, the option to only visualize READemption results is maintained by the implementation of a toolkit functionality. This functionality also eneables the performance of coverage calculation, based on BAM/SAM output files from read aligning tools. For this purpose, two further modules were created. Similar to 'reademtion', the 'reademption_tk' module contains the subcommands and parameters necessary for the toolkit functionality. Dependent on the subcommands given, the 'controller_tk' executes the existing READemption modules, in order to perform the desired action. As shown in figure 9, the toolkit functionality provides four different subcommands, which are 'viz_align', 'viz_gene_quanti', 'viz_deseq' and ’coverage’.

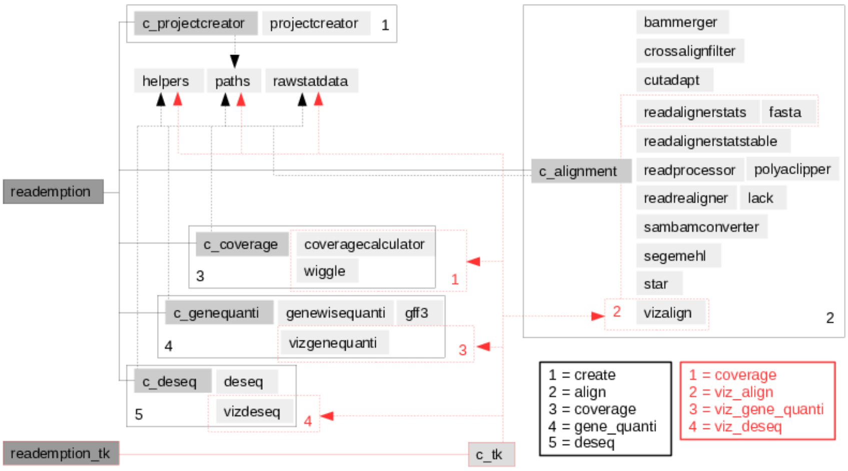


Figure 9: *Modified module structure of READemption. The black boxes indicate, which controllers and other modules are executed upon which input command managed by the ’reademption’ executable. The subcommands of the toolkit function are managed by the ’reademption_tk’ executable. The red arrows and boxes indicate, which modules are executed upon which toolkit command.*

Basis for visualizing the alignment process within the pipeline are two JSON files. In case of the toolkit, the visualizations can be generated based on both or either one of the files by specifing it via parameters. The parameter ’–input_process’ refers to the file ’read_processing.json’ and ’–input_align’ to the file ’read_alignments_final.json’. If none of the files is existant, ’read_alignments_final.json’ can be created based on the aligning result BAM files via the subcommand ’–input_BAM’. The visualization can then be performed using ’–input_align’. The subcommand ’viz_gene_quanti’ generates the plots regarding the gene-wise quantification step. Hereby the READemption specific file ’gene_wise_quantifications_combined.csv’ is required, which is submitted by the parameter ’–input_file’. Additionally the user needs to submit the library names via the parameter ’–lib_names’, as those are identified in the pipeline process upfront the gene-wise quantification step. The visualization of the differential gene expression analysis step is initiated by the subcommand ’viz_deseq’. For this, the pipeline specific output files are required, as they include information which are not provided by DESeq2 only. The input file is submitted by ’–input_file’ and the conditions by the parameter ’–condition’. For both, the MA plot and manhattan plot, a default padj cutoff of 0.05 is set, but the user can optionally set a different value. This leads to an altered appearance of the plots, as data glyphs representing significantly and non-significantly changed expression levels of reads are colored differently. Fot the MA plot, the user can optionally choose the glyph size, glyph transparency, glyph shape and the glyph colors, for glyphs representing significant and non-significant fold changes. With the subcommand ’coverage’, coverage calculations based upon BAM files can be performed. The parameters correspond to the ones from the pipeline, only parameters for the input and output handling were added. These new toolkit options increase the users flexibility and some analyis steps can be performed independent from the pipeline. In the case of the MA and manhattan plots the visualizations can be repeated using cutoffs other than the default ones.

3.3 Third-party tools

Read processing. The self implemented read processing procedure of READemption can perform adapter and poly(A) clipping, as well as quality trimming on the basis of a user-defined minimal phred score. Also the removal of reads below a user-defined minimal length is possible. For paired-end libraries, poly(A)-clipping is not possible. In order to increase read processing options, cutadapt was integrated into the pipeline. Cutadapt supports several read modifications, such as trimming of 5' or 3' adapters, including anchored or linked adapters. By specifying the adapter as muliplte consecutive As, a poly(A) clipping can be performed. As cutadapt supports multiple adapter removal, the clipping of poly(A)s and adapters can be performed at the same time. Furthermore the adapter clipping is error tolerant, with allowed errors being mismatches or indels. Hereby the number of allowed errors is definable by the user. Quality trimming can be performed for both ends of the reads, whereas the quality scores must be ASCII-encoded. Also read filtering via cutadapt offers more options, than the READemptions solution does. This includes the removal of reads that exceed a defined length or untrimmed reads. Most of the options for single-end read trimming also apply for paired-end reads, such as multiple adapter clipping. As cutadapt was initially designed to trim reads derived from the ABi SOLiD sequencer, the tool is able to trim colorspace reads [108, 111]. In general, read processing upfront the read alignment is a crucial working step as it influences the accuracy of the alignment. The various HTS technologies have a characteristic impact on the raw reads, i.e. the extension of the read sequences by one or more adapter sequences. This can be adressed by the variety of options provided by cutadapt. Compared to other trimming tools, cutadapt performs very good in terms of quality trimming by maximizing the number of alignable reads [112, 113]. Therefore integrating cutadapt into READemption increases the range of raw read data that can be processed and therefore most likely the accuracy of upstream processes, like read alignment. Note, that visualizations are not provided, when performing the 'align' step using cutadapt. The necessary files are generated by the self implemented read processing step and after the actual alignment, respectively. This functionality has not yet been implemented for combining the read alignment with cutadapt and needs to be addressed in the future.

