Chapter 3
Transcriptome Analysis Using RNA-seq and scRNA-seq

Waldeyr Mendes Cordeiro Silva, Fabián Andrés Hurtado, Kelly Simi, Pedro Henrique Aragão Barros, Dimitri Sokolowskei, Ildinete Silva-Pereira, Maria Emilia Walter, and Marcelo Brigido

3.1 High-Throughput Sequencing Techniques

Since Sanger's technology in the 1970s, DNA sequencing has been continuously improved regarding both throughput and low cost. Next-generation sequencing (NGS), also called high-throughput or deep sequencing, constitutes a new breakthrough in increasing research power, a revolutionary advancement in molecular biology knowledge. An increasing number of biological questions may be addressed by NGS technologies, which provide a much larger comprehensive survey compared to the Sanger method, and under a system biology perspective. Transcriptomics has been particularly benefited by the use of these new technologies, also called RNA-seq, allowing a complete characterization of the whole transcriptome at both gene (Kvam et al. 2012) and exon (Anders et al. 2012) levels,

W. M. C. Silva
Laboratório de Biologia Molecular, CEL/IB, Universidade de Brasília, Brasília, DF, Brazil

Instituto Federal de Educação, Ciência e Tecnologia de Goiás, Formosa, GO, Brazil

F. A. Hurtado · P. H. A. Barros · D. Sokolowskei
Laboratório de Biologia Molecular, CEL/IB, Universidade de Brasília, Brasília, DF, Brazil

Graduation program in Molecular Pathology/FM/UnB, Universidade de Brasília, Brasília, DF, Brazil

K. Simi
UniCEUB, Centro Universitario de Brasília, Brasília, DF, Brazil

I. Silva-Pereira · M. Brigido (✉)
Laboratório de Biologia Molecular, CEL/IB, Universidade de Brasília, Brasília, DF, Brazil
e-mail: brigido@unb.br

M. E. Walter
Departamento de Ciência da Computação, Instituto de Ciências Exatas, Universidade de Brasília, Brasília, DF, Brazil

G. A. Passos (ed.), *Transcriptomics in Health and Disease*,
https://doi.org/10.1007/978-3-030-87821-4_3

and with an additional ability to identify rare transcripts, new genes, novel splicing junctions, and gene fusions (Wang et al. 2009; Katz et al. 2010; Van Verk et al. 2013). More recently, single-cell sequencing had become a feasible task allowing a deeper and systemic view of individual cell's transcriptomes.

This chapter first addresses a brief overview of sequencing techniques and the most common next-generation platforms and computational methods for RNA-seq data analysis. Then, we present two case studies to assess the capabilities of RNA-seq in addressing important biological issues.

3.1.1 Sanger's Sequencing Technology

In 1977, Frederick Sanger and colleagues (1977) developed the DNA sequencing method, which in 2001 allowed the first human genome draft (Lander et al. 2001). This method, called dideoxy chain-termination or simply the Sanger method, is based on special nucleotide molecules (called ddTNP), lacking a 3′-OH at the deoxyribose, which blocks the DNA elongation. These special nucleotides are mixed in lower concentrations to the regular nucleotides and used as reagents for DNA polymerase reaction. Therefore, with the polymer synthesis stopped by the ddNTP's inclusion, the last nucleotide can be determined. Each of the four ddNTPs was added separately in four different reactions. In the beginning, one of the regular nucleotides, most commonly dATP or dCTP, was radioactively labeled (e.g., 32P or 35S) to achieve the radioactive signal. Usually, polyacrylamide gel electrophoresis was used to separate the DNA molecules, which diverged in length by a single nucleotide. Then the gel was dried and exposed to X-ray film.

An important modification of the method was substituting the radioactive label with a fluorescent dye (Smith et al. 1986). Each distinct wavelength produced by the fluorescent dyes linked to dideoxynucleotides corresponds to a different nucleotide, with the four sequencing reactions performed in the same tube. With the Sanger sequencing method's automation, the performance reached up to 96 different reactions running in parallel capillary gel electrophoresis (Marsh et al. 1997), which is considered the first-generation technology. At the top of the technology, 384 samples could be sequenced at once in a single multi-well plate. The Sanger method's main sequencing devices are ABI (Applied Biosystems) and MegaBACE (GE Healthcare Life Sciences).

3.1.2 Next Generation Sequencing

Regulatory mechanisms and gene expression profiles have been widely investigated toward the elucidation of several essential cellular processes. Hybridization-based technology, e.g., microarray, has been beneficial for determining global gene expression. However, the high background levels due to cross-hybridization, a limited range of quantification, and a restricted detection of known genes are bottlenecks for large-scale use of this technique (Shendure 2008). RNA-seq allows a genome-scale transcriptome analysis, including novel genes and splice variants, with a wide range of quantification and reduced sequencing costs (Wang et al. 2009; Soon et al. 2013). These advantages make RNA-seq a better and attractive solution for whole-genome transcriptome analysis of several organisms, even for those with no sequenced reference genomes.

Nowadays, the most commonly used NGS platforms for RNA-seq research are Illumina, PacBio, and Nanopore. These and other novel platforms are rapidly becoming more popular as they profile short and longer reads at a reasonable price per base. The substitution of older NGS technology is fast and pioneer methods, such as pyrosequencing, are nowadays wholly abandoned. A comparison of current NGS technologies is shown in Table 3.1.

The enormous amounts of data generated by NGS create new challenges to the downstream bioinformatics analysis, which has to handle large sequence files while searching for comprehensive and useful biological information, discussed later in this chapter.

3.1.3 Illumina Sequencing

Illumina sequencing uses a reversible dye-terminator technique that adds a single nucleotide to the DNA template in each cycle (Bentley et al. 2008). This system was initially developed in 2007 by Solexa and was subsequently acquired by Illumina, Inc. Illumina is widely used in several transcriptome studies since it reaches the deepest depth among NGS technologies, despite its small sequence size (150–300 bp).

Illumina sequencing is based on sequencing-by-synthesis. Sequencing is performed in a solid slide covered by adaptors complementary to those added to the fragmented DNA sequences (Metzker 2010). This procedure, called bridge PCR, consists of amplifying bent DNA sequences attached by both ends to the solid surface (Fig. 3.1a). By the end of the clonal amplification, clusters of identical DNA sequences (Polonies) will be formed to amplify the fluorescence signals. In each round, one single nucleotide is added to the single-strand template sequences followed by fluorescence detection by a high-sensitivity CCD camera (Fig. 3.1b).

Table 3.1 Comparison of next-generation sequencing technologies

	ABI 3730xl (Sanger)	Illumina	PacBio	Nanopore
Read length (bp)	900	75–300	5000–60000	500–2300000
Cost (US$/Mb)	500	0.01–0.063	0.013–0.933	0.021–2
Output data/run	2,88 Mb	~120 Gb	2–160 Gb	10–300 Gb
Time run (hours)	3	12–44	Up to 4 h	0.017–72

Data from Amarasinghe et al. (2020); Logsdon et al. (2020)

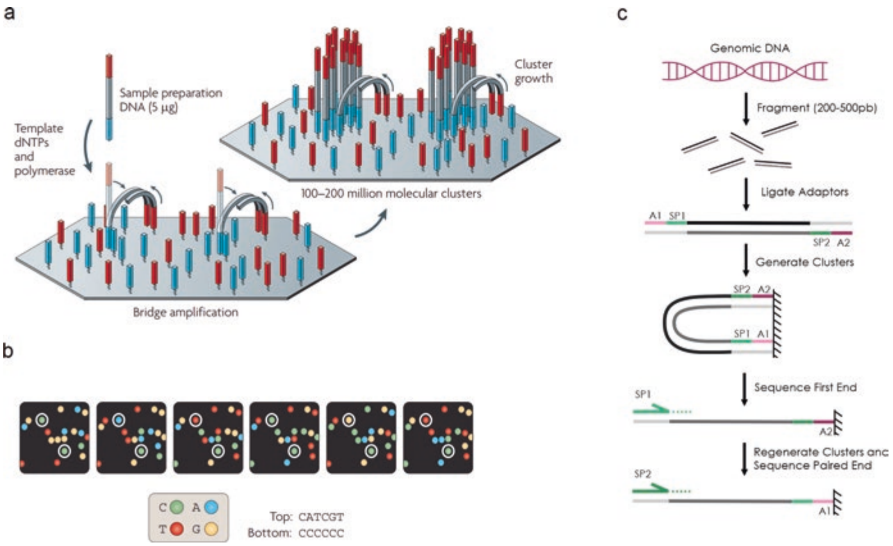

Fig. 3.1 The Illumina sequencing technology. (**a**) Two basic steps encompass an initial priming and extending of the single-stranded, single-molecule template, and bridge amplification of the immobilized template in a solid device with immediately adjacent primers to form clusters; (**b**) In the images, the sequencing data is highlighted from two sequence clusters; (**c**) Paired-end sequencing by which reads are generated from both template strand. “A” block indicates the device-ligation adaptors and “SP,” sequencing primers. (Source: Metzker (2010) and http://www.illumina.com/)

As in Sanger’s technology, different fluorophore molecules are attached to each nucleotide; however, only one nucleotide is incorporated in each cycle. The fluorescence emission releases the 3′OH of the recently added nucleotide, allowing it to receive new monomers in the subsequent sequencing cycle.

Single-end sequencing, i.e., reads generated from a single-end adaptor, is being replaced by the paired-end sequencing since the accuracy of downstream analysis is greater with a fair price. Paired-end reads are produced from the adaptor priming sites in both template sequence ends, the second adaptor primer being used in a subsequent sequencing run (Fig. 3.1c).

3.1.4 Pacific BioSciences Sequencing

Single-molecule real-time (SMRT) sequencing was devised by Pacific BioSciences (PacBio) in 2009, and it is also called PacBio sequencing (Eid et al. 2009). This platform uses a single DNA polymerase attached to the bottom of a picolitre well – zero-mode waveguides (ZMW) – which replicates a single-molecule template per well to produce a signal for light detection in the smallest volume. In this method, the template is capped by hairpin adapters at both ends of the double-stranded DNA molecule, forming a single-stranded circular DNA (called a SMRTbell). Consequently, the polymerase repeatedly passes over the circular template and sequencing it multiple times, resulting in long read lengths and, thus, providing higher accuracy (Rhoads and Au 2015). The PacBio platform enables simultaneous analysis of millions of wells per chip in a single run, providing long read lengths to up to 60 kb (with average read lengths of 20 kb) (Nakano et al. 2017).

Overall, this technology is considered highly accurate and robust, even as its first sequencers have some drawbacks that narrow down its application. For instance, the limited high-throughput, higher cost, and error rate compared with those of second-next generation sequencing (SGS) technologies (Kanzi et al. 2020; Wang et al. 2020). However, in 2019, PacBio launched the Sequel II System, which asserts improvements in the sequencing to deal with these limitations, generating highly accurate (99.9%) individual long reads up to 25 kb (HiFi reads) and reduces the costs and time of the project, in comparison with its prior versions (Wenger et al. 2019; Logsdon et al. 2020). These HiFi reads are generated by using the circular consensus sequencing (CCS) due to continuous circular sequencing (Wenger et al. 2019; Pereira et al. 2020).

For transcriptomic analysis, the SMRT isoform sequencing (Iso-Seq) from PacBio increased the read length compared to other SGS technologies. This platform achieves full-length transcripts sequencing, improving the analysis in different applications, including gene annotation, isoform identification, fusion transcripts identification, and long non-coding RNA discovery (Weirather et al. 2015; Nattestad et al. 2018; Wang et al. 2019; Zhang et al. 2020a; Hu et al. 2020).

3.1.5 Nanopore MinION Sequencing

The long-read-length sequencer MinION, the first nanopore sequencer device, was announced by Oxford Nanopore Technologies (ONT) in 2012 as a portable, compact, real-time sequencing controlled by a laptop computer device (Deamer et al. 2016). Since then, new nanopore platforms have quickly emerged, such as PromethION, which offers a greater scale of sequencing, and SmidgION, the smallest sequencing platform designed for use with smartphones or other mobile devices.

After library preparation, each strand is attached to adapters. The adaptors bind to a protein motor that guides the sequence to the protein pore, which processes it. Beginning at the 5′-end, the DNA or RNA polymer passes through the pore controlled by the motor protein, which unzips dsDNA and translocates a single strand sequence (Fig. 3.2). The translocated strand modulates the ion current flow through the pore membrane (Ip et al. 2015). The variation of the electrochemical current promoted by each different nucleotide is measured by a sensor and enables identification by different signal patterns. The resultant signals are stored in a FAST5 format file and can be finally used for base-calling, a process in which the nucleotides are predicted from the Raw signals and transferred to a FASTQ file.

