Introduction to sequencing: Sequencing technology and preprocessing of sequencing data

Koen Van den Berge

6/25/2021

Contents

1 The study of gene expression

The first part of this course has focussed on proteomics, studying the concentration of proteins in biological samples. We have seen that identification of proteins and measuring their respective concentrations are extremely challenging, leading to many technological and statistical challenges in order to interpret these data. In the second part of the course, we will now focus on measuring gene expression, i.e., measuring the concentration of mRNA molecules, that may eventually be translated into proteins, but may also have functions on their own.

2 Sequencing technology

- Measuring mRNA molecules typically happens through sequencing.
- The technology continues to evolve at an incredible speed. The data output of so-called 'next generation sequencing' machines has more than doubled each year! Simultaneously, the cost of sequencing (in terms of $ per Gigabase) is dropping. Each year, we're able to sequence more for less money.
- This tremendous technological revolution has revolutionized biology, and genomic sequencing is now a core component of the modern-day biologist's toolkit.

- The large majority of sequencing data is generated using sequencing-by-synthesis using machines produced by the company Illumina. While new players such as Pacific Biosciences and Oxford Nanopore have entered the scene, these are typically most useful for DNA sequencing rather than gene expression studies, owing to their capability of sequencing long molecules.

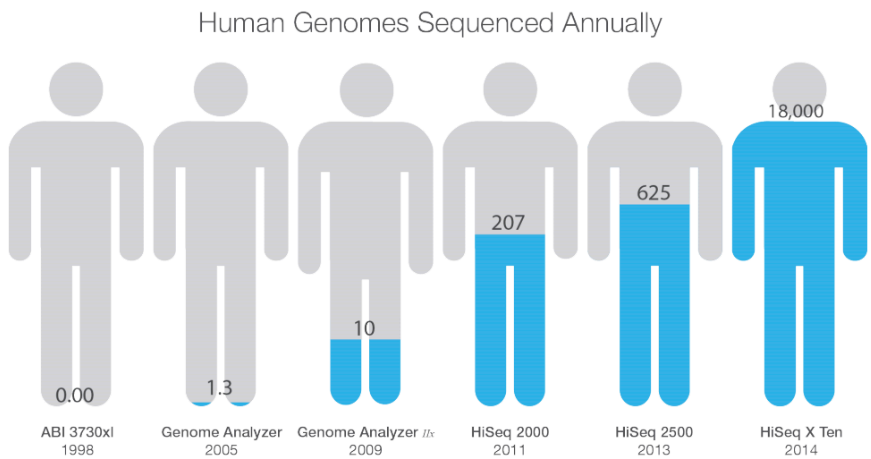

Figure 2: Human Genome Sequencing Over the Decades—The capacity to sequence all 3.2 billion bases of the human genome (at 30× coverage) has increased exponentially since the 1990s. In 2005, with the introduction of the Illumina Genome Analyzer System, 1.3 human genomes could be sequenced annually. Nearly 10 years later, with the Illumina HiSeq X Ten fleet of sequencing systems, the number has climbed to 18,000 human genomes a year.

Figure 1: Figure: The data output revolution of sequencing machines. Image from Illumina documentation.

2.1 The sequencing workflow

Library preparation steps:

1. First, the biological samples of interest are collected. Owing to the maturity of different protocols for sequencing, several types of biological input samples are amenable to sequencing, such as frozen tissues or FFPE-preserved samples.
2. The mRNA molecules from our sample are captured. If the tissue sample used as input is still intact, this also involves cell lysis in order to release the mRNA molecules from within the cells. The mRNA molecules are mosten often captured using (i) polyA-capture to select for polyadenylated RNA, or (ii) ribosomal deplation, where ribosomal and transfer RNAs are depleted. In the case of 'targeted sequencing', where relevant molecules are of main interest (e.g., a gene panel), these targets can be specifically targeted in this step.
3. Fragmentation of captured molecules. The captured molecules are fragmented, either chemically or mechanically. The appropriate size of fragments depends on the sequencing machines (often in the range of 300 - 500bp).
4. Reverse transcription. Current dominant sequencing machines only sequence double-stranded DNA molecules. Therefore, in order to measure single-stranded mRNA, we must first isolate the mRNA molecules from the biological sample(s) of interest, and reverse transcribe these molecules to a double-stranded complementary (cDNA) molecule.
5. Addition of adapters. Adapters are oligonucleotides (short sequences of nucleotides) that are platform-specific sequences for fragment recognition by the sequencing machines. These are added either to the 3' or 5' end of the cDNA molecules or used as primers in the reverse transcription reaction. The final cDNA library consists of cDNA inserts flanked by an adapter sequence on each end.
6. PCR amplification. To increase concentration, several PCR reactions are performed.

7. Loading the amplified cDNA library on the sequencing machine. Find out how sequencing-by-synthesis works through this video. Note that the video shows paired-end sequencing, where a number of basepairs are sequenced at each end of the fragment. All previous steps together are described as 'sample prep' in that video.