Read Alignment. As a supplement to segemehl for read alignment, the alignment tool STAR (Spliced Transcript Alignment to a Reference) was integrated into READemption. The accurate alignment of RNA-Seq reads containing mismatches or indels, but also reads representing transcripts derived from splicing events, are computational expensive. In order to challenge those tasks, the STAR algorithm consists of two central steps. The first step consists of a seed search, which is a sequential search for the Maximal Mappable Prefix (MMP) within the reference genome. In order to do so, the reference genome is first converted into a uncompressed suffix array (SA) [114]. By comparing every read to the indexed genome, the MMP is defined as the maximal possible prefix of a read that exactly matches the reference sequence. This MMP search then is repeated for the unmapped portion of the read. If a read spans a splice junction, then the unmapped portion can be precisly mapped to another region of the reference sequence. If a MMP does not span the whole read and the unmapped portion cannot be aligned precisly elsewhere, the MMP serves as an anchor in the genome. Those anchors are then extended, allowing for mismatches and indels. If the extension process hints on a adapter or poor quality sequence, the sequence gets trimmed of. The second step involves clustering of MMPs by their proximity to selected anchors. MMPs are then stitched together within a user-defined genomic region around the anchors, wheras any number of mismatches, but only one indel is allowed. If a read within a genomic region cannot be fully aligned, other regions are searched for complete alignment, which may result in chimeric alignments. Based on user-defined quality scores, stitched combinations fullfilling the highest score are reported as alignments. The efficient MMP search algorithm in combination with the usage of uncompressed SAs makes STAR very fast [109]. Earlier studies suggest, that STAR aligns RNA-Seq reads with a equivalent high senitivity compared to segemehl. However, segemehl reveals lower false positive alignments, when aligning RNA-Seq reads. Both alignment tools offer an option for realigning previously unaligned reads. Hereby the discovery rate of segemehl's lack tool showed to be well, especially with reads of ∼ 1000 bp derived from a 454 pyrosequencing platform [88]. Nevertheless, STAR performs considerable faster. This can be demonstrated by comparing READemption's 'align' step, once using segemehl and once using STAR as a mapper. Hereby the process using segemehl requires on average 94 minutes and 47.74 seconds, whereas the same process performed with STAR requires 54 minutes and 8.32 seconds. STAR reveals a mean percentage of 77.52 % aligned reads, with 55.04 % of them being uniquely aligned. Segemehl however, achieves a mean of 68.16 % aligned reads, of which 60.08 % are uniquely aligned. Note, that STAR achieves a higher total number of uniquely aligned reads than segemehl does, only the percentage of uniquely aligned reads is lower. By using real data derived from an experiment, it is generally impossible to tell for certain, if a read got mapped correctly or not. The number of aligned reads also depends on how many indels or mismatches are allowed per read by the alignment tool. Taking this into consideration, the fact that STAR shows more aligned reads, does not necessarily reflect a higher proportion of correctly mapped reads. Still, the percentage of mapped reads is used as an indicator for overall sequencing accuracy [68]. Consequently, this indicates a higher accuracy of STAR for a short-read RNA-Seq dataset, while the runtime is nearly decreased by half. The accuracy might be even increased by performing the mapping with the annotation, enabled by the parameter '–include_annotation'.

3.4 Software testing

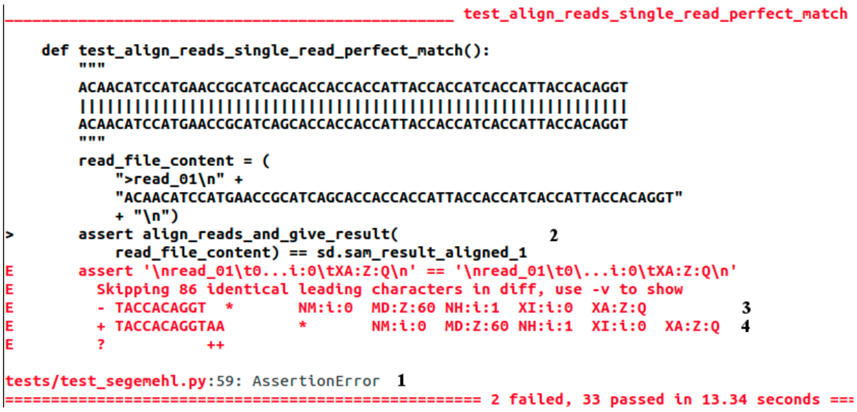
Software testing is a verification technique in order to improve softwares. Software testing can be subdivided into static and dynamic testing. Static testing denotes the analysis of the code without executing it, for example by code review. Setting up test cases and executing the written code is called dynamic testing. Within test cases different conditions can be simulated, under which the software should perform in a specified manner. This is valuable, as testing is usually performed at the development stage and therefore helps to minimize bugs and errors before the software release. One part of dynamic testing is unit testing, whereby single modules of a software are executed and checked upon correct functionality. In the case of READemption, this is put into practice, mainly by generating dummy data and the expected output. Then a module is esecuted processing the dummy data, whereas this result is compared with the expected output. As the most important modules and working steps were already covered by READemption, the priority was put on the testing framework. In concrete terms, instaed of using unittest from the python standard library, now the pytest framework is utilized. Still, changes had to be made, to adapt to the changed module structure of the pipeline. Pytest reveals several features to favor it over unittest, for instance auto-discovery. By executing the command 'pytest' in the command-line interpreter, pytest collects any python module prefixed with 'test_' starting from the current directory. Hereby pytest can also run other test cases, e.g. implemented in unittest. This makes the test execution very easy. Furthermore, using pytest decreases boilerplate code. Comparisons, e.g. between processed dummy data and expected output, can be performed in a simple manner using the assert statement. As an example, a unittest based data comparison looks like the following:

```
import unittest

class UnittestObject(unittest.TestCase):

    def compare_output_and_expectation(self):
        self.assertEqual(True, True)


if __name__ == '__main__':
    unittest.main()
```

By using pytest, the same functionality can be achieved through following code:

```
def compare_output_and_expectation():
    assert True == True
```

As shown in the example above, pytest works without the usage of objects. Only so called 'mock objects', which mimic the behaviour of objects of the pipeline, were kept by. As an example, mock objects mimic the initialisation of objects containing arguments. By defining different mock objects, the execution of subcommands with different parameters can be simulated. In addition to converting the test framework, some of the dummy data or mock classes were stored in external files, which are linked to the respective test modules. This increases the readibility of the test modules and prevents multiple creation of the same data within different files. A further advantage of pytest is its detailed information on failing tests, especially of assert statements (Fig. 10). This is useful in order to detect reasons for failures in a short period of time.



```
_________________________________________________ test_align_reads_single_read_perfect_match

    def test_align_reads_single_read_perfect_match():
        """
        ACAACATCCATGAACCGCATCAGCACCACCACCATTACCACCATCACCATTACCACAGGT
        ||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||
        ACAACATCCATGAACCGCATCAGCACCACCACCATTACCACCATCACCATTACCACAGGT
        """
        read_file_content = (
            ">read_01\n" +
            "ACAACATCCATGAACCGCATCAGCACCACCACCATTACCACCATCACCATTACCACAGGT"
            + "\n")
>       assert align_reads_and_give_result(                    2
            read_file_content) == sd.sam_result_aligned_1
E       assert '\nread_01\t0...i:0\tXA:Z:Q\n' == '\nread_01\t0\...i:0\tXA:Z:Q\n'
E         Skipping 86 identical leading characters in diff, use -v to show
E         - TACCACAGGT  *       NM:i:0  MD:Z:60 NH:i:1  XI:i:0  XA:Z:Q          3
E         + TACCACAGGTAA        *       NM:i:0  MD:Z:60 NH:i:1  XI:i:0  XA:Z:Q  4
E         ?           ++

tests/test_segemehl.py:59: AssertionError  1
===================================================== 2 failed, 33 passed in 13.34 seconds ===
```