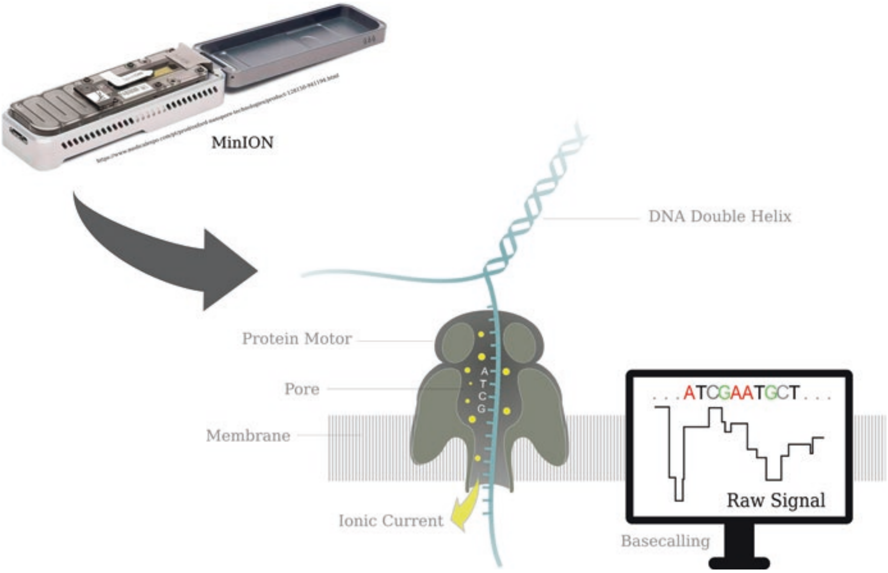

Fig. 3.2 Schematic view of the nanopore sequencer. MinION device process double DNA helix. First, the protein motor unzips DNA passing a single strand through the pore. The movement of the single strand promotes an ionic current flow that is measured and converted to nucleotides data by the base calling analysis

Base-calling can be performed using only information from one strand (1D) or two strands (2D) for consensus, with information from both strands resulting in better base prediction (Lu et al. 2016). Currently use of neural networks in base-calling reached an accuracy interval between 85% and 95% with the detection of signal patterns (Zhang et al. 2020b).

Although sequencing full-length reads allows improvement of isoforms identification and discovery in transcriptome sequencing, it deals with high error rates (Kovaka et al. 2019). To reduce error rates before analysis, nanopore correcting errors can be made by a hybrid error correction strategy. This strategy uses high accuracy short reads to correct long-reads, self-correction methods that rely only on long-reads, or reference-based methods that use a reference genome for error correction (Zhao et al. 2019).

3.2 Bioinformatics Pipelines for Transcriptome Projects

Illumina sequencing is the most used technique in transcriptome studies, since the number of sequenced reads (named raw data) allows to find out virtually the complete set of expressed genes (transcripts). However, longer reads allow a more precise definition of the transcripts. In both cases, the metaphor for reconstructing the transcripts is like mounting a puzzle, where the pieces (the reads) have to be assembled (relative to a reference genome or not) to obtain the picture (transcripts in a transcriptome). After this, different analyses can be performed on these reconstructed transcripts, e.g., quantitative and differential expression. In a transcriptomic project, the tasks of reconstructing transcripts and performing biological analyses are performed by bioinformatics pipelines, discussed next.

3.2.1 Pipelines

A bioinformatics pipeline or workflow is a computational system composed of a sequence of programs sequentially executed. The output data from one software is the input data for the following software (Wercelens et al. 2019). In general, transcriptome bioinformatics pipelines have the following steps, which can be combined according to the raw input data and the objectives of each project:

- Quality control of raw data: This initial step allows visualization, analysis, and filtering (cleaning) the data. Usually, this process takes two sub-steps as follows: clipping and trimming. In the clipping step, adapters (primers) attached to the ends of the sequenced reads (or even the whole read) are removed. In the trimming step, low-quality sequences in the reads are filtered. The filtering guarantees a reliable dataset of quality reads to be used in the following phases of the pipeline.
- Assembly: in the absence of a reference genome or transcriptome, it is necessary to assembly one. For that, overlapping reads (the end of a read is similar to the beginning of another read) are joined in groups of reads (called contig), allowing to construct of one larger sequence (called consensus), which is a predicted (fragment of) transcript. The complete set of transcripts is the predicted transcriptome (Fig. 3.3).
- Mapping: The filtered reads can be aligned to the transcriptome's reference genome to find the actively expressed exons or transcripts. The amount of reads mapping to a single exon/transcript is proportional to its expression.
- Analysis: The whole set of (fragments of) transcripts obtained from the mapping or the assembling step allows to obtain relevant biological information, e.g.

  (a) quantitative analysis: among others, coverage analysis shows the abundance of genes expressed in one RNA-seq sample, more precisely, the number of reads mapped in a certain region of the chromosome.
  (b) differential expression: allows to analyze the differences and variability of gene expression between samples along distinct genomic regions.
  (c) annotation: assigns a biological function to each transcript.

Designing a particular pipeline mainly depends on the transcriptome project's objectives and other information, such as the sequencing platform employed (since the sequencing techniques may cause specific errors in the raw data). It also depends

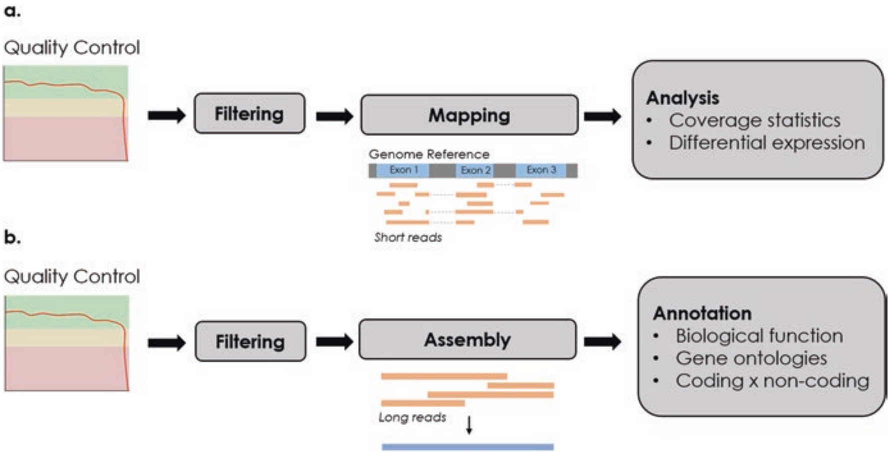

Fig. 3.3 Examples of pipelines for transcriptome analysis: (**a**) Pipelines for short reads, with a well-characterized reference genome, and two types of analyses – coverage statistics and differential expression. (**b**) Pipeline for longer reads, with no reference genome, and annotation (biological function, gene categories, and ontologies)

on the availability of a reference genome or transcriptome in the mapping step and the analysis step's accuracy and biases. Two generic bioinformatics pipelines for transcriptomes are discussed next.

Pipeline 1 The organisms of interest have already been sequenced, preferably with high coverage, well-annotated genes, and other relevant biological characteristics. The reads are usually short (about 150–300 bp), typically produced by Illumina sequencing platforms. This pipeline would be composed of a minimum of three steps (Fig. 3.3a): quality control, mapping, and quantitative analysis.

Pipeline 2 The organism of interest has not been sequenced before. The reads are usually long (up to 40 kb), heavily produced by the PacBio sequencing platform. This pipeline would be composed of a minimum of three steps (Fig. 3.3b): quality control, assembly, and annotation. The assembly phase constructs one consensus sequence for each group of reads presenting similar extremities. This approach heavily depends on sequencing quality, and the multiplatform approach improves the final assembled transcriptome. Finally, the annotation phase assigns biological functions to the consensus sequences.

A bioinformatics pipeline is usually implemented using command lines (e.g., GNU/Linux terminal) mainly because it is a fast, relatively simple, and reliable way to control and manipulate large amounts of datasets. Programming languages such as Shell Script, Python, R, and Perl might also help implement a pipeline and resolve minor tasks by scripting. The pipeline's files/data can be organized in directories or database management systems, relational databases (e.g., MySQL, Oracle), or NoSQL databases (e.g., MongoDB, Neo4J) to store, retrieve, and manage the data.

Most software used in pipelines are free, open-source, publicly available, and some of the most common ones are described next.

Frameworks to manage workflows are also available, such as Snakemake (Köster and Rahmann 2012) and Common Workflow Language (CWL) (https://www.commonwl.org/v1.0). They provide a reliable way to standardize the syntax and semantics for program evoking and create robust and reproducible workflows.

3.2.2 Bioinformatics Software

3.2.2.1 Software for Quality Control

The overall quality of the output sequencing data must be assessed to eliminate bad quality, poorly sequenced, or ambiguous raw data that could negatively impact further analysis. Thus, filtering (or cleaning) strategies capable of clipping and trimming are essential to guarantee the reliability of transcriptomics data and ensure obtaining relevant and trustworthy biological information. The sequenced reads are stored using FASTQ format, gathering the nucleotides sequences of each read and their corresponding quality scores.

Some tools are used to assess and visualize the overall quality of data, such as FastQC (https://www.bioinformatics.babraham.ac.uk/projects/fastqc), a popular java-based quality control check program. Other tools to perform filtering steps like FASTX-Toolkit (http://hannonlabcshledu/fastx_toolkit) provide options for performing both clipping and trimming. Other commonly used tools are Cutadapt (Martin 2011) for clipping, PRINSEQ (Schmieder and Edwards 2011), and Trimmomatic (Bolger et al. 2014) for trimming. Fastp (Chen et al. 2018) is an ultra-fast all-in-one quality control, and data-filtering tool that can be an alternative to multiple and insufficiently fast software for quality control. They all present several options, such as minimum size for a read, minimum quality score, and polyadenylation removal.

3.2.2.2 Software for Mapping

The mapping phase's main objective is to find where each filtered short read corresponds in a reference genome/transcriptome (Fig. 3.4).

There are many programs capable of performing the mapping process. In general, these software are computationally intensive (to process and store data), and mapping techniques use indices to accelerate the search procedure and reduce the memory cost associated with finding the location of reads to the reference genome.

Bowtie (Langmead et al. 2009) is a fast short aligner that tolerates a small number of mismatches. Bowtie first concatenates all the reference genome in one single string and performs the Burrows-Wheeler transformation (BWT) to generate one index to this reference genome. Next, character by character of each read is mapped

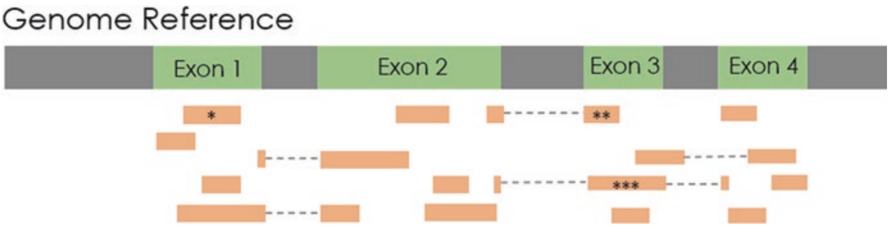

Fig. 3.4 Short reads mapped to a reference genome. Reads are aligned to a reference genome and the accumulation of data brings in evidence expressed exons and splice junctions

until the entire sequence is aligned. If a read cannot find a perfect alignment location, the program backtracks one character, substitutes this character, and the process is repeated until the alignment is completed. The maximum number of character substitutions is a parameter in Bowtie. The rapid improvement of throughput and increase of read length of sequencing technologies required the development of Bowtie2 (Langmead and Salzberg 2012), a gapped supported alignment tool that performs a faster and more sensitive mapping for reads longer than 50 bp.

TopHat (Trapnell et al. 2009) can identify exons splicing sites by mapping RNA-seq reads against a reference genome. First, the Bowtie mapping program is employed to map the short unspliced reads to the reference genome. The reads that are not initially mapped are not filtered out but are just set apart. After the main alignment, each unmapped reads are split into shorter fragments and then aligned individually and independently to identify splice junctions between exons. TopHat2 (Kim et al. 2013) is an updated version of TopHat with an overall accuracy improvement and better alignment procedure.

The Spliced Transcripts Alignment to a Reference (STAR) (Dobin et al. 2013) represents a significant mapping alignment algorithm for RNA-seq data. STAR aligns non-contiguous (exons) sequences straight to a reference genome by two main steps. First, in the seed searching phase, a maximal mappable prefix (MMP) is employed to correctly map the reads against the reference genome even if the read contains a splice junction. Later, the algorithm attaches the seeds previously aligned and constructs alignments of all read sequences. Finally, using a defined local alignment score system, a seed combination is called the best alignment for a read if it has the highest score.

Segemehl (Hoffmann et al. 2009, 2014) maps short reads to reference genomes, detecting mismatches, insertions, and deletions. Moreover, Segemehl can deal with different read lengths and can map primer or polyadenylation contaminated reads correctly. Segemehl matching method is based on enhanced suffix arrays, supporting the SAM format and queries with gzipped reads to save disk and memory space and allowing both bisulfite sequencing and split read mappings.