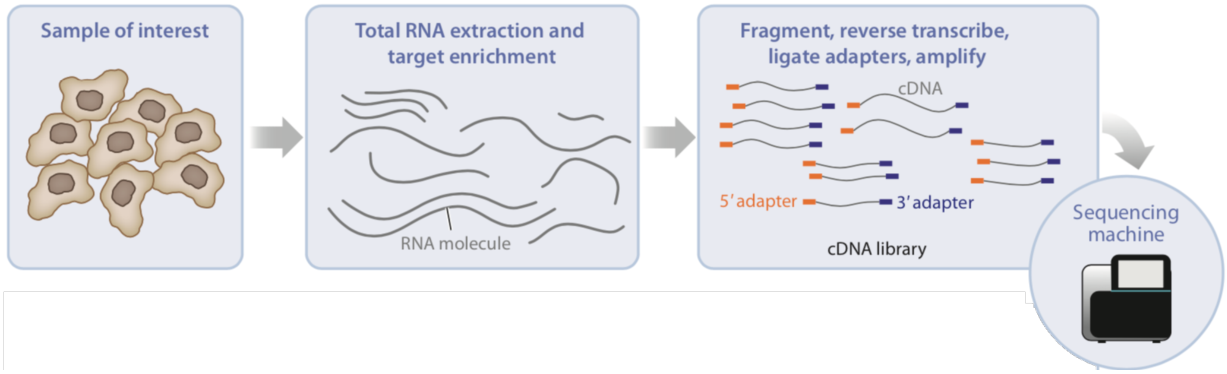

Figure 2: Figure: The sequencing workflow. Image adapted from Van den Berge et al. (2019).

Note that several variants of library preparation protocols are available. The most important ones are:

- Single-end vs paired-end sequecing: In single-end sequencing, a single end (3' or 5') of the cDNA fragment is sequenced. In paired-end sequencing, both ends are sequenced.
- Strand specific protocols: Some library preparation protocols allow measuring strand specificity, where the strand information of each read can be preserved.

2.2 The sequencing output files

- The typical output of a sequencing machine we will be working with are FASTA or FASTQ files for each sample. Each of these files are several gigbases large and contain millions of sequences, which we will call **reads**. For paired-end sequencing, there are two files for each sample, one for each end of the sequenced fragments.
- The difference between a FASTA file and a FASTQ file, is that while FASTA files only store the results of base calls (sequences), FASTQ files also store the quality score of each base call (i.e., each called nucleotide), which can be useful in downstream analyses such as mapping or variant calling.
- A fASTQ file contains four lines for each sequenced read:
    1. Sequence identifier line, starting with @.
    2. The sequence.
    3. Another sequence identifier line, now starting with +.
    4. Quality scores.

As you'll have noticed, the base call quality scores are encoded as ASCII characters for efficient storage. These ASCII characters can be convered into integers called Phred scores, which are logarithmically related to the probability of an erroneous base call.

3 Preprocessing of raw sequencing data

After sequencing, we typically do a quality control (QC) check to verify the quality of the samples. During QC check, aberrant samples due to e.g. degraded mRNA can be detected.

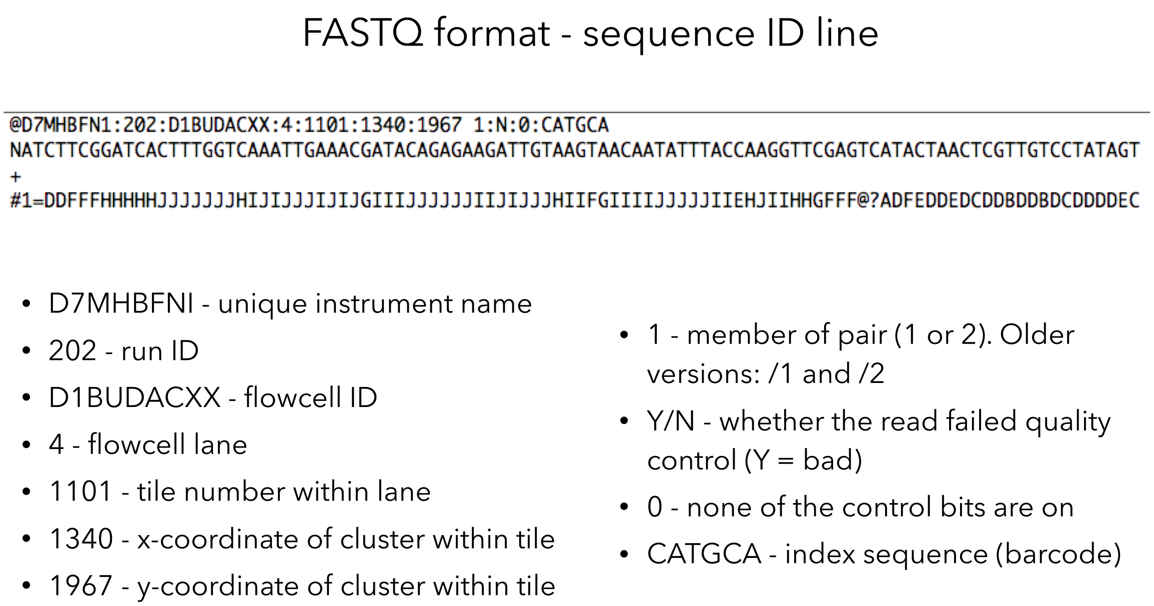

Figure 3: Figure: One read in a FASTQ file. Slide courtesy by Charlotte Soneson.