Figure 10: *Pytest error report. It is displayed, what sort of error is caused in which module (1) by which assertion (2). Furthermore the result (3) and the predefined expectation (4) are directly compared, thereby highlighting the cause of the error.*

3.5 Future prospects

READemption is a pipeline that offers a variety of options to analyse RNA-Seq data. Still, some aspects may be further improved. More test modules, or optimally a system test, would increase READemption's robustness. The efficiency of data procession on a code level can be further increased, e.g. by applying features like panda's splicing method in more possible cases. The read procession by cutadapt does not generate the required output files for visualization. However, the information about how many reads of which length are aligned is derived from the alignment output and also cutadapt can optionally generate a report file about the read procession. Based on this, at least some of the visualizations could be generated. The toolkit function could be further extended, to offer more analysis steps that can be performed in a standalone manner. More parameters for the toolkit visualizations would enable users to shape plots according to their taste. Furthermore, error reports of some pipeline processes could be integrated. Sometimes, if the alignment process does not work for any reason, an empty SAM file is created which is detected at the stage of converting SAM to BAM files. This results in a misleading error report in the command-line interpreter. To solve this, a function checking the alignment output upon content could be implemented. This may be combined with a module like logging [115], in order display information about a current checkup. This principle of implementing more checkups and providing the user with information about the pipeline's status could also be applied in other cases. Also, if a process of the pipeline is interrupted, the process needs to be repeated from scratch. This mostly includes manual removal of generated output files. In order to offer the possibility of automatically reentering the process from a stage shortly before interruption, a wrapper like luigi [116] may be included. READemption is originally designed as a command-line tool for linux operation systems. For somebody not being familiar to perform tasks from a command-line interface, a graphical user interface would be beneficial. Also the integration of other tools at different stages of the analysis process is quite conceivable. As an example, the analysis of CLIP-Seq data requires procession steps different than the ones for analysing dRNA- or dual RNA-Seq data, which requires additional tools [117].

3.6 Conclusion

READemption is now capable of processing input read files in FASTQ format without the necessity of setting a parameter. Also .xz compressed input read files can be extracted automatically. Those measures make the pipeline more user-friendly, as considerations and actions of the user upfront the process are reduced. The generation of additional output increases the range of information about the processed data. At the read processing and alignment step two further visualizations are generated, a bar chart and a histogram. The bar chart reflects the poly(A)-clipping process of the input reads, which is useful as a sanity check. The histogram gives an overview of how many of the total input reads are used as input for the alignment, as well as how many of those are aligned either multiple times or uniquely. Hereby the user gets an immediate feeling for the quality of the read alignment process. The gene-wise quantification step now creates an additional CSV file containing the read number per RNA class per library. This was done upon user request, as the exact numbers of reads could not have been derived from the respective visualization. Also upon user request, the final output CSV file of the differential gene expression analysis step now contains an additional column showing the fold changes of the respective row. Additionally, a manhattan plot instead of a volcano plot now is created at this step. The manhattan plot gives an impression of the fold change significance of reads in combination with the gene position in the genome coding for those reads. By additionally generating interactive plots with bokeh, a deeper exploration of the data using bokeh tools is possible. Not only do the data visualizations increase the output information spectrum. They enable the digestion of complex datasets in the first place, including the recognition of certain patterns. Using pandas to extract data from input files further increases efficiency of the differential gene expression step. The integration of the third-party tools cutadapt and STAR increase functionality and performance of the pipeline. Cutadapt is a sophisticated tool for quality trimming and adapter-clipping. The tool's whole functionlity can be exploited within the pipeline, which increases READemption's read procession options. The integration of the read alignment tool STAR drastically decreases the alignment's runtime and memory footprint. At the same time STAR presumably slightly increases the quality of short read alignment. In contrast, segemehl shows advantages when it comes to the alignment of longer reads, whereas segemehl's remapping tool lack tackles the issue of reads spanning multiple splice junctions. By using pytest instead of unittest for software testing, boilerplate code is reduced. This increases the readability of the test modules and therefore also simplifies their extension. Furthermore, the error reports provided by pytest are more precise, making error detection more efficient. Due to the toolkit functionality some procession steps can now be performed in a standalone manner, whereas the visualiazations require input files generated by READemption. Via several parameters the visualizations can be altered straight away in order to represent data in a different manner, e.g. by changing padj cutoffs of MA plots or manhattan plots. Also coverage calculation can be performed in a standalone manner using the toolkit functionality, whereas the input not necessarily needs to originate from READemption. Changing the module structure of READemptions also bears advantages. In the course of this measure, visualizations were made standard output for the respective procession steps. This reduces the need for subcommands, thus further increasing READemption's usability. Note, that the former provided flexibility through the subcommands for visualizations is preserved by the toolkit functionality. Furthermore, splitting the controller into five controllers increases the readability. This is especially beneficial with regard to future extensions of READemption's functionality. This extensions may include the integration of tools, which are designed for processing data derived from other RNA-Seq experiments, such as CLIP-Seq. Providing the user with information about the current status could be achieved by integrating checkups in combination with a module like logging. In order to increase READemption's robustness, a wrapper might be utilized. In addition to this, some made changes could also be applied in other cases, for example panda's splicing function. Also the toolkit functionality could be extended, as well as the range of tools for interactice visualizations. The robustness of READemption could be further increased by more test modules or a system test. An option to increase the spectrum of potential users would be the integration of a graphical user interface. All in all READemption has been improved regarding many aspects and now offers a wider range of functionalities. Nevertheless, there is still room for improvement, whereas a solid basis for future integrations and optimizations is laid.

References

[1] Heather, J. M. and Chain, B. (2016) The sequence of sequencers: The history of sequencing DNA. *Genomics,* **107**(1), 1 – 8.

[2] Maxam, A. M. and Gilbert, W. (1977) A New Method for Sequencing DNA. *Proceedings of the National Academy of Sciences of the United States of America,* **74**(2), 560–564.

[3] Sanger, F. and Coulson, A. (1975) A rapid method for determining sequences in DNA by primed synthesis with DNA polymerase. *Journal of Molecular Biology,* **94**(3), 441 – 448.

[4] Sanger, F., Air, G. M., Barrell, B. G., Brown, N. L., Coulson, A. R., Fiddes, J. C., Hutchison, C. A., Slocombe, P. M., and Smith, M. (1977) Nucleotide sequence of bacteriophage Phi X174 DNA. *Nature,* **265**(5596), 687 – 695.