Minimap2 (Li 2018) is a fast RNA-seq aligner that maps long-reads against a reference database. Minimap deals with long noisy reads at high error rates generated from both ONT and PacBio sequencing. In aligning spliced sequences, it recovers insertions and deletions and predicts correct splice junctions for correct alignment.

There are many other computational methods to map short reads to a reference genome, as shown in Table 3.2.

3.2.2.3 Software for Assembling

Mapping approaches for transcriptome reconstruction can be particularly tricky since correctly assigning reads to a reference genome are usually computational demanding, prone to errors by splice junctions, sequencing inaccuracy, absence, or unfinished reference sequences. Contrarily, assembly (or de novo assembly) approaches do not require any reference genome, the desired feature, especially when genomic sequences are not available or do not attend minimum quality demands.

The assembly tools algorithms usually aim to group reads with similar extremities, i.e., the overlapping of one read's end to another indicates that both probably belong to the same transcript (Fig. 3.5). These similar extremities enable the reconstruction of larger regions of the transcripts. As said before, each of these groups is called a contig. The sequence resulting from the overlapping reads in one contig, called consensus, is a predicted (fragment of) transcript.

Short reads sequencing usually have greater accuracy than long reads; however, short reads often align in multiple regions, causing problems to find correct isoforms. Thus, long reads sequencing can improve the discovery and identification of isoforms, but it is less accurate due to base-calling errors. When possible, the mixture of long reads and Illumina short reads are the best strategy for assembling complete and accurate transcriptomes (Kovaka et al. 2019).

Trinity (Grabherr et al. 2011) software package represents a major de novo assembly method composed of three modular components: Inchworm, Chrysalis, and Butterfly. Initially, the inchworm algorithm decomposes and selects from all reads the most common k-mer ($k = 25$) as the seed promotes contig assembly based on greedy extension ($k-1$)-mer overlaps. Chrysalis clusters and connects Inchworm contigs in components that could be originated from alternative splicing or related

Table 3.2 Mapping software and their websites

Mapping softwares	Website/repository
Bowtie1/Bowtie2	http://bowtie-bio.sourceforge.net/bowtie2/index.shtml
Minimap2	https://github.com/lh3/minimap2
Segemehl	https://www.bioinf.uni-leipzig.de/Software/segemehl/
STAR	https://github.com/alexdobin/STAR
TopHat2	http://ccb.jhu.edu/software/tophat/index.shtml
NextGenMap	http://cibiv.github.io/NextGenMap/
Kallisto	https://pachterlab.github.io/kallisto/
HPG Aligner	https://github.com/opencb/hpg-aligner

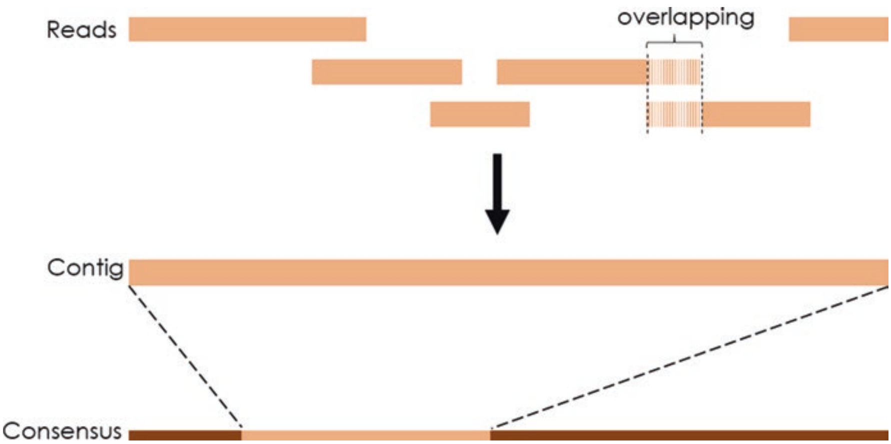

Fig. 3.5 Reads that contain overlapping extremities indicate that they are parts of the same transcript. Multiple reads overlapping each other creates a longer fragment called contig that represents a specific *locus* of consensus sequence

genes. If contigs overlap k−1 bases between themselves and reads span the splicing junction among different contigs, then highly structured de Bruijn graphs are built for each component. Finally, the Butterfly component integrates de Bruijn graphs produced in the Chrysalis stage to their corresponding RNA-seq read, allowing the reconstruction of the transcriptome sequences similar to the original transcripts.

Trans-ABySS (Transcriptome Assembly By Short Sequences) (Robertson et al. 2010) is a de novo assembly tool designed to reconstruct paired-end short reads from transcriptome data. Trans-AbySS derived from ABySS (Simpson et al. 2009), a short-read genomic data assembler. Trans-AbySS employees de Bruijn graph approach promoting data assembly with standard k-mers (k = 32) promoting a good balance between assembling frequent and rare transcripts. Trans-ABySS single-processor version is useful for assembling genomes of up to 100 Mbases. In contrast, the parallel version (implemented using MPI) can be assembled larger genomes, benefiting from multi-threaded processing.

MaSuRca (Zimin et al. 2013) process hybrid assembly, using “super-reads” from short-reads to de novo assemble reads and construct synthetic long reads with a low error rate and combining with long reads from Nanopore/Pacbio. Its assembly permits work with long reads and short reads at the same time, overcoming high error rates from long-reads sequencing (Table 3.3).

3.2.2.4 Software for Analysis

In transcriptome projects, quantitative analysis, differential expression, and transcript annotation are extensively used. Many suitable tools for these analyses are available in R language, which provides a wide variety of statistical and graphical

Table 3.3 Assembly software and their websites

Assembly	Website/repository
BWA	https://github.com/lh3/bwa
Cufflinks	http://cole-trapnell-lab.github.io/cufflinks/
MaSuRca	https://github.com/alekseyzimin/masurca
SPAdes	http://cab.spbu.ru/software/spades/
StringTie2	https://github.com/skovaka/stringtie2
Trans-ABySS	https://github.com/bcgsc/transabyss
Trinity	https://github.com/trinityrnaseq/trinityrnaseq/wiki
SOAPdenovo	https://github.com/aquaskyline/SOAPdenovo-Trans
Oases	https://github.com/dzerbino/oases

resources. R is highly extensible, allowing us to output well-designed publication-quality plots, including effective data handling and storage facility and a collection of intermediate tools for data analysis. Bioconductor (https://www.bioconductor.org) is a (mostly) R packages repository that provides open-source tools to analyze biological high-throughput data. Similarly, there are many Python-based resources as Biopython (https://biopython.org), a set of freely available tools for biological computation written in Python.

Quantitative Analysis

The transcript coverage is the number of reads "covering" (or the number of mapped reads in) a transcript. The greater the number, the more abundant is the expressed gene in an RNA-seq sample (Fig. 3.6). The RNASeqMap library (Leśniewska and Okoniewski 2011), for instance, provides classes and functions to analyze the RNA-sequencing data using the coverage profiles in multiple samples at a time.

Differential Expression

The differential expression refers to the study of the variability of genetic expression between samples. One important objective of RNA-seq projects is to identify the differentially expressed genes in two or more conditions (Rapaport et al. 2013). These genes are selected based on parameters, usually based on p-values generated by statistical modeling. The expression level is measured by the number of reads mapping to the transcript, such as transcripts per million (TPM), which is expected to correlate directly with its abundance level. This measure is different from gene probe-based methods, e.g., microarrays. In RNA-seq, the expression of a transcript is limited by the sequencing depth. It depends on the expression levels of other transcripts, in contrast to array-based methods, in which probe intensities are independent of each other. That one and other technical differences have motivated many statistical algorithms, with different approaches for normalization and

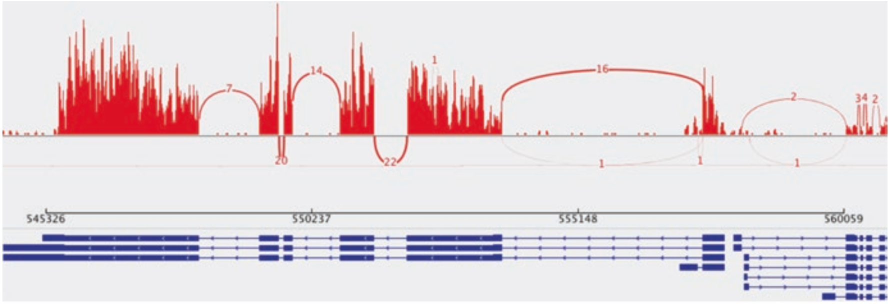

Fig. 3.6 Read coverage of transcripts relative to a reference genome. Each red bar plotted indicates a *locus* alignment coverage. The arcs represent splicing junctions between exons. Finally, the arc numbers are the observed numbers of reads across the junction. (Source: https://training.galaxyproject.org/training-material/topics/transcriptomics/tutorials/ref-based/tutorial.html)

differential expression detection. For example, Poisson or negative binomial distributions to model the gene count data and various normalization procedures are common approaches.

Cufflinks (Trapnell et al. 2010) may be used to measure global de novo transcript isoform expression. It assembles transcripts, estimates their abundances, and determines differential expression (Trapnell et al. 2013) in RNA-seq samples. Moreover, Cufflinks accepts reads aligned by other mappers and assembles the alignments to a parsimonious set of transcripts. It then estimates the relative abundances of these transcripts based on how many reads support each one, considering biases in library preparation protocols.

Some articles discuss and compare statistical methods to compute differential expression. In a review, Kvam et al. (2012) compared four statistical methods – edgeR, DESeq, baySeq, and a method with a two-stage Poisson model (TSPM). Rapaport et al. (2013) describe an extensive evaluation of common methods – Cuffdiff (Trapnell et al. 2013), edgeR (Robinson et al. 2010), DESeq (Anders and Huber 2010), PoissonSeq (Li et al. 2012), baySeq (Hardcastle and Kelly 2010), and limma (Smyth 2004) adapted for RNA-seq use, using the Sequencing Quality Control (SEQC) benchmark dataset and ENCODE data.

Splice Junctions

Splice junctions are nucleotide sequences at the exon–intron boundary in the pre-messenger RNA of eukaryotes removed during the RNA splicing. This process can generate many processed transcripts from a single gene. Computationally, the problem is to recognize, given a sequence of DNA, the boundaries between exons (the parts of the DNA sequence retained after splicing) and introns (the parts of the DNA sequence that are spliced out). This problem consists of two subtasks: recognizing exon/intron boundaries (called EI sites) and recognizing intron/exon boundaries (IE sites). IE borders are called "acceptor sites," while EI borders are called "donor sites." The recognition and quantification of splice variants are among the advances of RNA-seq over microarray to measure differential gene expression.

The splice junctions help to delineate and quantify the transcript model, as observed in Fig. 3.6.

Tophat (Trapnell et al. 2009) identifies splice junctions, producing the junctions. bed file, where the field score is used to indicate coverage depth. The identified splice junctions can be displayed in browsers (e.g., UCSC genome browser (Kuhn et al. 2013)) using.bed files encoding splice junctions. Junction files should be in the standard.bed format. Pasta (Patterned Alignments for Splicing and Transcriptome Analysis) (Tang and Riva 2013) is a splice junction detection algorithm designed for RNA-seq data, based on a highly accurate alignment strategy on a combination of heuristic and statistical methods to identify exon–intron junctions with high accuracy.

Annotation

The annotation step aims to assign a biological function for each transcript, identifying genes and finding more information, e.g., biological categories and ontologies. The annotation process is characteristic of novel transcriptomes since reference genomes and transcriptomes are typically associated with curated gene annotation.

The annotation methods can be organized into two classes:

- Pairwise comparison of every transcript against a file with known transcripts and their corresponding annotation. This can be done by comparing the nucleotides or the translated nucleotides.
- *Ab initio* gene prediction, where the presence of structural features and motifs of known genes are used to infer function.

The pairwise sequence comparison (or pairwise alignment), where a query sequence (transcript of the organism of interest) is compared with annotated sequences datasets, relies on an algorithm that computes an alignment among two transcripts. The hypothesis is based on Darwin's evolution theory, which claims that living organisms evolved from ancestor organisms. Therefore, if two transcripts have similar sequences, they may be homologs and probably share the same biological functions. This means that biological function may be inferred from similar sequences. Important pairwise algorithms, which produce alignments between pairs of sequences, are Smith-Waterman (Smith and Waterman 1981) and BLAST (Altschul et al. 1990).

Similar to the assembly step, the main difficulty in the annotation is due to the transcript length. The resulting genes may be fragmented, causing loss of information. Since alignment programs are error-tolerant, it is reasonable to expect that the annotation for transcripts (predicted from reads generated by high-throughput sequencers) is correct if functions of genes of other organisms have been found correctly.