The sequencing reads on their own contain a lot of information, but are most useful if we would be able to assign sequencing reads to genomic features (genes, exons, transcripts, etc.), i.e., for each sequencing read we will try to derive the (set of) feature(s) that could have plausibly produced the fragment through the process of gene expression. This process is called 'mapping'. Most often we map reads to genes.

3.1 Quality control

During quality control, diagnostic plots are created for each sample in order to determine its quality. The most popular QC tool for bulk RNA-seq data is FastQC. If many samples are sequenced, then MultiQC can be used to aggregate the QC checks across samples in a conveniently organized overview.

The FastQC website provides interesting example reports for us to look at and compare against. Here are example reports of high-quality Illumina data and low-quality Illumina data.

3.2 Mapping

Mapping is a critical step in the interpretation of RNA-seq data, where we are attributing reads to genomic features. This attribution eventually allows us to measure how strong a feature such as a gene is expressed. Indeed, the number of reads mapping to a gene serve as a proxy for how high that gene has been expressed in the sample. This opens the door to downstream data analysis, but of course relies on an appropriate mapping.

Note that we are typically not able to assign each individual read uniquely to one specific gene; some reads cannot be unambiguously mapped and are compatible with multiple genes. These reads are said to be 'multi-mapping'.

While here we will focus on reference-based alignment, i.e., alignment where a reference genome or transcriptome is available, note that a *de novo* construction of a reference transcriptome is also possible, where the reference may be constructed from the observed sequencing reads.

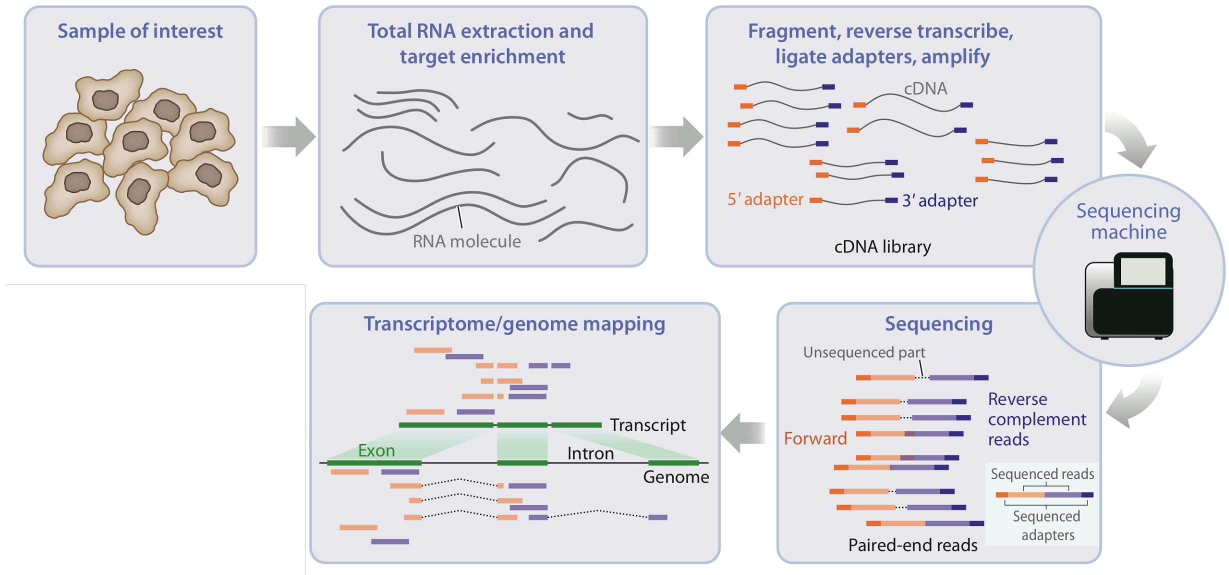

Figure 4: Figure: An updated sequencing workflow, including sequencing and mapping. Image adapted from Van den Berge et al. (2019).

Finally, a note on terminology. In this text we will use the words 'read' or 'fragment' (referring to the fragmented mRNA molecule being sequenced) to designate a datum, note that this could be either a single read (in single-end sequencing) or a read pair (in paired-end sequencing). The literature may also use these words interchangably, although 'fragment' seems better at avoiding ambiguity between single-end reads and paired-end read pairs.

3.2.1 Reference files

- The alignment most often relies on a reference genome of the species, which can be considered a 'representative example' of the genome sequence of that species. Reference genomes are contiuously updated and released periodically.
- Reference genomes can be freely downloaded from several providers, for example Ensembl or Gencode.
- Along with a reference genome, an annotation GFF or GTF file defines the coordinates of specific genomic features.