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

[6] Anderson, S. (1981) Shotgun DNA sequencing using cloned DNase I-generated fragments.. *Nucleic Acids Research,* **9**(13), 3015 – 3027.

[7] Smith, L. M., Sanders, J. Z., Kaiser, R. J., Hughes, P., Dodd, C., Connell, C. R., Heiner, C., Kent, S. B. H., and Hood, L. E. (1986) Fluorescence detection in automated DNA sequence analysis. *Nature,* **321**(6071), 674 – 679.

[8] Vesterberg, O. (1989) History of electrophoretic methods. *Journal of Chromatography A,* **480**(doi:10.1016/S0021-9673(01)84276-X).

[9] Hunkapiller, T., Kaiser, R., Koop, B., and Hood, L. (1991) Large-scale and automated DNA sequence determination. *Science,* **254**(5028), 59–67.

[10] Venter, J. C., Adams, M. D., Myers, E. W., Li, P. W., Mural, R. J., Sutton, G. G., Smith, H. O., Yandell, M., Evans, C. A., Holt, R. A., Gocayne, J. D., Amanatides, P., Ballew, R. M., Huson, D. H., Wortman, J. R., Zhang, Q., Kodira, C. D., Zheng, X. H., Chen, L., Skupski, M., Subramanian, G., Thomas, P. D., Zhang, J., Gabor Miklos, G. L., Nelson, C., Broder, S., Clark, A. G., Nadeau, J., McKusick, V. A., Zinder, N., Levine, A. J., Roberts, R. J., Simon, M., Slayman, C., Hunkapiller, M., Bolanos, R., Delcher, A., Dew, I., Fasulo, D., Flanigan, M., Florea, L., Halpern, A., Hannenhalli, S., Kravitz, S., Levy, S., Mobarry, C., Reinert, K., Remington, K., Abu-Threideh, J., Beasley, E., Biddick, K., Bonazzi, V., Brandon, R., Cargill, M., Chandramouliswaran, I.,

Charlab, R., Chaturvedi, K., Deng, Z., Francesco, V. D., Dunn, P., Eilbeck, K., Evangelista, C., Gabrielian, A. E., Gan, W., Ge, W., Gong, F., Gu, Z., Guan, P., Heiman, T. J., Higgins, M. E., Ji, R.-R., Ke, Z., Ketchum, K. A., Lai, Z., Lei, Y., Li, Z., Li, J., Liang, Y., Lin, X., Lu, F., Merkulov, G. V., Milshina, N., Moore, H. M., Naik, A. K., Narayan, V. A., Neelam, B., Nusskern, D., Rusch, D. B., Salzberg, S., Shao, W., Shue, B., Sun, J., Wang, Z. Y., Wang, A., Wang, X., Wang, J., Wei, M.-H., Wides, R., Xiao, C., Yan, C., Yao, A., Ye, J., Zhan, M., Zhang, W., Zhang, H., Zhao, Q., Zheng, L., Zhong, F., Zhong, W., Zhu, S. C., Zhao, S., Gilbert, D., Baumhueter, S., Spier, G., Carter, C., Cravchik, A., Woodage, T., Ali, F., An, H., Awe, A., Baldwin, D., Baden, H., Barnstead, M., Barrow, I., Beeson, K., Busam, D., Carver, A., Center, A., Cheng, M. L., Curry, L., Danaher, S., Davenport, L., Desilets, R., Dietz, S., Dodson, K., Doup, L., Ferriera, S., Garg, N., Gluecksmann, A., Hart, B., Haynes, J., Haynes, C., Heiner, C., Hladun, S., Hostin, D., Houck, J., Howland, T., Ibegwam, C., Johnson, J., Kalush, F., Kline, L., Koduru, S., Love, A., Mann, F., May, D., McCawley, S., McIntosh, T., McMullen, I., Moy, M., Moy, L., Murphy, B., Nelson, K., Pfannkoch, C., Pratts, E., Puri, V., Qureshi, H., Reardon, M., Rodriguez, R., Rogers, Y.-H., Romblad, D., Ruhfel, B., Scott, R., Sitter, C., Smallwood, M., Stewart, E., Strong, R., Suh, E., Thomas, R., Tint, N. N., Tse, S., Vech, C., Wang, G., Wetter, J., Williams, S., Williams, M., Windsor, S., Winn-Deen, E., Wolfe, K., Zaveri, J., Zaveri, K., Abril, J. F., Guigó, R., Campbell, M. J., Sjolander, K. V., Karlak, B., Kejariwal, A., Mi, H., Lazareva, B., Hatton, T., Narechania, A., Diemer, K., Muruganujan, A., Guo, N., Sato, S., Bafna, V., Istrail, S., Lippert, R., Schwartz, R., Walenz, B., Yooseph, S., Allen, D., Basu, A., Baxendale, J., Blick, L., Caminha, M., Carnes-Stine, J., Caulk, P., Chiang, Y.-H., Coyne, M., Dahlke, C., Mays, A. D., Dombroski, M., Donnelly, M., Ely, D., Esparham, S., Fosler, C., Gire, H., Glanowski, S., Glasser, K., Glodek, A., Gorokhov, M., Graham, K., Gropman, B., Harris, M., Heil, J., Henderson, S., Hoover, J., Jennings, D., Jordan, C., Jordan, J., Kasha, J., Kagan, L., Kraft, C., Levitsky, A., Lewis, M., Liu, X., Lopez, J., Ma, D., Majoros, W., McDaniel, J., Murphy, S., Newman, M., Nguyen, T., Nguyen, N., Nodell, M., Pan, S., Peck, J., Peterson, M., Rowe, W., Sanders, R., Scott, J., Simpson, M., Smith, T., Sprague, A., Stockwell, T., Turner, R., Venter, E., Wang, M., Wen, M., Wu, D., Wu, M., Xia, A., Zandieh, A., and Zhu, X. (2001) The Sequence of the Human Genome. *Science,* **291**(5507), 1304–1351.

[11] Churko, J. M., Mantalas, G. L., Snyder, M. P., and Wu, J. C. (2013) Overview of High Throughput Sequencing Technologies to Elucidate Molecular Pathways in Cardiovascular Diseases. *Circulation research,*

(doi:10.1161/CIRCRESAHA.113.300939).

[12] Mardis, E. R. (2017) The impact of next-generation sequencing technology on genetics. *Trends in Genetics,* **24**(3), 133 – 141.

[13] Fullwood, M. J., Wei, C.-L., Liu, E. T., and Ruan, Y. (2009) Next-generation DNA sequencing of paired-end tags (PET) for transcriptome and genome analyses. *Genome Research,* **19**(4), 521 – 532.

[14] Lee, H., Gurtowski, J., Yoo, S., Nattestad, M., Marcus, S., Goodwin, S., McCombie, W. R., and Schatz, M. (2016) Third-generation sequencing and the future of genomics. *bioRxiv,* (doi:10.1101/048603).