In contrast, finding genes ab initio is not so error robust since sequencing errors can lead to incorrect gene prediction. In particular, sequencing errors introducing a stop codon can result in an incorrectly predicted gene.

3.3 Single-Cell Transcriptome Sequencing (scRNA-seq)

Although cells in an organism share almost identical genotypes, gene expression is heterogeneous and reflects the activity of a subset of genes. ScRNA-seq technologies are capable of generating data sets that describe the transcriptome of single cells. Single-cell transcriptome sequencing (scRNA-seq) expands the biological panorama granted by RNA-seq. It allows to estimate the expression levels of the whole transcriptome or targeted gene expression from a single cell and addresses new biological questions such as the heterogeneity of cell responses and their gene regulatory networks. It emerged with an mRNA-seq assay where a single mouse blastomere was sequenced, detecting the expression of 75% more genes than microarray techniques (Tang et al. 2009). This pioneer scRNA-seq method profiled RNA transcriptomes from single cells using oligo-dT primers followed by ligation adapter PCR (Tang et al. 2009). This method's limitation is the reverse transcriptase's inefficiency on the first-strand cDNA synthesis, causing a 3′ bias.

Eventually, new protocols and lower sequencing costs made scRNA-seq more accessible as technologies advance, resulting in continuously growing datasets, ranging from ~10^2 to ~10^6 cells. Some of the most distinguished methods for scRNA-seq are Smart-seq (Ramsköld et al. 2012), Smart-seq2 (Picelli et al. 2014), Drop-seq (Macosko et al. 2015), inDrop (Klein et al. 2015), CEL-seq2 (Hashimshony et al. 2016), 10× Chromium (Zheng et al. 2017), and Smart-seq3 (Hagemann-Jensen et al. 2020).

In general, scRNA-seq methods tag transcripts to make it possible to identify their cell of origin and generate libraries for sequencing. scRNA-seq sequencing data can both come from next-generation sequencing (NGS) and single-molecule sequencing (SMS) (Gao 2018). Smart-seq, Smart-seq2, Smart-seq3, and CEL-seq2 can be considered low-throughput plate-based methods, where the cells are sorted into wells of a multi-well plate. Alternatively, bead-based high-throughput methods distribute the cell suspension into tiny droplets containing reagents and barcoded beads (Drop-seq, 10× Chromium, and inDrops) or into well microplates (Seq-Well and sci-RNA-seq) to produce single droplets or well microplates with one cell and one bead marking the cDNA generated from that cell (Ding et al. 2019).

The Smart-Seq (Ramsköld et al. 2012) addressed this problem using a Moloney Murine Leukemia Virus Reverse Transcriptase (M-MLV RT) to synthesize cDNA with long messenger RNA templates. Unique molecular identifiers (UMI) were incorporated into each RNA molecule as unique barcodes before the whole transcriptome amplification (WTA) amplification (Islam et al. 2014). Smart-seq2 (Picelli et al. 2014) is an approach that combines sensitivity (it captures a considerable fraction of RNAs present in cells) with full-length coverage of transcripts and can detect genes per cell and across cells enabling quantifying isoform-level expression from single cells, but without the incorporation of unique molecular identifiers (UMIs). Smart-seq3 (Hagemann-Jensen et al. 2020) improves the sensitivity of Smart-seq2, adding optimized reverse transcriptase and buffer conditions together with a partial Tn5 motif and a tag sequence in the template-switching oligonucleotide to directly assign individual RNA molecules to isoforms and establish their allelic origin in single cells.

Drop-Seq dissociates a tissue into individual cells and encapsulates them into droplets with microparticles that deliver barcoded primers. After associating barcodes to each cell's RNAs, they are reverse-transcribed into cDNAs to generate beads called "Single-cell Transcriptomes Attached to Microparticles" (STAMPs). Then, the STAMPs are amplified in pools for high-throughput mRNA-seq (Macosko et al. 2015) (Fig. 3.7). The 10× Chromium system works, generating a large number of "Gel Bead-in-emulsions partitions" (GEMs) to index each cell's transcriptome separately. The barcoded gel beads (read, 10xbarcode, UMI, oligo-dT) are mixed with cells, enzymes, and partitioning oil to create single-cell GEMs. Then, the single-cell GEMs undergo reverse transcriptase (RT) to generate a 10× barcoded cDNA library where cDNA from individual cells share a common 10× barcode that can be used for single-cell whole transcriptome sequencing or target sequencing workflows (10× Genomics Inc. 2020). In the inDrops method, the cells are also encapsulated into droplets with lysis buffer, hydrogel microspheres carrying barcoded primers, and an RT mix. After the release of primers, cDNA in each droplet is barcoded during reverse transcription. After the droplets are broken, all cellular material can be amplified for sequencing (Klein et al. 2015).

The Smart-seq methods can detect many genes in a cell, including low abundance transcripts and alternatively spliced transcripts. CEL-seq2 (Hashimshony et al. 2016), Drop-seq, 10× Chromium, and inDrops can quantify mRNA levels with less amplification noise using UMIs, enabling less and profiling isoform-level RNA counting. As a limitation, inDrops droplets may contain two cells or two different types of barcodes. Table 3.4 shows a comparison of some important aspects of these scRNA-seq methods.

3.3.1 *scRNA-seq Computational Analysis*

Despite the different methods available, the scRNA-seq data is essentially the result of high-throughput sequencing cDNA reverse transcribed from mRNA isolated from a pool of cells. The primordial difference is that the sequenced data is somehow tagged to assign its origin to individual cells. Some standard steps remain the same as RNA-seq, such as the reads quality filtering and reads mapping to a reference genome. Reads quality filtering can be applied to filter the read quality using a quality metric for sequencing like the percentage of base calls (Q score). The reads are then mapped to a reference genome and quantified to generate an expression profile matrix. Some scRNA-seq specialized tools can both align and quantify the reads. Additionally, a second filtering step can be performed after quantifying reads to discard cells expressing a low number of genes or a high number of mitochondrial genes (Park and Lee 2020). The next step of the pipeline is data normalization using a metric for expression normalization as TPM (Transcripts Per Kilobase Million) or RPKM (Reads Per Kilobase per Million) (Gao 2018). At this point, the scRNA-seq computational analysis reaches its two fundamental problems: cluster analysis and sample/feature reduction.

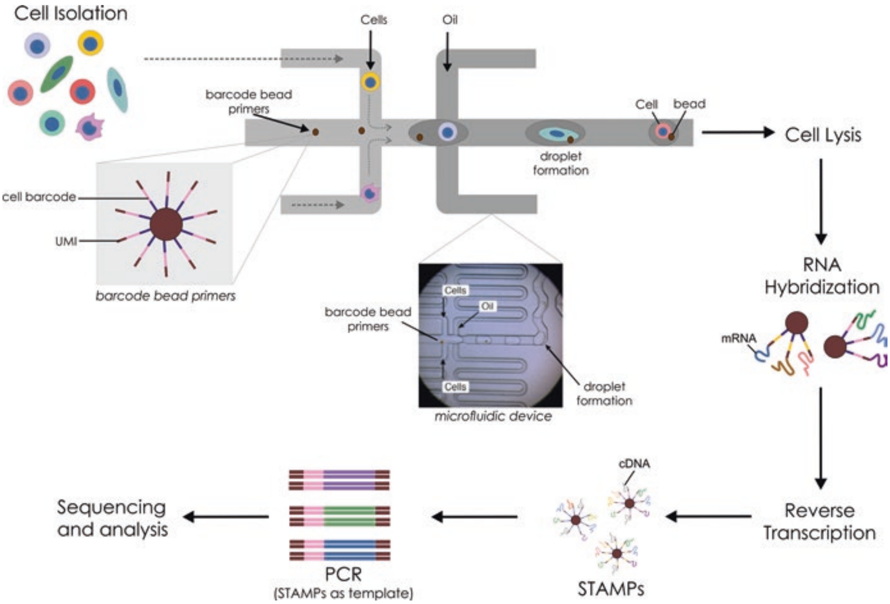

Fig. 3.7 Individual cell's transcriptome can be analyzed using scRNA-seq. Tissue disrupted single cells are mixed with barcode bead primers and reagents in oil droplets in a microfluidic device. The formed droplet contains a single cell and a barcode. After lysis and primer hybridization, RNA is reverse transcribed and sequenced as in a conventional RNA-seq experiment. The UMI and barcode sequence will be incorporated in the final sequenced reads and will guide the scRNA-seq processing

Normalization allows consistent comparison of gene expression measurements in individual cells, including technical variation due to the numbers of sequenced readings or transcripts identified per cell. A normalized gene expression matrix is a matrix with n samples (cells) by m features (genes, transcripts, or exons), depending on the read's size. For example, for transcripts as features, PacBio full-length transcriptome could be the right choice, or for Illumina short-length reads, the features could be genes. As the number of annotated genes of the target organism, the matrix could be large and sparse, which justifies the sample and feature reduction. The feature selection can be understood as removing genes unhelpful to distinguish biological variation across samples.

Clustering cells allow us to identify cells with correlated phenotype by grouping them based on their gene expression profiles' similarity. This is achieved using dimension reduction algorithms to embed the expression matrix into a low-dimensional space that summarizes the data structure in as few dimensions as possible (Gao 2018; Luecken and Theis 2019). These low-dimensional spaces can come from dimension reduction methods as Principal Component Analysis (PCA), Independent Component Analysis (ICA), Linear Discriminant Analysis (LDA), Multidimensional Scaling (MDS), and t-distributed Stochastic Neighbor Embedding (t-SNE).

Table 3.4 Comparison of some aspects of low- and high-throughput scRNA-seq methods

	Low-throughput				High-throughput		
Methods	Smart-Seq (Ramsköld et al. 2012)	Smart-seq2 (Picelli et al. 2014)	Smart-seq3 (Hagemann-Jensen et al. 2020)	CEL-seq2 (Hashimshony et al. 2016)	Drop-seq (Macosko et al. 2015)	10× Chromium (Zheng et al. 2017)	inDrop (Klein et al. 2015)
Single-cell isolation	Micromanipulation	[a]MCP/FACS	FACS	FACS	Droplets	Droplets	Droplets
Coverage	Full length	Full length	Full length	–	–	–	–
UMI	No	No	No	6 bp	8 bp	28 bp (Cell barcode & UMI)	6 bp
First-strand synthesis	oligo-dT	oligo-dT	oligo-dT	oligo-dT	oligo-dT	oligo-dT	oligo-dT
Second-strand synthesis	Template switching	Template switching	Template switching	RNAseH/DNA Pol	Template switching	Template switching	Template switching
cDNA Amplification	PCR	PCR	PCR	IVT	RT-PCR	RT-PCR	RT-PCR

[a] Microcapillary Pipette (MCP)

3.3.2 scRNA-seq Analysis Tools

Seurat (Hao et al. 2020) is an R package that integrates quality control, analysis, and exploration of single-cell RNA-seq data. It is based on a *Seurat object*, which serves as a container for both data (like the count matrix) and analysis (like PCA, or clustering results). Also, Seurat can make simultaneous measurements of multiple data types from the same cell, known as multimodal analysis, and analyze spatially resolved RNA-seq data.

Cell Ranger is a set of tools to process Chromium single-cell RNA-seq data. The package contains *cellranger mkfastq* which demultiplexes raw base call (BCL) Illumina files into fastq files. These files are then taken as input by *cellranger count* to perform alignment, filtering, barcode, and UMI counting. In the next step, *cellranger aggr* aggregates and normalizes the outputs from multiple runs of *cellranger count* recomputing the feature-barcode matrices and analyzing the combined data. The *cellranger reanalyze* reruns the dimensionality reduction, clustering, and gene expression algorithms from the feature-barcode matrices produced by *cellranger count* or *cellranger aggr*. Cell Ranger also uses the aligner STAR (Dobin et al. 2013) and the output is delivered in formats like bam, mex, csv, hdf5, and html.

Meta Cell (Baran et al. 2019) is a tool for deriving metacells and analyzing scRNA-seq data. Metacells are a theoretical group of scRNA-seq cell profiles statistically equivalent to samples derived from the same RNA pool, which is obtained by computing partitions of scRNA-seq datasets into disjoint and homogenous groups of cells.

SEQC (Azizi et al. 2018) is a Python package for scRNA-seq analysis in a cloud and subsequent analyzes on a local machine. It has Spliced Transcripts Alignment to a Reference – STAR (Dobin et al. 2013), Samtools (Li et al. 2009), and HDF5 data model as dependencies and has been tested for 10× Genomics v2 and inDrop v2 data.

zUMIs (Parekh et al. 2018) is a pipeline to process RNA-seq data with or without UMIs. zUMIs take cDNA fastq files and other reads containing UMI and Cell Barcode information as input. It was written using R, Perl shell, and Python programming languages and has as dependencies STAR (Dobin et al. 2013).

robustSingleCell is an R package that provides clustering and comparison of population compositions across tissues and experimental models through a similarity analysis characterizing transcriptomic similarities in meta-clusters by identifying their defining overexpressed genes (Magen et al. 2019) (Table 3.5).