- More recently, mapping of RNA-seq data occurs more often against a reference transcriptome, which is a reference file containing the sequences all known isoforms of a particular species, e.g., using kallisto or Salmon.
- The set of spliced transcripts is much smaller than the entire genome, and therefore mapping against a reference transcriptome is typically fast and memory efficient.
- However, it has been noted that mapping against a reference transcriptome may also introduce spurious expression for genes that are not expressed. These observations can be explained by intronic reads that share some sequence similarity with transcripts, and could map to spliced transcript sequences. Recent methods, such as alevin-fry, avoid this by expanding the reference transcriptome to also include intronic sequences.

3.2.2 Alignment-based workflows

- Traditionally, alignment-based workflows have been used to map reads, where one tries to find the exact coordinates a read most likely maps on the reference genome or the reference transcriptome.

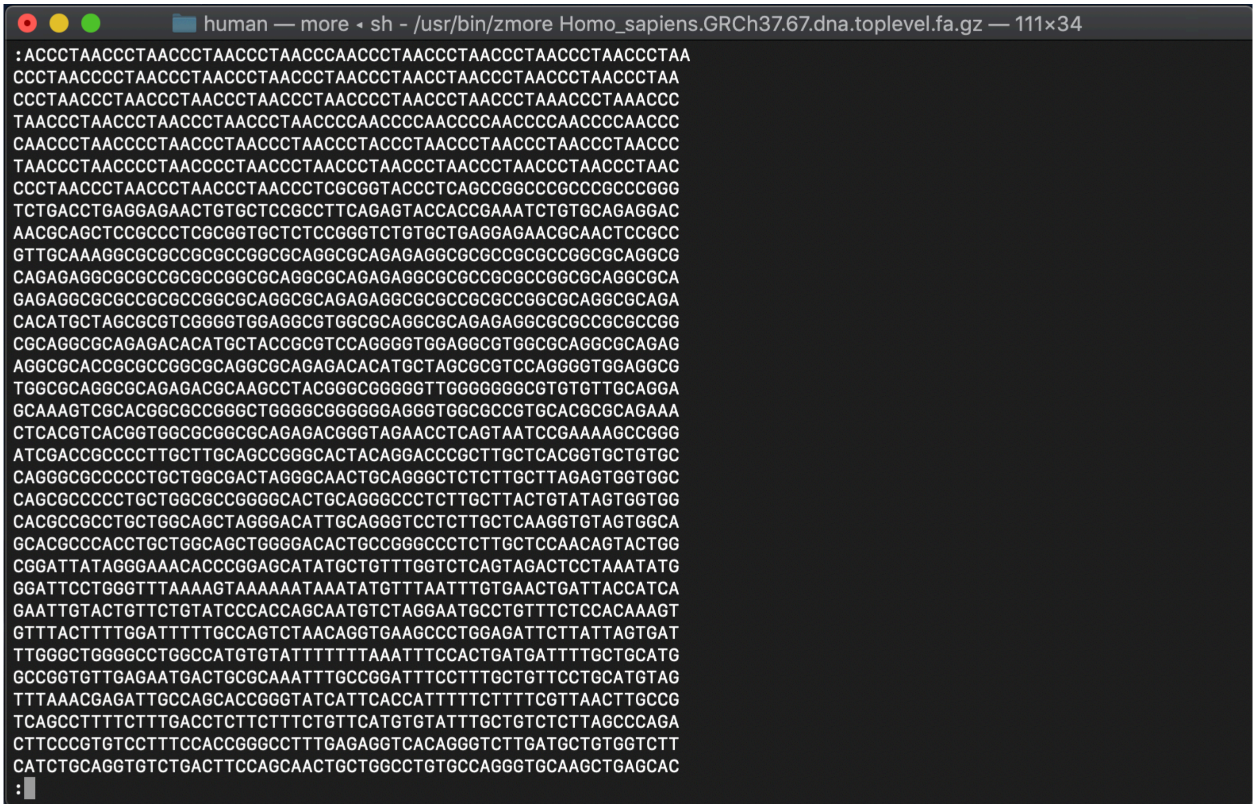

Figure 5: Figure: A reference sequence of human chr1.