[15] Park, S.-J., Saito-Adachi, M., Komiyama, Y., and Nakai, K. (2016) Advances, practice, and clinical perspectives in high-throughput sequencing. *Oral Diseases,* **22**(5), 353–364.

[16] Brockman, W., Alvarez, P., Young, S., Garber, M., Giannoukos, G., Lee, W. L., Russ, C., Lander, E. S., Nusbaum, C., and Jaffe, D. B. (2008) Quality scores and SNP detection in sequencing-by-synthesis systems. *Genome Research,* **18**(5), 763 – 770.

[17] Atkinson, M., Deutscher, M., Kornberg, A., Russell, A., and Moffatt, J. (1969) Enzymatic synthesis of deoxyribonucleic acid. XXXIV. Termination of chain growth by a 2',3'-dideoxyribonucleotide.. *Biochemistry,* **8**(12), 4897 – 4904.

[18] Chidgeavadze, Z., Beabealashvilli, R., Atrazhev, A., Kukhanova, M., Azhayev, A., and Krayevsky, A. (1984) 2', 3'-Dideoxy-3' aminonucleoside 5'-triphosphates are the terminators of DNA synthesis catalyzed by DNA polymerases. *Nucleic Acids Research,* **12**(3), 1671 – 1686.

[19] Shendure, J. and Ji, H. (2008) Next-generation DNA sequencing. *Nature Biotechnology,* **26**(10), 1135 – 1145.

[20] Margulies, M., Egholm, M., Altman, W. E., Attiya, S., Bader, J. S., Bemben, L. A., Berka, J., Braverman, M. S., Chen, Y.-J., Chen, Z., Dewell, S. B., Du, L., Fierro, J. M., Gomes, X. V., Godwin, B. C., He, W., Helgesen, S., Ho, C. H., Irzyk, G. P., Jando, S. C., Alenquer, M. L. I., Jarvie, T. P., Jirage, K. B., Kim, J.-B., Knight, J. R., Lanza, J. R., Leamon, J. H., Lefkowitz, S. M., Lei, M., Li, J., Lohman, K. L., Lu, H., Makhijani, V. B., McDade, K. E., McKenna, M. P., Myers, E. W., Nickerson, E., Nobile, J. R., Plant, R., Puc, B. P., Ronan, M. T., Roth, G. T., Sarkis, G. J., Simons, J. F., Simpson, J. W., Srinivasan, M., Tartaro, K. R., Tomasz, A., Vogt, K. A., Volkmer, G. A., Wang, S. H.,

Wang, Y., Weiner, M. P., Yu, P., Begley, R. F., and Rothberg, J. M. (2005) Genome sequencing in microfabricated high-density picolitre reactors. *Nature,* **437**(7057), 376 – 380.

[21] Ronaghi, M., Karamohamed, S., Pettersson, B., Uhlén, M., and Nyrén, P. (1996) Real-Time DNA Sequencing Using Detection of Pyrophosphate Release. *Analytical Biochemistry,* **242**(1), 84 – 89.

[22] Tawfik, D. S. and Griffiths, A. D. (1998) Man-made cell-like compartments for molecular evolution. *Nature Biotechnology,* **16**(7), 652 – 656.

[23] Cummings, P. J., Ahmed, R., Durocher, J. A., Jessen, A., Vardi, T., and Obom, K. M. (2013) Pyrosequencing for Microbial Identification and Characterization. *Journal of Visualized Experiments : JoVE,* **78**(doi:10.3791/50405).

[24] Voelkerding, K. V., Dames, S. A., and Durtschi, J. D. (2009) Next-Generation Sequencing: From Basic Research to Diagnostics. *Clinical Chemistry,* **55**(4), 641 – 658.

[25] Adessi, C., Matton, G., Ayala, G., Turcatti, G., Mermod, J.-J., Mayer, P., and Kawashima, E. (2000) Solid phase DNA amplification: characterisation of primer attachment and amplification mechanisms. *Nucleic Acids Research,* **28**(20), e87.

[26] Fedurco, M., Romieu, A., Williams, S., Lawrence, I., and Turcatti, G. (2006) BTA, a novel reagent for DNA attachment on glass and efficient generation of solid-phase amplified DNA colonies. *Nucleic Acids Research,* **34**(3), e22.

[27] Chen, F., Dong, M., Ge, M., Zhu, L., Ren, L., Liu, G., and Mu, R. (2013) The History and Advances of Reversible Terminators Used in New Generations of Sequencing Technology. *Genomics, Proteomics & Bioinformatics,* **11**(1), 34 – 40.

[28] Ansorge, W. J. (2009) Next-generation DNA sequencing techniques.. *New Biotechnology,* **25**(4), 195 – 203.

[29] van Dijk, E. L., Auger, H., Jaszczyszyn, Y., and Thermes, C. (2017) Ten years of next-generation sequencing technology. *Trends in Genetics,* **30**(9), 418 – 426.

[30] Mardis, E. R. (2008) Next-Generation DNA Sequencing Methods. *Annual Review of Genomics and Human Genetics,* **9**, 387 – 402.

[31] Valouev, A., Ichikawa, J., Tonthat, T., Stuart, J., Ranade, S., Peckham, H., Zeng, K., Malek, J. A., Costa, G., McKernan, K., Sidow, A., Fire, A., and

Johnson, S. M. (2008) A high-resolution, nucleosome position map of C. elegans reveals a lack of universal sequence-dictated positioning. *Genome Research,* **18**(7), 1051 – 1063.

[32] Reuter, J. A., Spacek, D., and Snyder, M. P. (2015) High-Throughput Sequencing Technologies. *Molecular cell,* **58**(4), 586 – 597.

[33] Rothberg, J. M., Hinz, W., Rearick, T. M., Schultz, J., Mileski, W., Davey, M., Leamon, J. H., Johnson, K., Milgrew, M. J., Edwards, M., Hoon, J., Simons, J. F., Marran, D., Myers, J. W., Davidson, J. F., Branting, A., Nobile, J. R., Puc, B. P., Light, D., Clark, T. A., Huber, M., Branciforte, J. T., Stoner, I. B., Cawley, S. E., Lyons, M., Fu, Y., Homer, N., Sedova, M., Miao, X., Reed, B., Sabina, J., Feierstein, E., Schorn, M., Alanjary, M., Dimalanta, E., Dressman, D., Kasinskas, R., Sokolsky, T., Fidanza, J. A., Namsaraev, E., McKernan, K. J., Williams, A., Roth, G. T., and Bustillo, J. (2011) An integrated semiconductor device enabling non-optical genome sequencing. *Nature,* **475**(7356), 348 – 352.