Table 3.5 Computational tools for scRNA-seq analysis

Tools	Availability
Seurat (Hao et al. 2020)	https://github.com/satijalab/seurat
SEQC (Azizi et al. 2018)	https://github.com/ambrosejcarr/seqc
zUMIs (Parekh et al. 2018)	https://github.com/sdparekh/zUMIs
CellRanger (10× Genomics)	https://support.10xgenomics.com/single-cell-gene-expression/software/pipelines/latest/what-is-cell-ranger
Meta Cell (Baran et al. 2019)	https://tanaylab.github.io/metacell
robustSingleCell (Magen et al. 2019)	https://github.com/asmagen/robustSingleCell

3.4 Case Study 1

RNA-seq as an Efficient Tool to Analyze and Identify Gene Expression Patterns Related to Murine Bone Marrow-Derived Macrophage's Susceptibility and Resistance to Candida albicans *Infection*

The improvements in organ transplantation techniques and the rise of immune-compromised diseases, like AIDS, are directly linked to the exponential growth of opportunist infections in these patients. Therefore, the study of the etiological agents of these diseases, particularly fungal pathogens, together with the immune response they elicit, became an important issue (Marr et al. 2002; Richardson and Lass-Flörl 2008; Miceli et al. 2011). *Candida albicans* appears to be the leading cause of invasive infections among fungi, showing high morbidity and mortality rates (Chi et al. 2011; Shigemura et al. 2014).

Many studies have been done to understand the aspects of immune responses to *C. albicans* (Tierney et al. 2012; Miramón et al. 2013; Hünniger et al. 2014; Martínez-Álvarez et al. 2014). In this case study, the transcriptomic response of murine bone marrow-derived macrophages (BMDMs) from BALB/c (resistant) and DBA/2J (susceptible) mice strains to *C. albicans* infection was analyzed by RNA-seq to compare both transcriptomic patterns. Therefore, this case study's main objective was to identify BMDMs gene expression patterns between resistant and susceptible mice after *C. albicans* infection by the analysis of the resulting transcriptome profiles.

Bone marrow was extracted from the mice, and the hematopoietic stem cells were then differentiated into macrophages. An amount of 2×10^6 BMDMs were co-cultured with 4×10^6 *C. albicans* yeasts for 90 min, and the RNA was extracted using RNeasy (Qiagen). RNA quality and concentration were verified employing a Bioanalyzer (Agilent) and NanoDrop (Thermo Scientific), respectively. Three μg of total RNA was used for the library preparation, including a step of rRNA depletion using Ribozero (Epicentre) before library construction and sequencing in an Illumina Hiseq platform.

The sequencing results were provided in fastq format. FastQC was used to assess quality. Adaptors clipping and quality trimming were performed using Cutadapt

(Martin 2011). Two mapping software, NextGenMap (NGM) (Sedlazeck et al. 2013) and Tophat2 (Kim et al. 2013), were employed. Since both generate a similar number of mapped reads, we chose NextGenMap due to its faster analysis. Low-quality mappings were removed using Samtools (Li et al. 2009), which was also used to sort, index, and convert the mapping results from sam to.bam files. Bedtools (Quinlan and Hall 2010) were then used to count reads for both genes and exons, and generate a table of these counts, to be analyzed for differential expression. As said before, differential expression can be analyzed using different methodologies (Wagner et al. 2012; Soneson and Delorenzi 2013), and EdgeR (Robinson et al. 2010) and DESeq (Anders and Huber 2010) were chosen. Both outputted very similar results. Alternative splicing can be checked by differential exons usage (Anders et al. 2012). Therefore, the resulting list of genes or transcripts differentially expressed (adjusted p-value <0.05 and fold change $\geq \pm 2.5$) was checked for gene ontology (GO terms) using ClusterProfiler (Yu et al. 2012) Bioconductor package.

Several problems may occur in RNA-seq projects, and here we point out some of these:

- Infection conditions: the optimization of the protocols of co-culture conditions, as well as RNA extraction, may be hard to adjust. Setting a multiplicity of infection (MOI – proportion of host/pathogen cells in the co-culture) that suffices to induce a transcriptomic response in the host cells is the first step. However, a very high MOI may result in host cells' death and apoptosis, which may result in altered gene expression or low amounts of RNA extracted from these cells.
- Infection time: the definition of correct time intervals of interaction between pathogen and host cells is essential since different genes have different kinetics of transcription during co-culture. This may vary drastically for different host-pathogens and also depends on the major question of interest.
- Biological replicates: in transcriptomic studies, robust statistical analysis is fundamental. In this sense, the experimental design has to incorporate proper biological replicates to allow valid statistical inferences (Robles et al. 2012).
- Library preparation and sequencing parameters: the choice of the preparation methodologies, e.g., poly-A enrichment protocols versus rRNA depletion protocols, or paired-end versus single-end sequencing, may strongly impact the results. Improper handling of samples in this step may also result in sample degradation or inefficient rRNA depletion, which may compromise the whole experiment if not properly adjusted. A well-defined experimental design for the sequencing step must also be taken into consideration. A final low coverage of the transcriptome can result in an inadequate analysis of differential gene expression.

A significant disparity was observed in the differentially expressed genes upon *C. albicans* infection between BMDMs from both mice strains. BMDMs from the susceptible DBA/2J strain modulated a higher number of genes (4021) upon infection with *C. albicans* than BMDMs from the resistant BALB/c strain (99), and both sets have few genes in common (60) (Fig. 3.8).

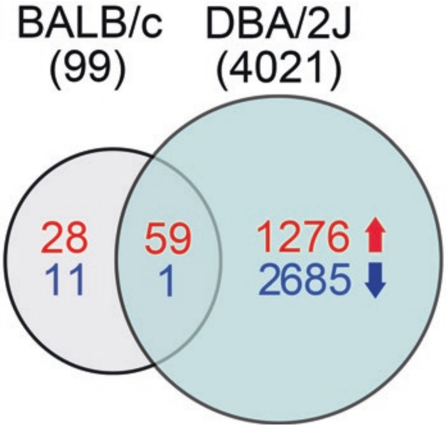

Fig. 3.8 Venn diagram of positively (red) and negatively (blue) regulated genes in BMDMs from BALB/c and DBA/2J mice strains infected with *C. albicans*. Differentially expressed genes were considered when adjusted *p*-value <0.05 and fold change ≥±2.5

Analysis focusing on GO categories of biological processes revealed enrichment (*p* <0.01) of upregulated genes in terms related to inflammatory response, cellular response to biotic stimulus, and cytokine production in both resistant and susceptible strains (Fig. 3.9). However, they markedly differed in the modulation of some terms. For example, macrophages from the resistant strain upregulated genes related to apoptosis and neutrophil chemotaxis. In contrast, macrophages from the susceptible strain upregulated genes involved in innate immune response and leukocyte migration.

3.5 Case Study 2

Single-Cell Sequencing of SARS-CoV-2 Infected Individuals with Distinct Levels of Severity

COVID-19 outbreak has caused critical consequences for all countries, including many deaths and hospitalization, beyond the economic issues. Beyond the vaccination, it is important to research specific drugs to treat the affected individuals. Monoclonal antibodies have demonstrated their effectiveness in medicine (Maranhão et al. 2020). Therefore, developing new potential antibodies as an alternative against viral proteins remains highly valuable.

This example of scRNA-seq analysis is based on the work "Single cell RNA and immune repertoire profiling of COVID-19 patients reveals novel neutralizing antibody" from Fang Li et al. (2020). They have conducted a study using single-cell transcriptome sequencing (scRNA-seq), single-cell BCR sequencing (scBCR-seq), and deep BCR repertoire to reveal neutralizing antibody sequences in patients who have recently cleared the virus. They collected blood samples (peripheral blood mononuclear cells – PBMCs) from 16 COVID-19 patients and eight healthy controls to reveal immune cells' changes caused by SARS-CoV-2 infection. Fang Li et al. (2020) scRNA-seq was performed using 10× Genomics. The original data is available in the Zenodo under the accession URL: https://zenodo.org/record/3744141.

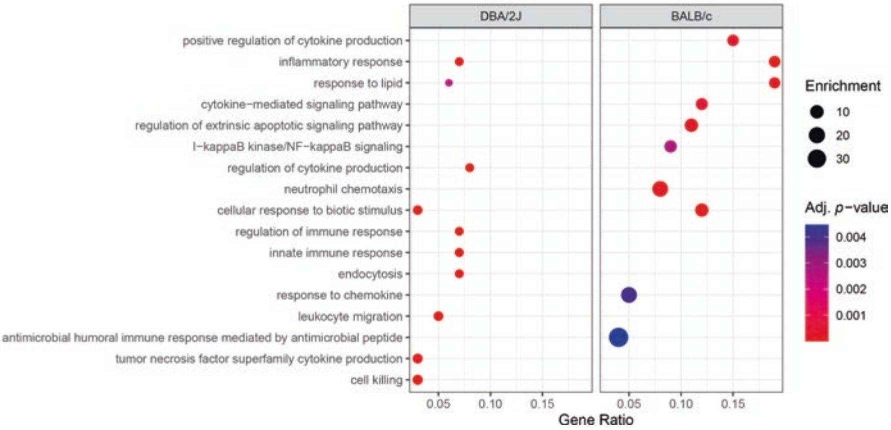

Fig. 3.9 Gene ontology enrichment of upregulated genes in BMDMs from DBA/2J and BALB/c mice strains upon *C. albicans* infection. Enriched GO terms (adjusted *p*-value <0.01) from biological processes category associated with upregulated genes in BMDMs derived from the susceptible DBA/2J (left) and the resistant BALB/c (right) mice strains. Dot size is representative of enrichment (gene modulated ratio/gene background ratio) for each GO term. Only major terms related to immune response were plotted

This case study uses a Fang Li et al. (2020) sample subset with data from two patients to demonstrate how to identify distinct types of cells based on clustering their transcripts and how to obtain the differentially expressed genes. The input files are barcodes.tsv, datasets.rds, genes.tsv, and matrix.mtx. For this case study, we filtered the complete data to work only with patient 3 (P3) and patient 10 (P10) samples, both from 59 years old females with distinct levels of COVID-19 severity. P3 had severe symptoms, and P10 had moderate symptoms.

This example uses the R package Seurat 4.0 (Hao et al. 2020) to perform the analysis directly from the matrix. The following R codes are commented, and their results presented. The first step is to install and load the required R packages. Seurat 4.0 requires R version 4.x.

```
if (!requireNamespace("BiocManager", quietly = TRUE))
    install.packages("BiocManager")
BiocManager::install(version = "3.12")

BiocManager::install('ggplot2')
BiocManager::install('ggrepel')
BiocManager::install('limma')
BiocManager::install('calibrate')
BiocManager::install('dplyr')
BiocManager::install('Matrix')
BiocManager:: install('Seurat')

library(ggplot2)
library(ggrepel)
library(limma)
library(calibrate)
library(dplyr)
library(Matrix)
library(Seurat)
```

The next step is to download, extract, and read the COVID-19 data. This will result in a matrix with 33,538 lines and 96,404 columns. The columns represent each tagged transcript, and the lines represent the genes where those transcripts were mapped.

```
system("wget https://zenodo.org/record/3744141/files/COVID-19.tar.gz")
system("tar -xzvf COVID-19.tar.gz")
covid_19_data <- Read10X(data.dir = "COVID-19")
dim(covid_19_data) # dimensions for full data
```

Once loaded the full data, now it is possible to filter them to work only with P3 and P10 samples by using regular expression to identify only data from patients P3 and P10. The new dimensions of P3 and P10 data will be 33,538 lines (genes) by 16,056 columns (tagged transcripts).

```
p3_and_p10_data <- covid_19_data[, grep(pattern = "P3|P10", colnames(covid_19_data))]
dim(p3_and_p10_data) # dimensions for selected data
```

The function CreateSeuratObject() initializes the Seurat object with the non-normalized data constrained by the following parameters: minimal of two cells with at least 20 expressed genes and at least 2,000 features. The dimension of the object in this case will be 17,169 genes and 2,123 tagged transcripts that met the criteria.

```
covid_p3_p10 <- CreateSeuratObject(
    counts = p3_and_p10_data,
    project = "COVID-19",
    min.cells = 2,
    min.genes = 20,
    min.features = 2000
)
dim(covid_p3_p10) # dimensions for loaded data
```