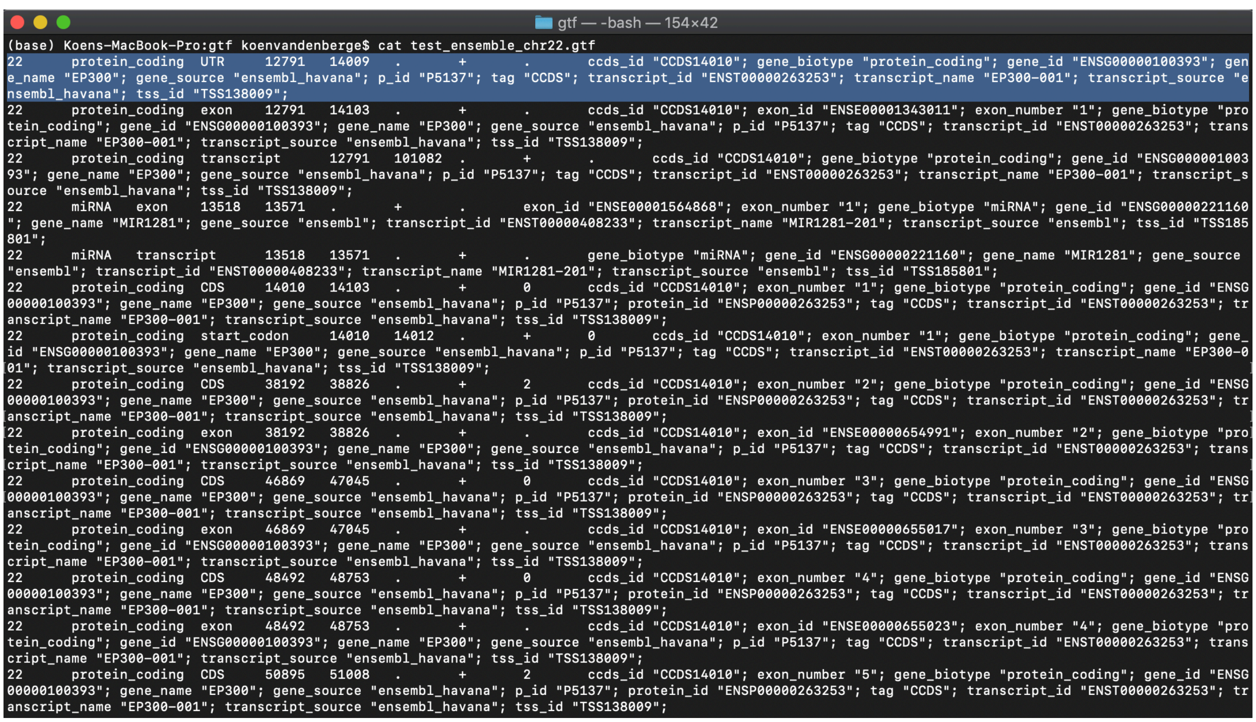

Figure 6: Figure: An example GTF file.

- Note that due to alternative splicing, reads do not necessarily map contiguously on a reference genome, as a read can overlap with a splicing junction, where an intron has been excised. When mapping against a transcriptome, however, reads should be mapping contiguously.
- A main challenge in spliced alignment against a reference genome is the proper alignment of reads that span a splice junction, especially when these junctions are not annotated a priori. Indeed, in spliced alignment reads can be split at any nucleotide, and the corresponding subsequences can map several thousands of basepairs apart. Meanwhile, the main challenge in unspliced alignment to a transcriptome is the redundant sequence among related transcripts in the transcriptome, which often leads to a high multi-mapping rate (i.e., reads that cannot be unambiguously assigned to a single transcript).
- Spliced alignment against a genome is therefore computationally a much harder task. Since the transcript sequences are already spliced when aligning to a reference transcriptome, reads should align contiguously, and many of the computationally expensive steps and heuristics can be avoided.

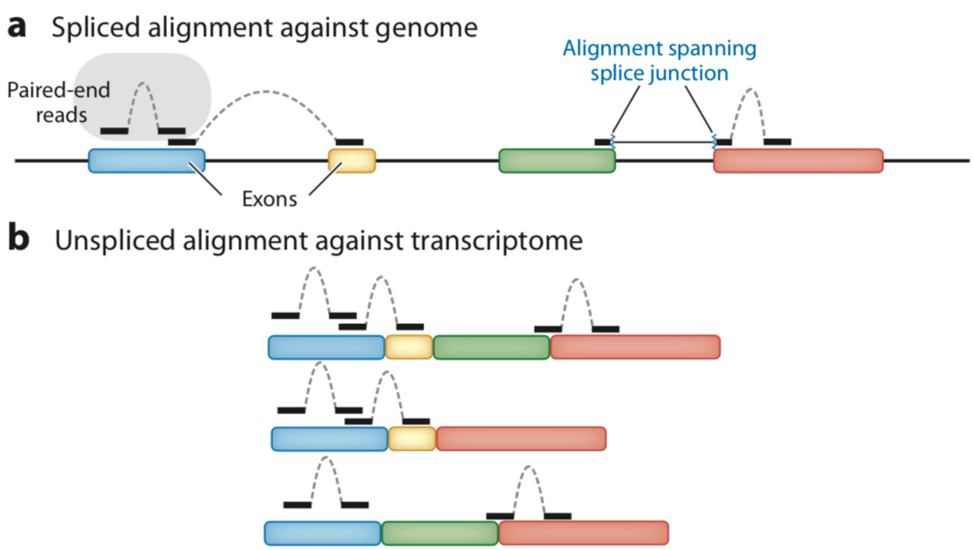

Figure 4

An illustration of spliced alignment of RNA sequencing (RNA-seq) fragments to a genome (*a*) and direct alignment to a transcriptome (*b*). Reads are designated by thick solid lines, while dashed arcs represent the pairing relationship between paired-end reads. This illustration depicts alignment to a single four-exon gene consisting of three distinct transcripts. In the spliced alignment (*a*), the left read of the rightmost pair is a junction-spanning alignment to the red–green exon boundary. In the direct alignment to the transcriptome (*b*), one observes how the same alignment (e.g., the alignment to the blue exon) is repeated for each transcript.