[34] Eid, J., Fehr, A., Gray, J., Luong, K., Lyle, J., Otto, G., Peluso, P., Rank, D., Baybayan, P., Bettman, B., Bibillo, A., Bjornson, K., Chaudhuri, B., Christians, F., Cicero, R., Clark, S., Dalal, R., deWinter, A., Dixon, J., Foquet, M., Gaertner, A., Hardenbol, P., Heiner, C., Hester, K., Holden, D., Kearns, G., Kong, X., Kuse, R., Lacroix, Y., Lin, S., Lundquist, P., Ma, C., Marks, P., Maxham, M., Murphy, D., Park, I., Pham, T., Phillips, M., Roy, J., Sebra, R., Shen, G., Sorenson, J., Tomaney, A., Travers, K., Trulson, M., Vieceli, J., Wegener, J., Wu, D., Yang, A., Zaccarin, D., Zhao, P., Zhong, F., Korlach, J., and Turner, S. (2009) Real-Time DNA Sequencing from Single Polymerase Molecules. *Science,* **323**(5910), 133–138.

[35] Levene, M. J., Korlach, J., Turner, S. W., Foquet, M., Craighead, H. G., and Webb, W. W. (2003) Zero-Mode Waveguides for Single-Molecule Analysis at High Concentrations. *Science,* **299**(5607), 682–686.

[36] Quail, M. A., Smith, M., Coupland, P., Otto, T. D., Harris, S. R., Connor, T. R., Bertoni, A., Swerdlow, H. P., and Gu, Y. (2012) A tale of three next generation sequencing platforms: comparison of Ion Torrent, Pacific Biosciences and Illumina MiSeq sequencers. *BMC Genomics,* **13**(1), 341.

[37] Wagner, J., Coupland, P., Browne, H. P., Lawley, T. D., Francis, S. C., and Parkhill, J. (2016) Evaluation of PacBio sequencing for full-length bacterial 16S rRNA gene classification. *BMC Microbiology,* **16**(1), 274.

[38] Chaisson, M. J. P., Huddleston, J., Dennis, M. Y., Sudmant, P. H., Malig, M., Hormozdiari, F., Antonacci, F., Surti, U., Sandstrom, R., Boitano, M., Landolin, J. M., Stamatoyannopoulos, J. A., Hunkapiller, M. W., Korlach, J., and Eichler, E. E. (2015) Resolving the complexity of the human genome using single-molecule sequencing. *Nature,* **517**(7536), 608 – 611.

[39] Karlsson, E., Lärkeryd, A., Sjödin, A., Forsman, M., and Stenberg, P. (2015) Scaffolding of a bacterial genome using MinION nanopore sequencing. *Scientific Reports,* **5**(doi:10.1038/srep11996).

[40] Schneider, G. F. and Dekker, C. (2012) DNA sequencing with nanopores. *Nature Biotechnology,* **30**(4), 326 – 328.

[41] Ashton, P. M., Nair, S., Dallman, T., Rubino, S., Rabsch, W., Mwaigwisya, S., Wain, J., and O'Grady, J. (2015) MinION nanopore sequencing identifies the position and structure of a bacterial antibiotic resistance island. *Nature Biotechnology,* **33**(3), 296 – 300.

[42] Laver, T., Harrison, J., O'Neill, P., Moore, K., Farbos, A., Paszkiewicz, K., and Studholme, D. (2015) Assessing the performance of the Oxford Nanopore Technologies MinION. *Biomolecular Detection and Quantification,* **3**(doi:10.1016/j.bdq.2015.02.001).

[43] Nagalakshmi, U., Waern, K., and Snyder, M. RNA-Seq: A Method for Comprehensive Transcriptome Analysis isbn:9780471142720, John Wiley & Sons, Inc. (2001).

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

[45] Nookaew, I., Papini, M., Pornputtapong, N., Scalcinati, G., Fagerberg, L., UhlÃĺn, M., and Nielsen, J. (2012) A comprehensive comparison of RNA-Seq-based transcriptome analysis from reads to differential gene expression and cross-comparison with microarrays: a case study in Saccharomyces cerevisiae. *Nucleic Acids Research,* **40**(20), 10084.

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

[47] Sorek, R. and Cossart, P. (2010) Prokaryotic transcriptomics: a new view on regulation, physiology and pathogenicity. *Nature Reviews Genetics,* **11**(1), 9 – 16.

[48] Cech, T. R. and Steitz, J. A. (2017) The Noncoding RNA Revolution-Trashing Old Rules to Forge New Ones. *Cell,* **157**(1), 77 – 94.

[49] Chen, G., Wang, C., and Shi, T. (2011) Overview of available methods for diverse RNA-Seq data analyses. *Science China Life Sciences,* **54**(12), 1121–1128.

[50] Soneson, C. and Delorenzi, M. (2013) A comparison of methods for differential expression analysis of RNA-seq data. *BMC Bioinformatics,* **14**(1), 91.

[51] Dodt, M., Roehr, J. T., Ahmed, R., and Dieterich, C. (2012) FLEXBAR - Flexible Barcode and Adapter Processing for Next-Generation Sequencing Platforms. *Biology,* **1**(3), 895–905.

[52] Mamanova, L., Coffey, A. J., Scott, C. E., Kozarewa, I., Turner, E. H., Kumar, A., Howard, E., Shendure, J., and Turner, D. J. (2010) Target-enrichment strategies for next-generation sequencing. *Nature Methods,* **7**(2), 111 – 118.

[53] Sharma, C. M., Hoffmann, S., Darfeuille, F., Reignier, J., Findeisz, S., Sittka, A., Chabas, S., Reiche, K., Hackermuller, J., Reinhardt, R., Stadler, P. F., and Vogel, J. (2010) The primary transcriptome of the major human pathogen Helicobacter pylori. *Nature,* **464**(7286), 250 – 255.

[54] Sharma, C. M. and Vogel, J. (2014) Differential RNA-seq: the approach behind and the biological insight gained. *Current Opinion in Microbiology,* **19**, 97 – 105.

[55] Babski, J., Haas, K. A., Näther-Schindler, D., Pfeiffer, F., Förstner, K. U., Hammelmann, M., Hilker, R., Becker, A., Sharma, C. M., Marchfelder, A., and Soppa, J. (2016) Genome-wide identification of transcriptional start sites in the haloarchaeon Haloferax volcanii based on differential RNA-Seq (dRNA-Seq). *BMC Genomics,* **17**(doi:10.1186/s12864-016-2920-y).

[56] Westermann, A. J., Gorski, S. A., and Vogel, J. (2012) Dual RNA-seq of pathogen and host. *Nature Reviews Microbiology,* **10**(9), 618 – 630.

[57] Saliba, A.-E., Santos, S. C., and Vogel, J. (2017) New RNA-seq approaches for the study of bacterial pathogens. *Current Opinion in Microbiology,* **35**, 78 – 87.