Before starting the data processing, we will create two new columns to add meta-information for the patients (P3 or P10) and for the mitochondrial percent in transcripts. The [[]] operator can add columns to an object. In this case, we create a column to identify patients P3 and P10. We also stash quality control (QC) stats for their mitochondrial samples, which are identified starting by “MT-”.

```
covid_p3_p10[["patient"]] <- sapply(strsplit(colnames(covid_p3_p10),"-"), `[`, 1)
covid_p3_p10[["perc_mitochondrial"]] <- PercentageFeatureSet(covid_p3_p10, pattern = "^MT-")
```

Next, it is possible to build a violin plot to visualize the QC metrics for number of features, read count and mitochondrial percentage, grouped by patient (Fig. 3.10).

```
plot_perc_mitochondrial <- VlnPlot(
    covid_p3_p10,
    features = c("nFeature_RNA", "nCount_RNA", "perc_mitochondrial"),
    ncol = 3,
    Group.by = "patient",
    log = TRUE
)
plot_perc_mitochondrial
```

The next step is to remove unwanted cells from the dataset. In this case we can apply a new filter to keep only samples with the number of features at least equal to 2000 and less than 5% of mitochondrial samples.

```
covid_p3_p10 <- subset(covid_p3_p10, subset = nFeature_RNA >= 2000 & perc_mitochondrial < 5)
```

To normalize the data, we can use function LogNormalize(), which normalizes the feature expression measurements for each cell by the total expression. It multiplies this by a scale factor (10,000 by default), and log-transforms the result.

```
covid_p3_p10 <- NormalizeData(covid_p3_p10, normalization.method = "LogNormalize", scale.factor = 10000)
```

Once normalized, the next step is to identify highly variable features (feature selection) using the method *vst* which, according to the manual of Seurat, fits a line to the relationship of log (variance) and log (mean) using local polynomial regression (loess). Then, it standardizes the feature values using the observed mean and expected variance (given by the fitted line). Then, it is computed the feature variance on the standardized values after clipping to a maximum (default is “auto” which sets this value to the square root of the number of cells).

```
covid_p3_p10 <- FindVariableFeatures(covid_p3_p10, selection.method = "vst", nfeatures = 2000)
```

At this point, it is possible to find, for instance, the 20 most highly variable genes identified (Fig. 3.11) that would be: ‘IGHA1’, ‘JCHAIN’, ‘IGHG1’, ‘IGKC’, ‘IGLC2’, ‘IGHG2’, ‘DERL3’, ‘IGLL5’, ‘IGHV3-23’, ‘ITM2C’, ‘IGKV3-20’, ‘MZB1’, ‘LILRA4’, ‘IGHV3-7’, ‘FKBP11’, ‘GNLY’, ‘IGKV4-1’, ‘TNFRSF17’, ‘STMN1’, and ‘HIST1H4C’. Interestingly, most of these genes are involved with the immune system, more precise to B lymphocytes, a known player in the inflammatory aspect of COVID-19. IGHA, the heavy constant chain of the immunoglobulin alpha, codes for an antibody isotype well characterized to participate in the mucosal immunity, the natural site of SARS-CoV-2 infection.

```
top20 <- head(VariableFeatures(covid_p3_p10), 20)
plot_top20 <- VariableFeaturePlot(covid_p3_p10)
plot_top20 <- LabelPoints(plot = plot_top20, points = top20, size = 2, hjust = .75, vjust = .75)
plot_top20
```

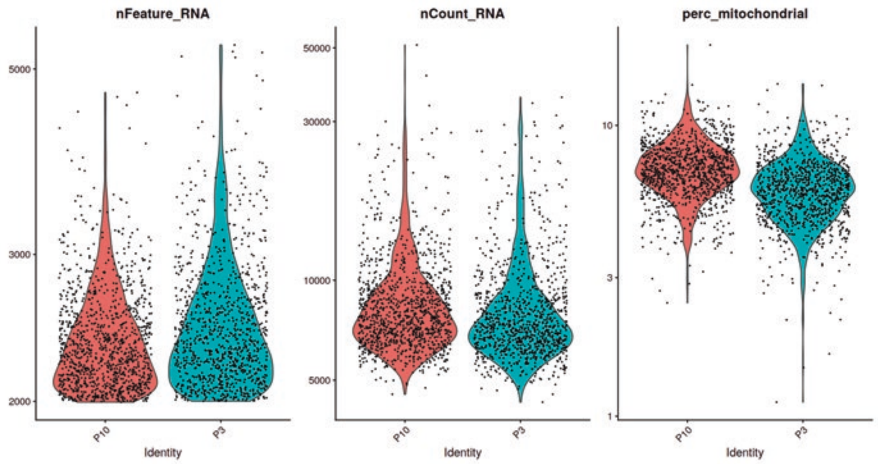

Fig. 3.10 Quality control (QC) metrics for the number of features, read count, and mitochondrial percentage, grouped by patient. Left: Number of featured genes for patients 3 (red) and 10 (blue) after filtering >2000 features. Middle: reads count for P3 and P10. Right: amount of reads from mitochondrial origin shown as percentage

Before performing the dimensional reduction, it is necessary to perform a linear transformation scaling the data. It is a standard pre-processing step prior to applying techniques like PCA.

```
all_genes_covid_p3_p10 <- rownames(covid_p3_p10)
covid_p3_p10 <- ScaleData(covid_p3_p10, features = all_genes_covid_p3_p10)
covid_p3_p10 <- RunPCA(covid_p3_p10, features = VariableFeatures(object = covid_p3_p10))
```

Now, it is possible to determine the dimensionality of the dataset. The function JackStraw() determines the statistical significance of PCA scores by randomly permuting a subset of data, and calculates projected PCA scores for these “random” genes. The ScoreJackStraw() function computes the scores significance by PCs showing a *p*-value distribution that is strongly skewed to the left compared to the *null* distribution.

```
covid_p3_p10 <- JackStraw(covid_p3_p10, num.replicate = 100)
covid_p3_p10 <- ScoreJackStraw(covid_p3_p10, dims = 1:5)
```

We can now cluster the cells. The function FindNeighbors() computes the k.param nearest neighbors for a given dataset using the k-nearest neighbors algorithm. Then, the function FindClusters() identifies clusters of cells from the SNN graph (result of the k-nearest neighbors algorithm). As higher is the resolution parameter, as larger will be the communities.

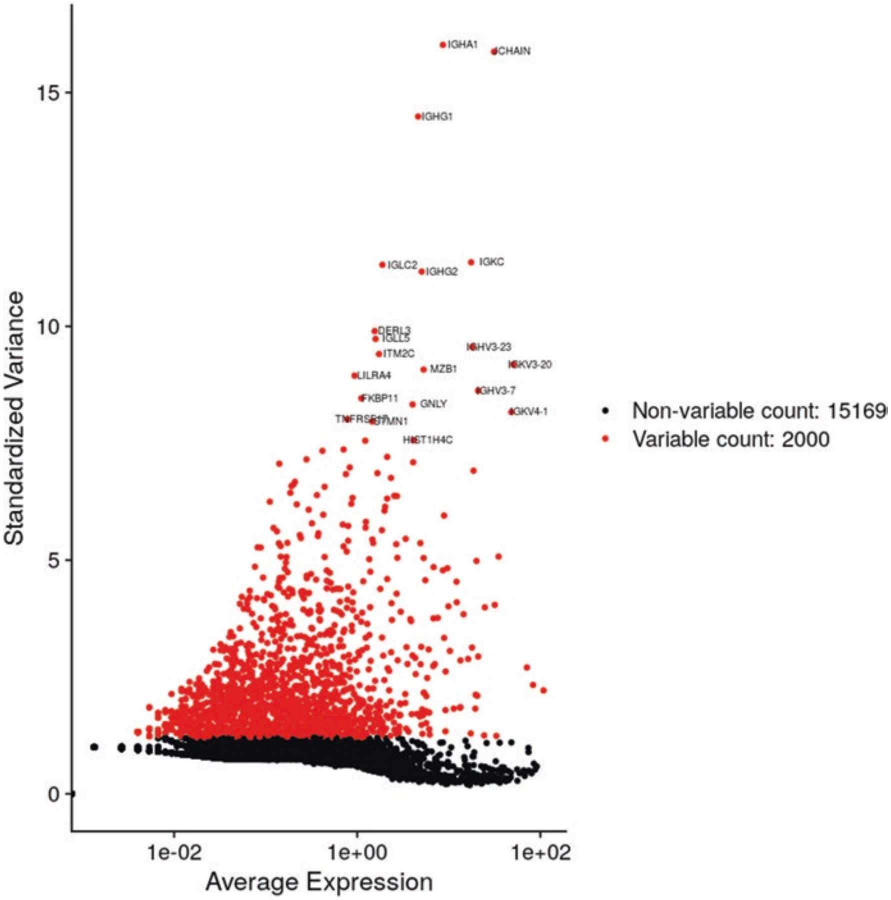

Fig. 3.11 Twenty most highly variable genes identified versus their average expression. In red are shown the 2000 most variable genes among cells, and 20 of them are labeled for exploration purposes

```
covid_p3_p10 <- FindNeighbors(covid_p3_p10, dims = 1:5)
covid_p3_p10 <- FindClusters(covid_p3_p10, resolution = 1)
```

Uniform Manifold Approximation and Projection (UMAP) is a dimensional reduction technique that can be used for visualization similarly to t-SNE, but also for general non-linear dimension reduction. It is founded on three assumptions about the data: (i) the data is uniformly distributed on a Riemannian manifold; (ii) the Riemannian metric is locally constant (or can be approximated as such); and (iii) the manifold is locally connected.

```
covid_p3_p10 <- RunUMAP(covid_p3_p10, dims = 1:5)
# It could be alternatively done using tSNE
# covid_p3_p10 <- RunTSNE(object = covid_p3_p10, dims.use = 1:5)
```

Finally, it is possible to plot the clusters of distinct types of cell in the samples. Using these parameters, we can find 10 clusters as can be seen in Fig. 3.12.

```
plot_clusters <- DimPlot(covid_p3_p10, label = TRUE)
plot_patient <- DimPlot(covid_p3_p10, label = TRUE, group.by = "patient")
plot_clusters + plot_patient
```

As it is possible to see in Fig. 3.12, the cluster number 4 has expressed genes both from patients 3 and 10. In this case, we first split data of patient 3 and 10 and then execute the function FindAllMarkers() can finds all differentially expressed genes for each of the patients in this dataset. Some constraints can be used to filter these genes, as min.pct that test for genes that are very infrequently expressed, which has as default value 0.1. The results are joined and the gene markers are filtered only for cluster number 4.

```
patient_splitted <- SplitObject(covid_p3_p10, split.by = "patient")
p3_markers <- FindAllMarkers(object = patient_splitted$P3)
p10_markers <- FindAllMarkers(object = patient_splitted$P10)
p3_markers[["patient"]] = "P3"
p10_markers[["patient"]] = "P10"
p3_p10_markers <- rbind(p3_markers, p10_markers)
cluster_4_markers <- p3_p10_markers[which(p3_p10_markers["cluster"] == "4"),]
```

The next step is to group the expressed genes as “Not Significant,” “Significant,” “FoldChange,” and “Significant&FoldChange” depending on the values of p-value and fold change. A plot (Fig. 3.13) with the most significant differentially expressed genes for the patients P3 and P10 can be built to highlight them.