Figure 7: Figure: Unspliced and spliced alignment. Figure from Van den Berge et al. (2019).

3.2.3 Alignment-free workflows

- Modern approaches avoid mapping each fragment individually (i.e., do not attempt to find the exact coordinates of a read's origin), and instead posit a probabilistic model where transcript abundances are typically defined using its constituent k-mers. These methods are sometimes referred to as *lightweight*.
- A k-mer is a short sequence of nucleotides of length k. The space of possible k-mers and the corresponding transcripts can be precomputed in advance using the reference transcriptome, providing a computational advantage as it only needs to be computed once.
- For each fragment, the transcripts its k-mers are compatible with is searched for using an indexed (efficiently searchable) transcriptome. The set of compatible transcripts is called the 'k-compatibility

class', 'equivalence class' or 'transcript compatibility class' of the fragment.

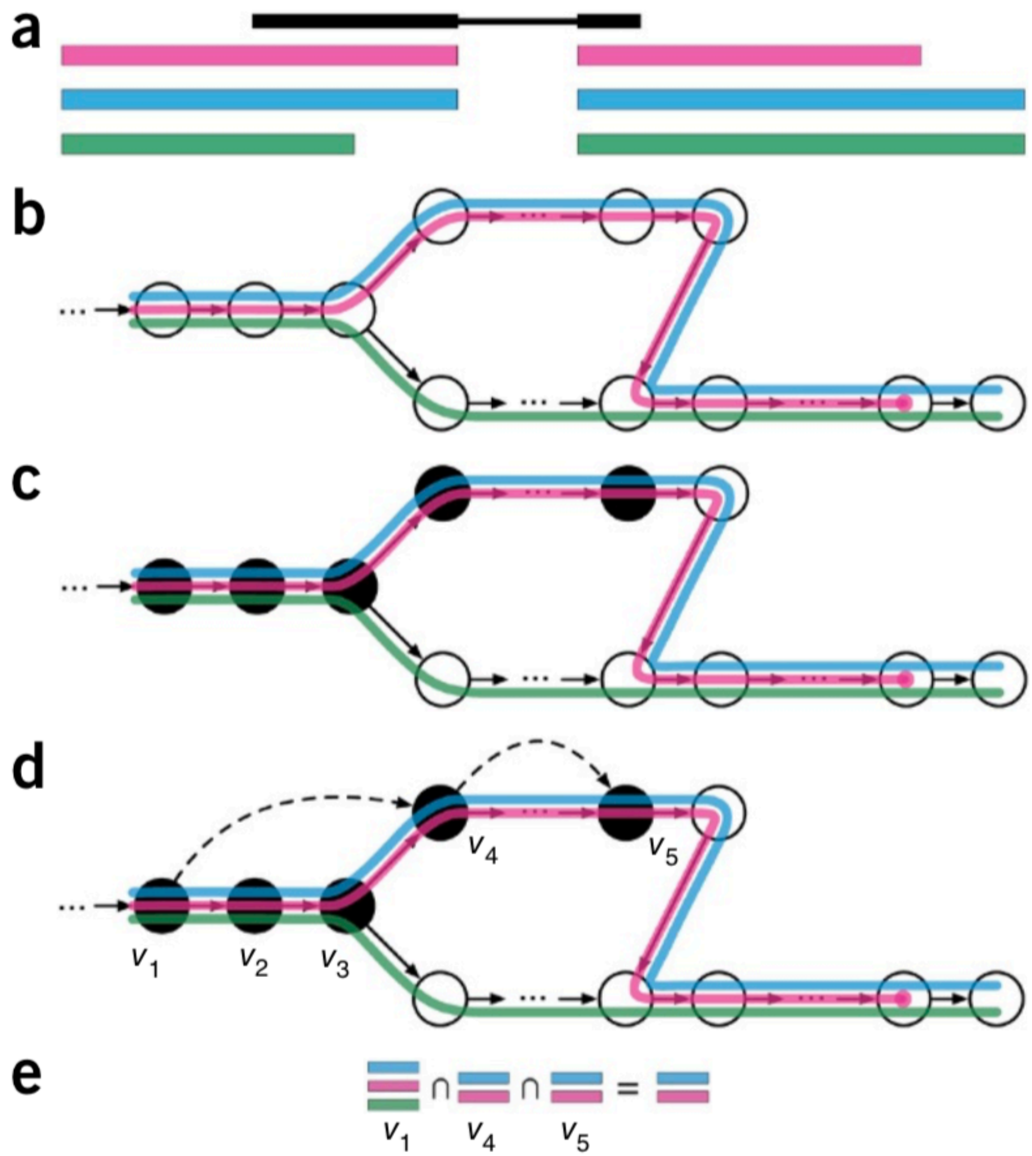

Figure 1 Overview of kallisto. The input consists of a reference transcriptome and reads from an RNA-seq experiment. (**a**) An example of a read (in black) and three overlapping transcripts with exonic regions as shown. (**b**) An index is constructed by creating the transcriptome de Bruijn Graph (T-DBG) where nodes (v_1, v_2, v_3, ...) are *k*-mers, each transcript corresponds to a colored path as shown and the path cover of the transcriptome induces a *k*-compatibility class for each *k*-mer. (**c**) Conceptually, the *k*-mers of a read are hashed (black nodes) to find the *k*-compatibility class of a read. (**d**) Skipping (black dashed lines) uses the information stored in the T-DBG to skip *k*-mers that are redundant because they have the same *k*-compatibility class. (**e**) The *k*-compatibility class of the read is determined by taking the intersection of the *k*-compatibility classes of its constituent *k*-mers.