[58] Hennig, J. and Sattler, M. (2015) Deciphering the protein-RNA recognition code: Combining large-scale quantitative methods with structural biology. *BioEssays,* **37**(8), 899–908.

[59] Licatalosi, D. D., Mele, A., Fak, J. J., Ule, J., Kayikci, M., Chi, S. W., Clark, T. A., Schweitzer, A. C., Blume, J. E., Wang, X., Darnell, J. C., and Darnell, R. B. (2008) HITS-CLIP yields genome-wide insights into brain alternative RNA processing. *Nature,* **456**(7221), 464 – 469.

[60] Ule, J., Jensen, K., Mele, A., and Darnell, R. B. (2005) CLIP: A method for identifying protein-RNA interaction sites in living cells. *Methods,* **37**(4), 376 – 386 Post-transcriptional Regulation of Gene Expression.

[61] Stajich, J. E. and Lapp, H. (2006) Open source tools and toolkits for bioinformatics: significance, and where are we?. *Briefings in Bioinformatics,* **7**(3), 287.

[62] Leipiz, J. (2016) A review of bioinformatic pipeline frameworks. *Briefings in Bioinformatics,* (doi:/10.1093/bib/bbw020).

[63] Meyer, B. (2008) Seven Principles of Software Testing. *Computer,* **41**(0018-9162), 99–101.

[64] Ekmekci, B., McAnany, C. E., and Mura, C. (06, 2016) An Introduction to Programming for Bioscientists: A Python-Based Primer. *PLOS Computational Biology,* **12**(6), 1–43.

[65] Lipman, D. and Pearson, W. (1985) Rapid and sensitive protein similarity searches. *Science,* **227**(4693), 1435–1441.

[66] Cock, P. J. A., Fields, C. J., Goto, N., Heuer, M. L., and Rice, P. M. (2009) The Sanger FASTQ file format for sequences with quality scores, and the Solexa/Illumina FASTQ variants. *Nucleic Acids Research,* **38**(6), 1767 – 1771.

[67] Williams, C. R., Baccarella, A., Parrish, J. Z., and Kim, C. C. (2016) Trimming of sequence reads alters RNA-Seq gene expression estimates. *BMC Bioinformatics,* **17**(103), 1 – 13.

[68] Conesa, A., Madrigal, P., Tarazona, S., Gomez-Cabrero, D., Cervera, A., McPherson, A., Szcześniak, M. W., Gaffney, D. J., Elo, L. L., Zhang, X., and Mortazavi, A. (2016) A survey of best practices for RNA-seq data analysis. *Genome Biology,* **17**(1), 13.

[69] Grabherr, M. G., Haas, B. J., Yassour, M., Levin, J. Z., Thompson, D. A., Amit, I., Adiconis, X., Fan, L., Raychowdhury, R., Zeng, Q., Chen, Z., Mauceli, E., Hacohen, N., Gnirke, A., Rhind, N., di Palma, F., Birren, B. W., Nusbaum,

C., Lindblad-Toh, K., Friedman, N., and Regev, A. (2011) Full-length transcriptome assembly from RNA-Seq data without a reference genome. *Nature Biotechnology,* **29**(7), 644 – 652.

[70] Pop, M., Phillippy, A., Delcher, A. L., and Salzberg, S. L. (2004) Comparative genome assembly. *Briefings in Bioinformatics,* **5**(3), 237.

[71] Zhao, S. (2014) Assessment of the Impact of Using a Reference Transcriptome in Mapping Short RNA-Seq Reads. *PLoS ONE,* **9**(7), e101374.

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

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

[74] Baruzzo, G., Hayer, K. E., Kim, E. J., Di Camillo, B., Fitzgerald, G. A., and Grant, G. R. (2017) Simulation-based comprehensive benchmarking of RNA-seq aligners. *Nature Methods,* **14**(2), 135 – 139.

[75] Borozan, I., Watt, S. N., and Ferretti, V. (10, 2013) Evaluation of Alignment Algorithms for Discovery and Identification of Pathogens Using RNA-Seq. *PLOS ONE,* **8**(10), 1–17.

[76] Engstrom, P. G., Steijger, T., Sipos, B., Grant, G. R., Kahles, A., Consortium, T. R., Ratsch, G., Goldman, N., Hubbard, T. J., Harrow, J., Guigo, R., and Bertone, P. (2013) Systematic evaluation of spliced alignment programs for RNA-seq data. *Nature Methods,* **10**(12), 1185 – 1191.

[77] Li, H., Handsaker, B., Wysoker, A., Fennell, T., Ruan, J., Homer, N., Marth, G., Abecasis, G., Durbin, R., and (2009) The Sequence Alignment/Map format and SAMtools. *Bioinformatics,* **25**(16), 2078.

[78] Huy Hoang, D. and Sung, W.-K. (2014) CWig: compressed representation of Wiggle/BedGraph format. *Bioinformatics,* **30**(18), 2543.

[79] Dillies, M.-A., Rau, A., Aubert, J., Hennequet-Antier, C., Jeanmougin, M., Servant, N., Keime, C., Marot, G., Castel, D., Estelle, J., Guernec, G., Jagla, B., Jouneau, L., Laloë, D., Le Gall, C., Schaëffer, B., Le Crom, S., Guedj, M., and Jaffrézic, F. (2013) A comprehensive evaluation of normalization methods for Illumina high-throughput RNA sequencing data analysis. *Briefings in Bioinformatics,* **14**(6), 671.

[80] Rapaport, F., Khanin, R., Liang, Y., Pirun, M., Krek, A., Zumbo, P., Mason, C. E., Socci, N. D., and Betel, D. (2013) Comprehensive evaluation of differential gene expression analysis methods for RNA-seq data. *Genome Biology,* **14**(9), 3158.

[81] Tao, Y., Liu, Y., Friedman, C., and Lussier, Y. A. (2004) Information Visualization Techniques in Bioinformatics during the Postgenomic Era. *Drug discovery today. Biosilico,* **2**(6), 237 – 245.

[82] Loraine, A. E., Blakley, I. C., Jagadeesan, S., Harper, J., Miller, G., and Firon, N. Analysis and Visualization of RNA-Seq Expression Data Using RStudio, Bioconductor, and Integrated Genome Browser pp. 481–501 isbn = 9781493924448 Springer New York (2015).

[83] Oberg, A. L., Bot, B. M., Grill, D. E., Poland, G. A., and Therneau, T. M. (2012) Technical and biological variance structure in mRNA-Seq data: life in the real world. *BMC Genomics,* **13**(1), 304.

[84] Blakesley, R. E., Mazumdar, S., Dew, M. A., Houck, P. R., Tang, G., Reynolds, C. F., and Butters, M. A. (12, 2013) Comparisons of Methods for Multiple Hypothesis Testing in Neuropsychological Research. *Neuropsychology,* **23**(2), 255 – 264.