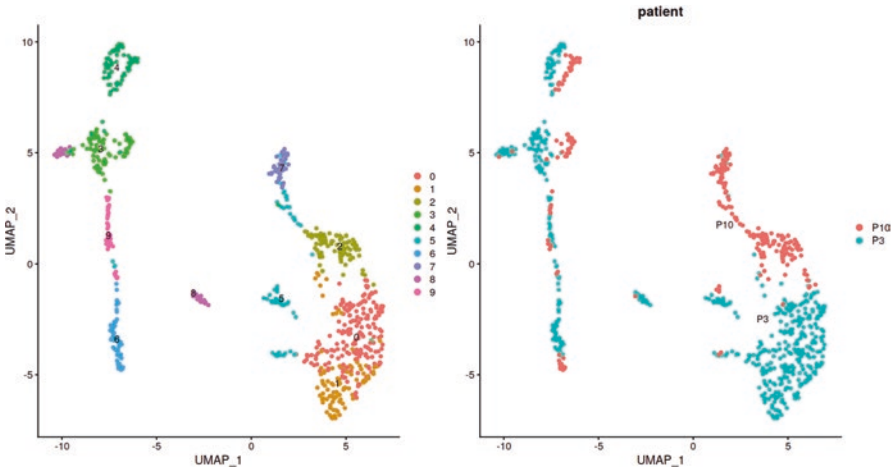

Fig. 3.12 Ten cell clusters belonging to the patients P3 and P10. Dimensionality reduction yields clusters of cells correlated by gene expression profile. Each cluster is labeled with a different color and is identified by a number that can be later annotated as a particular cell type based on the gene markers expressed in the cluster

```
# Preliminarly grouping all genes as "Not Significant"
cluster_4_markers["group"] <- "Not Significant"
# Change the grouping for the entries with significance but not a large enough Fold change
cluster_4_markers [which(cluster_4_markers["p_val_adj"] < 0.05 &
                     abs(cluster_4_markers["avg_log2FC"]) < 1 ),"group"] <- "Significant"
# Change the grouping for the entries a large enough Fold change but not a low enough p-value
cluster_4_markers [which(cluster_4_markers["p_val_adj"] > 0.05 &
                     abs(cluster_4_markers["avg_log2FC"]) > 1 ),"group"] <- "FoldChange"
# Change the grouping for the entries with both significance and large enough fold change
cluster_4_markers[which(cluster_4_markers["p_val_adj"] < 0.05 &
                    abs(cluster_4_markers["avg_log2FC"]) > 1 ),"group"] <- "Significant&FoldChange"
# Find and label the top peaks
top_peaks <- cluster_4_markers[which(cluster_4_markers["group"] == "Significant&FoldChange",
                               order(cluster_4_markers["p_val_adj"])),][1:10,]
p3_p10_plot <- ggplot(na.omit(cluster_4_markers)) +
  geom_point(aes(x = avg_log2FC, y = -log10(p_val_adj), colour = group, shape = patient), size = 5) +
  geom_text_repel(data=top_peaks[1:7,],aes(x = avg_log2FC, y = -log10(p_val_adj),label = gene))+
  scale_color_brewer(palette = "PuRd") +
  ggtitle("Most significant expressed genes in cluster 4 for patients P3 and P10") +
  xlab("log2 fold change") +
  ylab("-log10 adjusted p-value") +
  theme_minimal() +
  theme(legend.position = "bottom",
        legend.title = element_blank(),
        plot.title = element_text(size = rel(1), hjust = 0.5),
        axis.title = element_text(size = rel(1)))
p3_p10_plot
```

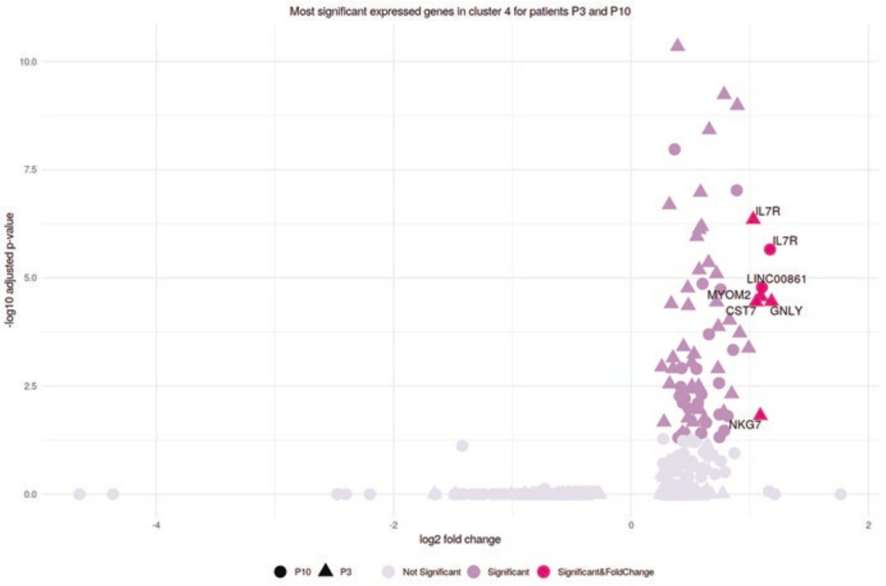

Fig. 3.13 Differentially expressed genes for patients 3 and 10. Each cluster of cell is tested against all remaining clusters. The most significant down- and upregulated genes are highlighted. Patient 3 is shown in the left and patient 10 in the right

The differentially expressed genes depicted in Fig. 3.13 reveal that six genes meet both statistical and fold change criteria. The IL7 receptor (IL7R) appears upregulated in both patients, while GNLY, MYOM2, CST7, and NKG7 are upregulated only in patient 3. The LincRNA 00861, a non-coding RNA, is upregulated only in patient 10, who had a milder infection. All of these genes are usually expressed in the cytotoxic CD8 lymphocytes, but patient 10, who evolved a strong inflammatory response, reveals a different gene response that is not associated with the LincRNA but strongly associated with genes involved in cytotoxicity (NKG7 and GNLY).

Single-cell computational analysis can consume vast computational resources. This case study uses only part of the original data to make it reproducible in a regular desktop or notebook computer. All these codes are available for download with the environment set-up instructions at https://github.com/waldeyr/single_cell_analysis.

Acknowledgments This work was supported by grants from CNPq, FAP-DF and CAPES/BioComp. FAH is supported by CAPES doctoral fellowship.

References

10x Genomics Inc (2020) Explore cellular diversity at scale. Product Sheet | Single Cell Gene Expression v3.1 with Feature Barcode technology. Pleasanton

Altschul SF, Gish W, Miller W et al (1990) Basic local alignment search tool. J Mol Biol 215:403–410. https://doi.org/10.1016/S0022-2836(05)80360-2

Amarasinghe SL, Su S, Dong X et al (2020) Opportunities and challenges in long-read sequencing data analysis. Genome Biol 21. https://doi.org/10.1186/s13059-020-1935-5

Anders S, Huber W (2010) Differential expression analysis for sequence count data. Genome Biol 11. https://doi.org/10.1186/gb-2010-11-10-r106

Anders S, Reyes A, Huber W (2012) Detecting differential usage of exons from RNA-seq data. Genome Res 22:2008–2017. https://doi.org/10.1101/gr.133744.111

Azizi E, Carr AJ, Plitas G et al (2018) Single-cell map of diverse immune phenotypes in the breast tumor microenvironment. Cell 174:1293–1308.e36. https://doi.org/10.1016/j.cell.2018.05.060

Baran Y, Bercovich A, Sebe-Pedros A et al (2019) MetaCell: analysis of single-cell RNA-seq data using K-nn graph partitions. Genome Biol 20. https://doi.org/10.1186/s13059-019-1812-2

Bentley DR, Balasubramanian S, Swerdlow HP et al (2008) Accurate whole human genome sequencing using reversible terminator chemistry. Nature 456:53–59. https://doi.org/10.1038/nature07517

Bolger AM, Lohse M, Usadel B (2014) Trimmomatic: a flexible trimmer for Illumina sequence data. Bioinformatics 30:2114–2120. https://doi.org/10.1093/bioinformatics/btu170

Chen S, Zhou Y, Chen Y, Gu J (2018) Fastp: an ultra-fast all-in-one FASTQ preprocessor. Bioinformatics 34:i884–i890. https://doi.org/10.1093/bioinformatics/bty560

Chi HW, Yang YS, Shang ST et al (2011) Candida albicans versus non-albicans bloodstream infections: the comparison of risk factors and outcome. J Microbiol Immunol Infect 44:369–375. https://doi.org/10.1016/j.jmii.2010.08.010

Deamer D, Akeson M, Branton D (2016) Three decades of nanopore sequencing. Nat Biotechnol 34:518–524. https://doi.org/10.1038/nbt.3423

Ding J, Adiconis X, Simmons SK et al (2019) Systematic comparative analysis of single cell RNA-sequencing methods. bioRxiv 632216. https://doi.org/10.1101/632216

Dobin A, Davis CA, Schlesinger F et al (2013) STAR: ultrafast universal RNA-seq aligner. Bioinformatics 29:15–21. https://doi.org/10.1093/bioinformatics/bts635

Eid J, Fehr A, Gray J et al (2009) Real-time DNA sequencing from single polymerase molecules. Science 323(80):133–138. https://doi.org/10.1126/science.1162986

Gao S (2018) Data analysis in single-cell transcriptome sequencing. In: Methods in molecular biology. Humana Press, pp 311–326

Grabherr MG, Haas BJ, Yassour M et al (2011) Full-length transcriptome assembly from RNA-Seq data without a reference genome. Nat Biotechnol 29:644–652. https://doi.org/10.1038/nbt.1883

Hagemann-Jensen M, Ziegenhain C, Chen P et al (2020) Single-cell RNA counting at allele and isoform resolution using Smart-seq3. Nat Biotechnol 38:708–714. https://doi.org/10.1038/s41587-020-0497-0

Hao Y, Hao S, Andersen-Nissen E et al (2020) Integrated analysis of multimodal single-cell data. bioRxiv:2020.10.12.335331. https://doi.org/10.1101/2020.10.12.335331

Hardcastle TJ, Kelly KA (2010) BaySeq: empirical Bayesian methods for identifying differential expression in sequence count data. BMC Bioinf 11. https://doi.org/10.1186/1471-2105-11-422

Hashimshony T, Senderovich N, Avital G et al (2016) CEL-Seq2: sensitive highly-multiplexed single-cell RNA-Seq. Genome Biol 17. https://doi.org/10.1186/s13059-016-0938-8

Hoffmann S, Otto C, Kurtz S et al (2009) Fast mapping of short sequences with mismatches, insertions and deletions using index structures. PLoS Comput Biol 5. https://doi.org/10.1371/journal.pcbi.1000502

Hoffmann S, Otto C, Doose G et al (2014) A multi-split mapping algorithm for circular RNA, splicing, trans-splicing and fusion detection. Genome Biol 15. https://doi.org/10.1186/gb-2014-15-2-r34

Hu Z, Lyu T, Yan C et al (2020) Identification of alternatively spliced gene isoforms and novel non-coding RNAs by single-molecule long-read sequencing in Camellia. RNA Biol 17:966–976. https://doi.org/10.1080/15476286.2020.1738703

Hünniger K, Lehnert T, Bieber K et al (2014) A virtual infection model quantifies innate effector mechanisms and Candida albicans immune escape in human blood. PLoS Comput Biol 10. https://doi.org/10.1371/journal.pcbi.1003479

Ip CLC, Loose M, Tyson JR et al (2015) MinION Analysis and Reference Consortium: phase 1 data release and analysis. F1000 Res 4. https://doi.org/10.12688/f1000research.7201.1

Islam S, Zeisel A, Joost S et al (2014) Quantitative single-cell RNA-seq with unique molecular identifiers. Nat Methods 11:163–166. https://doi.org/10.1038/nmeth.2772

Kanzi AM, San JE, Chimukangara B et al (2020) Next generation sequencing and bioinformatics analysis of family genetic inheritance. Front Genet 11. https://doi.org/10.3389/fgene.2020.544162

Katz Y, Wang ET, Airoldi EM, Burge CB (2010) Analysis and design of RNA sequencing experiments for identifying isoform regulation. Nat Methods 7:1009–1015. https://doi.org/10.1038/nmeth.1528

Kim D, Pertea G, Trapnell C et al (2013) TopHat2: accurate alignment of transcriptomes in the presence of insertions, deletions and gene fusions. Genome Biol 14:R36. https://doi.org/10.1186/gb-2013-14-4-r36

Klein AM, Mazutis L, Akartuna I et al (2015) Droplet barcoding for single-cell transcriptomics applied to embryonic stem cells. Cell 161:1187–1201. https://doi.org/10.1016/j.cell.2015.04.044

Köster J, Rahmann S (2012) Snakemake-a scalable bioinformatics workflow engine. Bioinformatics 28:2520–2522. https://doi.org/10.1093/bioinformatics/bts480

Kovaka S, Zimin AV, Pertea GM et al (2019) Transcriptome assembly from long-read RNA-seq alignments with StringTie2. Genome Biol 20. https://doi.org/10.1186/s13059-019-1910-1

Kuhn RM, Haussler D, James Kent W (2013) The UCSC genome browser and associated tools. Brief Bioinform 14:144–161. https://doi.org/10.1093/bib/bbs038

Kvam VM, Liu P, Yaqing S (2012) A comparison of statistical methods for detecting differentially expressed genes from RNA-seq data. Am J Bot 99:248–256. https://doi.org/10.3732/ajb.1100340

Lander ES, Linton LM, Birren B et al (2001) Initial sequencing and analysis of the human genome. Nature 409:860–921. https://doi.org/10.1038/35057062

Langmead B, Salzberg SL (2012) Fast gapped-read alignment with Bowtie 2. Nat Methods 9:357–359. https://doi.org/10.1038/nmeth.1923

Langmead B, Trapnell C, Pop M, Salzberg SL (2009) Ultrafast and memory-efficient alignment of short DNA sequences to the human genome. Genome Biol 10. https://doi.org/10.1186/gb-2009-10-3-r25

Leśniewska A, Okoniewski MJ (2011) rnaSeqMap: a bioconductor package for RNA sequencing data exploration. BMC Bioinf 12. https://doi.org/10.1186/1471-2105-12-200

Li H (2018) Minimap2: pairwise alignment for nucleotide sequences. Bioinformatics 34:3094–3100. https://doi.org/10.1093/bioinformatics/bty191