3.3 Abundance quantification

Given a set of mappings, using either alignment-based or alignment-free workflows, the estimation of expression of a gene/transcript/exon may occur in several ways.

Counting:

- In alignment-based workflows, one could do a direct counting of fragments at the gene-level, counting the number fragments mapping to each gene. This has been the dominant approach for the first decade of RNA-seq data, often obtained using reference genome alignments.
- Many choices need to be made: Do we count a fragment as soon as it intersects with the gene's coordinates, or do we require the full fragment to map to the gene? Do we count intronic reads? Do we count multi-mapping reads?

Estimation:

- Abundance quantification is more recently starting to shift from counting towards using statistical models to estimate the expression counts for a feature, which in this case is typically a transcript.
- This approach is amenable to alignment-free workflows, since the number of fragments in each equivalence class are sufficient statistics for the abundance quantification, meaning that they contain all information needed to estimate the parameters of the statistical model, and hence the feature-level abundances. Since the expression counts in this case are estimated, they are not necessarily integer counts, and will be referred to as 'estimated counts'.
- In order to derive these, the EM-algorithm is often used, although other approaches have been used by tools like Salmon. A big advantage of the estimation approach is that it probabilistically assigns fragments to transcripts, thereby automatically dealing with multi-mapping reads. The total number of fragments mapping to each transcript is then the sum of all fragment-level probabilities to be assigned to that respective transcript.

3.3.1 Abundance metrics

- For simplicity, we have only been talking about feature-level counts as in sums of fragments. However, this is merely one metric that can be used as a proxy for expression, and several others exist.

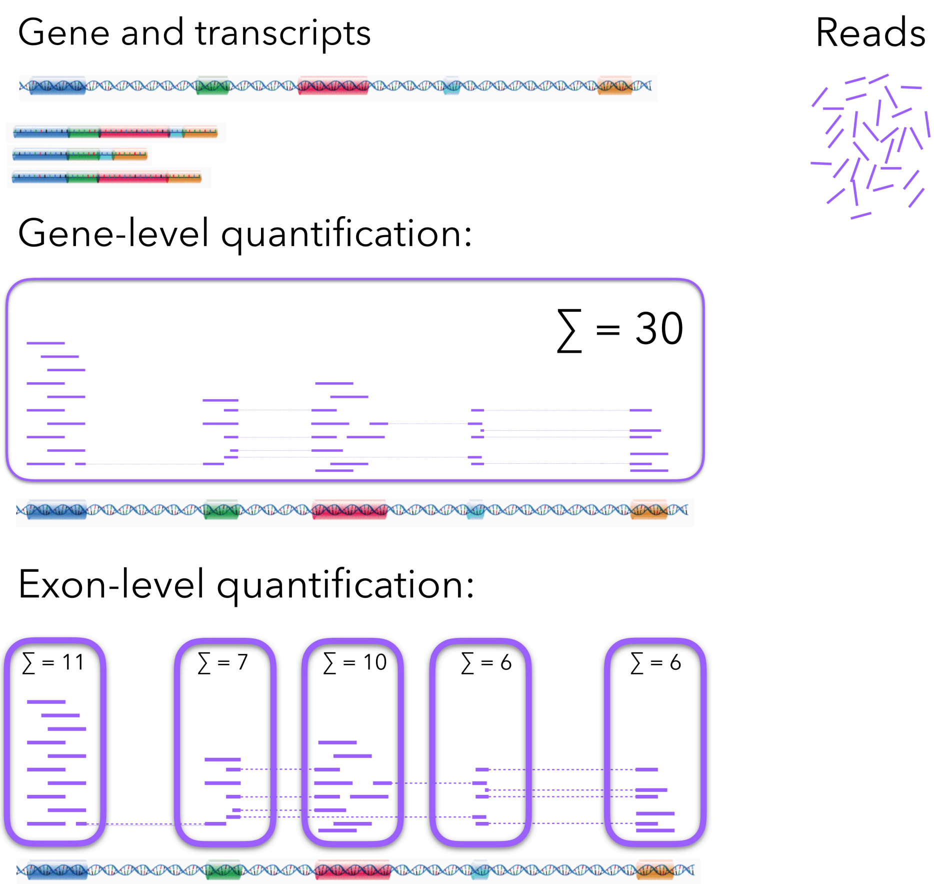

Figure 8: Figure: Gene- and exon-level read counting. Image adapted from Charlotte Soneson.