[85] Zhang, X. D. and Zhang, Z. (2013) displayHTS: a R package for displaying data and results from high-throughput screening experiments. *Bioinformatics,* **29**(6), 794.

[86] Förstner, K. U., Vogel, J., and Sharma, C. M. (2014) READemption - a tool for the computational analysis of deep-sequencingâĂŞbased transcriptome data. *Bioinformatics,* **30**(23), 3421.

[87] Hoffmann, S., Otto, C., Kurtz, S., Sharma, C. M., Khaitovich, P., Vogel, J., Stadler, P. F., and Hackermüller, J. (09, 2009) Fast Mapping of Short Sequences with Mismatches, Insertions and Deletions Using Index Structures. *PLOS Computational Biology,* **5**(9), 1–10.

[88] Otto, C., Stadler, P. F., and Hoffmann, S. (2014) Lacking alignments? The next-generation sequencing mapper segemehl revisited. *Bioinformatics,* **30**(13), 1837.

[89] Introduction JSON format. `http://json.org/`.

[90] Nicol, J. W., Helt, G. A., Blanchard, Jr., S. G., Raja, A., and Loraine, A. E. (2009) The Integrated Genome Browser: free software for distribution and exploration of genome-scale datasets. *Bioinformatics,* **25**(20), 2730.

[91] Thomason, M. K., Bischler, T., Eisenbart, S. K., Förstner, K. U., Zhang, A., Herbig, A., Nieselt, K., Sharma, C. M., and Storz, G. (2014) Global Transcriptional Start Site Mapping Using Differential RNA Sequencing Reveals Novel Antisense RNAs in *Escherichia coli. Journal of Bacteriology,* **197**(1), 18 – 28.

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

[93] 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), 550.

[94] Gayen, A. K. (1951) The Frequency Distribution of the Product-Moment Correlation Coefficient in Random Samples of Any Size Drawn from Non-Normal Universes. *Biometrika,* **38**(1/2), 219–247.

[95] Documentation 'READemption' (version 0.4.3). `https://pythonhosted.org/READemption/`.

[96] Main page 'python'. `http://python.org`.

[97] Documentation 'json' (version 19.2.). `https://docs.python.org/3/library/json.html#module-json`.

[98] Documentation 'pandas' (version 0.19.2). `http://pandas.pydata.org/pandas-docs/version/0.19.2/whatsnew.html`.

[99] Documentation 'csv' (version 14.1.). `https://docs.python.org/3/library/csv.html`.

[100] Documentation 'pysam' (version 0.10.0). `http://pysam.readthedocs.io/en/latest/api.html`.

[101] Cock, P. J. A., Antao, T., Chang, J. T., Chapman, B. A., Cox, C. J., Dalke, A., Friedberg, I., Hamelryck, T., Kauff, F., Wilczynski, B., and de Hoon, M. J. L. (2009) Biopython: freely available Python tools for computational molecular biology and bioinformatics. *Bioinformatics,* **25**(11), 1422.

[102] Oliphant, T. E. (May, 2007) Python for Scientific Computing. *Computing in Science Engineering,* **9**(3), 10–20.

[103] Documentation 'lzma' (version 13.4.). `https://docs.python.org/3/library/lzma.html`.

[104] Hunter, J. D. (May, 2007) Matplotlib: A 2D Graphics Environment. *Computing in Science Engineering,* **9**(3), 90–95.

[105] Main page 'bokeh' (version 0.12.4). `http://bokeh.pydata.org/en/latest/`.

[106] Documentation 'unittest' (version 26.4.). `https://docs.python.org/3/library/unittest.html`.

[107] Documentation 'pytest'(version 3.0). `http://doc.pytest.org/en/latest/contents.html`.

[108] Martin, M. (2011) Cutadapt removes adapter sequences from high-throughput sequencing reads. *EMBnet.journal,* **17**(doi:dx.doi.org/10.14806/ej.17.1.200).

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

[110] GitHub Repository of READemption from Konrad Förstner. `https://github.com/konrad/READemption`.

[111] Documentation 'cutadapt' (version 1.12). `https://cutadapt.readthedocs.io/en/stable/guide.html`.

[112] Chen, C., Khaleel, S. S., Huang, H., and Wu, C. H. (2014) Software for pre-processing Illumina next-generation sequencing short read sequences. *Source Code for Biology and Medicine,* **9**(1), 8.

[113] Del Fabbro, C., Scalabrin, S., Morgante, M., and Giorgi, F. M. (12, 2013) An Extensive Evaluation of Read Trimming Effects on Illumina NGS Data Analysis. *PLOS ONE,* **12**(doi:10.1371/journal.pone.0085024).

[114] Manber, U. and Myers, G. (1993) Suffix Arrays: A New Method for On-Line String Searches. *SIAM Journal on Computing,* **22**(5), 935–948.

[115] Documentation 'logging' (version 16.6.). `https://docs.python.org/3/library/logging.html`.

[116] Documentation 'luigi' (version 2.6.1). `https://luigi.readthedocs.io/en/stable/index.html`.

[117] Wang, T., Xiao, G., Chu, Y., Zhang, M. Q., Corey, D. R., and Xie, Y. (2015) Design and bioinformatics analysis of genome-wide CLIP experiments. *Nucleic Acids Research,* **43**(11), 5263 – 5274.

Eidesstattliche Erklärung

ERKLÄRUNG gemäß ASPO 2009 §23 Abs. 10 bzw. ASPO2015 §26 Abs. 11

Hiermit versichere ich, dass ich die vorliegende Arbeit selbstständig verfasst, keine anderen als die angegebenen Quellen und Hilfsmittel benutzt und die Arbeit bisher oder gleichzeitig keiner anderen Prüfungsbehörde unter Erlangung eines akademischen Grades vorgelegt habe.

Ort, Datum Unterschrift

Danksagung

Ein besonderer Dank gilt Konrad. Du hast es mir mit viel Geduld und einer schier endlosen Fachkenntnis überhaupt ermöglicht, in diesem Bereich Fuss zu fassen. Von dir kann man sich nicht nur fachlich eine Scheibe abschneiden. Ein großer Dank gilt auch Iotta, Malvika, Marga, Richa, Silvia, Sung-Huan, Thorsten und Till. Durch euch habe ich mich innerhalb der Arbeitsgruppe stets wohl gefühlt. Sung-Huan und Thorsten, ihr habt mir immer bei meinem Code weitergeholfen! Django, wäre einer von uns eine Frau, wären wir wohl verheiratet. Zu guter Letzt möchte ich meinen Eltern, Geschwistern und der ganzen restlichen Familie danken. In den verschiedenen Phasen meines Lebens hat mir immer mindestens einer von euch mit Rat und Tat zur Seite gestanden. Das ist keine Selbstverständlichkeit und ermöglicht mir erst diese Danksagung.