Li H, Handsaker B, Wysoker A et al (2009) The sequence alignment/map format and SAMtools. Bioinformatics 25:2078–2079. https://doi.org/10.1093/bioinformatics/btp352

Li J, Witten DM, Johnstone IM, Tibshirani R (2012) Normalization, testing, and false discovery rate estimation for RNA-sequencing data. Biostatistics 13:523–538. https://doi.org/10.1093/biostatistics/kxr031

Li F, Luo M, Zhou W et al (2020) Single cell RNA and immune repertoire profiling of COVID-19 patients reveal novel neutralizing antibody. Protein Cell. https://doi.org/10.1007/s13238-020-00807-6

Logsdon GA, Vollger MR, Eichler EE (2020) Long-read human genome sequencing and its applications. Nat Rev Genet 21:597–614. https://doi.org/10.1038/s41576-020-0236-x

Lu H, Giordano F, Ning Z (2016) Oxford nanopore MinION sequencing and genome assembly. Genomics Proteomics Bioinf 14:265–279. https://doi.org/10.1016/j.gpb.2016.05.004

Luecken MD, Theis FJ (2019) Current best practices in single-cell RNA-seq analysis: a tutorial. Mol Syst Biol 15. https://doi.org/10.15252/msb.20188746

Macosko EZ, Basu A, Satija R et al (2015) Highly parallel genome-wide expression profiling of individual cells using nanoliter droplets. Cell 161:1202–1214. https://doi.org/10.1016/j.cell.2015.05.002

Magen A, Nie J, Ciucci T et al (2019) Single-cell profiling defines transcriptomic signatures specific to tumor-reactive versus virus-responsive CD4+ T cells. Cell Rep 29:3019–3032.e6. https://doi.org/10.1016/j.celrep.2019.10.131

Maranhão AQ, Silva HM, da Silva WMC et al (2020) Discovering selected antibodies from deep-sequenced phage-display antibody library using ATTILA. Bioinf Biol Insights 14. https://doi.org/10.1177/1177932220915240

Marr KA, Patterson T, Denning D (2002) Aspergillosis pathogenesis, clinical manifestations, and therapy. Infect Dis Clin N Am 16:875–894. https://doi.org/10.1016/S0891-5520(02)00035-1

Marsh M, Tu O, Dolnik V et al (1997) High-throughput DNA sequencing on a capillary array electrophoresis system. J Capillary Electrophor 4:83–89

Martin M (2011) Cutadapt removes adapter sequences from high-throughput sequencing reads. EMBnet J 17:10. https://doi.org/10.14806/ej.17.1.200

Martínez-Álvarez JA, Pérez-García LA, Flores-Carreón A, Mora-Montes HM (2014) The immune response against Candida spp. and Sporothrix schenckii. Rev Iberoam Micol 31:62–66. https://doi.org/10.1016/j.riam.2013.09.015

Metzker ML (2010) Sequencing technologies the next generation. Nat Rev Genet 11:31–46

Miceli MH, Díaz JA, Lee SA (2011) Emerging opportunistic yeast infections. Lancet Infect Dis 11:142–151. https://doi.org/10.1016/S1473-3099(10)70218-8

Miramón P, Kasper L, Hube B (2013) Thriving within the host: Candida spp. interactions with phagocytic cells. Med Microbiol Immunol 202:183–195. https://doi.org/10.1007/s00430-013-0288-z

Nakano K, Shiroma A, Shimoji M et al (2017) Advantages of genome sequencing by long-read sequencer using SMRT technology in medical area. Hum Cell 30:149–161. https://doi.org/10.1007/s13577-017-0168-8

Nattestad M, Goodwin S, Ng K et al (2018) Complex rearrangements and oncogene amplifications revealed by long-read DNA and RNA sequencing of a breast cancer cell line. Genome Res 28:1126–1135. https://doi.org/10.1101/gr.231100.117

Parekh S, Ziegenhain C, Vieth B et al (2018) zUMIs – a fast and flexible pipeline to process RNA sequencing data with UMIs. Gigascience 7. https://doi.org/10.1093/gigascience/giy059

Park JH, Lee HK (2020) Re-analysis of single cell transcriptome reveals that the NR3C1-CXCL8-neutrophil axis determines the severity of COVID-19. Front Immunol 11. https://doi.org/10.3389/fimmu.2020.02145

Pereira R, Oliveira J, Sousa M (2020) Bioinformatics and computational tools for next-generation sequencing analysis in clinical genetics. J Clin Med 9:132. https://doi.org/10.3390/jcm9010132

Picelli S, Faridani OR, Björklund ÅK et al (2014) Full-length RNA-seq from single cells using Smart-seq2. Nat Protoc 9:171–181. https://doi.org/10.1038/nprot.2014.006

Quinlan AR, Hall IM (2010) BEDTools: a flexible suite of utilities for comparing genomic features. Bioinformatics 26:841–842. https://doi.org/10.1093/bioinformatics/btq033

Ramsköld D, Luo S, Wang YC et al (2012) Full-length mRNA-Seq from single-cell levels of RNA and individual circulating tumor cells. Nat Biotechnol 30:777–782. https://doi.org/10.1038/nbt.2282

Rapaport F, Khanin R, Liang Y et al (2013) Comprehensive evaluation of differential gene expression analysis methods for RNA-seq data. Genome Biol 14. https://doi.org/10.1186/gb-2013-14-9-r95

Rhoads A, Au KF (2015) PacBio sequencing and its applications. Genomics Proteomics Bioinf 13:278–289. https://doi.org/10.1016/j.gpb.2015.08.002

Richardson M, Lass-Flörl C (2008) Changing epidemiology of systemic fungal infections. Clin Microbiol Infect 14:5–24. https://doi.org/10.1111/j.1469-0691.2008.01978.x

Robertson G, Schein J, Chiu R et al (2010) De novo assembly and analysis of RNA-seq data. Nat Methods 7:909–912. https://doi.org/10.1038/nmeth.1517

Robinson MD, McCarthy DJ, Smyth GK (2010) edgeR: a Bioconductor package for differential expression analysis of digital gene expression data. Bioinformatics 26:139–140. https://doi.org/10.1093/bioinformatics/btp616

Robles JA, Qureshi SE, Stephen SJ et al (2012) Efficient experimental design and analysis strategies for the detection of differential expression using RNA-Sequencing. BMC Genomics 13. https://doi.org/10.1186/1471-2164-13-484

Sanger F, Nicklen S, Coulson AR (1977) DNA sequencing with chain-terminating inhibitors. Proc Natl Acad Sci U S A 74:5463–5467. https://doi.org/10.1073/pnas.74.12.5463

Schmieder R, Edwards R (2011) Quality control and preprocessing of metagenomic datasets. Bioinformatics 27:863–864. https://doi.org/10.1093/bioinformatics/btr026

Sedlazeck FJ, Rescheneder P, Von Haeseler A (2013) NextGenMap: fast and accurate read mapping in highly polymorphic genomes. Bioinformatics 29:2790–2791. https://doi.org/10.1093/bioinformatics/btt468

Shendure J (2008) The beginning of the end for microarrays? Nat Methods 5:585–587. https://doi.org/10.1038/nmeth0708-585

Shigemura K, Osawa K, Jikimoto T et al (2014) Comparison of the clinical risk factors between Candida albicans and Candida non-albicans species for bloodstream infection. J Antibiot (Tokyo) 67:311–314. https://doi.org/10.1038/ja.2013.141

Simpson JT, Wong K, Jackman SD et al (2009) ABySS: a parallel assembler for short read sequence data. Genome Res 19:1117–1123. https://doi.org/10.1101/gr.089532.108

Smith TF, Waterman MS (1981) Identification of common molecular subsequences. J Mol Biol 147:195–197. https://doi.org/10.1016/0022-2836(81)90087-5

Smith LM, Sanders JZ, Kaiser RJ et al (1986) Fluorescence detection in automated DNA sequence analysis. Nature 321:674–679. https://doi.org/10.1038/321674a0

Smyth GK (2004) Linear models and empirical Bayes methods for assessing differential expression in microarray experiments. Stat Appl Genet Mol Biol 3. https://doi.org/10.2202/1544-6115.1027

Soneson C, Delorenzi M (2013) A comparison of methods for differential expression analysis of RNA-seq data. BMC Bioinf 14. https://doi.org/10.1186/1471-2105-14-91

Soon WW, Hariharan M, Snyder MP (2013) High-throughput sequencing for biology and medicine. Mol Syst Biol 9. https://doi.org/10.1038/msb.2012.61

Tang S, Riva A (2013) PASTA: Splice junction identification from RNA-Sequencing data. BMC Bioinf 14. https://doi.org/10.1186/1471-2105-14-116

Tang F, Barbacioru C, Wang Y et al (2009) mRNA-Seq whole-transcriptome analysis of a single cell. Nat Methods 6:377–382. https://doi.org/10.1038/nmeth.1315

Tierney L, Linde J, Müller S et al (2012) An interspecies regulatory network inferred from simultaneous RNA-seq of Candida albicans invading innate immune cells. Front Microbiol 3. https://doi.org/10.3389/fmicb.2012.00085

Trapnell C, Pachter L, Salzberg SL (2009) TopHat: discovering splice junctions with RNA-Seq. Bioinformatics 25:1105–1111. https://doi.org/10.1093/bioinformatics/btp120

Trapnell C, Williams BA, Pertea G et al (2010) Transcript assembly and quantification by RNA-Seq reveals unannotated transcripts and isoform switching during cell differentiation. Nat Biotechnol 28:511–515. https://doi.org/10.1038/nbt.1621

Trapnell C, Hendrickson DG, Sauvageau M et al (2013) Differential analysis of gene regulation at transcript resolution with RNA-seq. Nat Biotechnol 31:46–53. https://doi.org/10.1038/nbt.2450

Van Verk MC, Hickman R, Pieterse CMJ, Van Wees SCM (2013) RNA-Seq: revelation of the messengers. Trends Plant Sci 18:175–179. https://doi.org/10.1016/j.tplants.2013.02.001

Wagner GP, Kin K, Lynch VJ (2012) Measurement of mRNA abundance using RNA-seq data: RPKM measure is inconsistent among samples. Theory Biosci 131:281–285. https://doi.org/10.1007/s12064-012-0162-3

Wang Z, Gerstein M, Snyder M (2009) RNA-Seq: a revolutionary tool for transcriptomics. Nat Rev Genet 10:57–63. https://doi.org/10.1038/nrg2484

Wang B, Kumar V, Olson A, Ware D (2019) Reviving the transcriptome studies: an insight into the emergence of single-molecule transcriptome sequencing. Front Genet 10. https://doi.org/10.3389/fgene.2019.00384

Wang XJ, Jiao Y, Ma S et al (2020) Whole-genome sequencing: an effective strategy for insertion information analysis of foreign genes in transgenic plants. Front Plant Sci 11. https://doi.org/10.3389/fpls.2020.573871

Weirather JL, Afshar PT, Clark TA et al (2015) Characterization of fusion genes and the significantly expressed fusion isoforms in breast cancer by hybrid sequencing. Nucleic Acids Res:43. https://doi.org/10.1093/nar/gkv562

Wenger AM, Peluso P, Rowell WJ et al (2019) Accurate circular consensus long-read sequencing improves variant detection and assembly of a human genome. Nat Biotechnol 37:1155–1162. https://doi.org/10.1038/s41587-019-0217-9

Wercelens P, da Silva W, Hondo F et al (2019) Bioinformatics workflows with NoSQL database in cloud computing. Evol Bioinforma 15. https://doi.org/10.1177/1176934319889974

Yu G, Wang LG, Han Y, He QY (2012) ClusterProfiler: an R package for comparing biological themes among gene clusters. Omi A J Integr Biol 16:284–287. https://doi.org/10.1089/omi.2011.0118

Zhang J, Su L, Wang Y, Deng S (2020a, 2020) Improved high-throughput sequencing of the human oral microbiome: from illumina to PacBio. Can J Infect Dis Med Microbiol. https://doi.org/10.1155/2020/6678872

Zhang YZ, Akdemir A, Tremmel G et al (2020b) Nanopore basecalling from a perspective of instance segmentation. BMC Bioinf 21. https://doi.org/10.1186/s12859-020-3459-0

Zhao L, Zhang H, Kohnen MV et al (2019) Analysis of transcriptome and epitranscriptome in plants using pacbio iso-seq and nanopore-based direct RNA sequencing. Front Genet:10. https://doi.org/10.3389/fgene.2019.00253

Zheng GXY, Terry JM, Belgrader P et al (2017) Massively parallel digital transcriptional profiling of single cells. Nat Commun 8. https://doi.org/10.1038/ncomms14049

Zimin AV, Marçais G, Puiu D et al (2013) The MaSuRCA genome assembler. Bioinformatics 29:2669–2677. https://doi.org/10.1093/bioinformatics/btt476