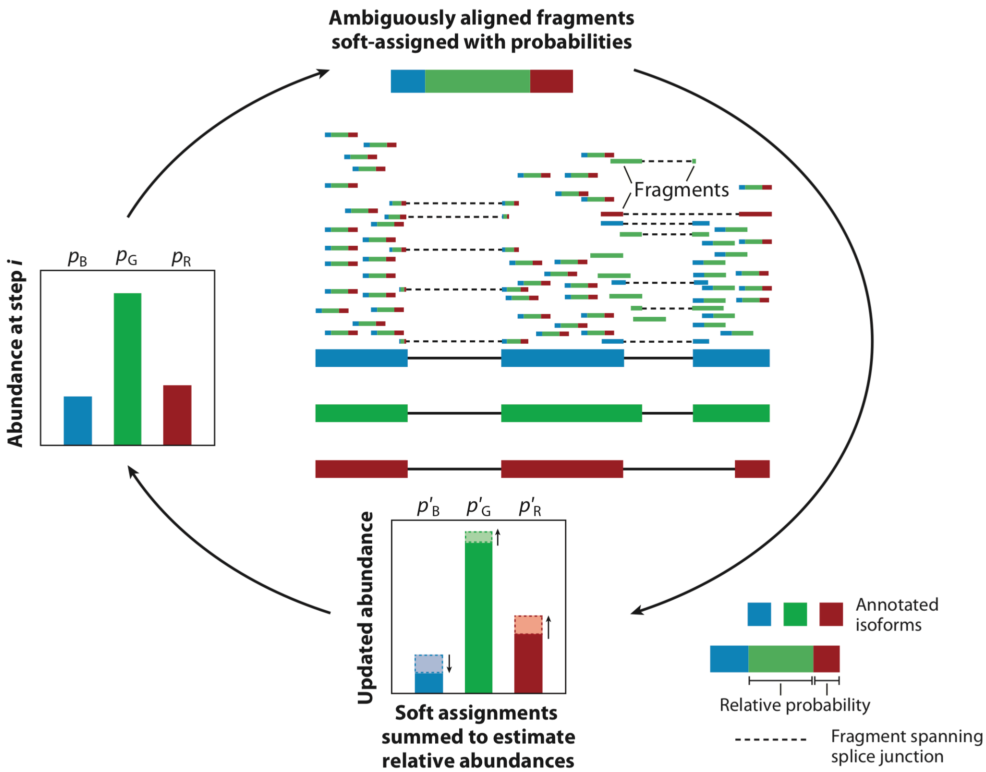

Figure 5

An illustration of the alignment of various reads to a gene with three isoforms: blue (B), green (G), and red (R). In this example, we wish to estimate the abundances of these isoforms, but most reads have ambiguous origins and need to be probabilistically assigned to the transcripts (relative probabilities for each read are shown by the magnitudes of the three colors). Some reads are consistent only with the B and G transcripts, and a few reads uniquely align to a single transcript (single color). In the expectation-maximization (or related) algorithm, given the current abundance estimates, fragments are probabilistically assigned to transcripts, and then estimated abundances are updated by summarizing the (proportional) allocations over all fragments; transcript abundance estimates are determined by iterating the procedure until convergence.

Figure 9: Figure: Abundance quantification using the EM algorithm. Figure from Van den Berge et al. (2019).

- Most of these were introduced to attempt to make the abundances more comparable across samples or features, as compared to the simple counts. These mainly serve to correct for technical biases such as transcript length and sequencing depth, both of which have significant impact on the observed counts.
- Indeed, for a gene with the same mRNA concentration in two samples, sequencing one sample deeper, will on average result in a higher count.
- Likewise, for two transcripts with the same mRNA concentration but different transcript lengths, one will tend to observe more fragments from the longer transcript due to the fragmentation step in the RNA-seq protocol, where longer transcripts can be split into more fragments of appropriate length.

Below we introduce several relevant abundance metrics, but note that most data analysis methods we will discuss in this course will work with (estimated) counts. In what follows, let Y_{fi} denote the random variable representing the expression counts of feature f in sample i (obtained either as a simple sum of fragments or estimated using lightweight approaches), and let $N_i = \sum_f Y_{fi}$ denote the sequencing depth of sample i.

- **Counts per million (CPM)** are the counts one could expect to observe if the sample was sequenced to a depth of one million.

$$CPM_{fi} = \frac{Y_{fi}}{N_i} 10^6$$

- **Transcripts per million (TPM)** refers to the concentration or proportion of your feature in the sample. TPMs take into account the length of the feature, which is often reformulated into an *effective length* $l_{fi}^{(eff)}$, relating to the number of possible start sites that a feature may have in order to generate fragments of a typical length observed in your dataset. This typical length is often calculated using the observed fragment length distribution from the data and defined as

$$l_{fi}^{(eff)} = l_f - \hat{\bar{F}}_i + 1,$$

where l_f is the total length of a feature in terms of number of nucleotides, and $\hat{\bar{F}}_i$ is the estimated average fragment length in sample i. We can use this to define

$$TPM_{fi} = \frac{Y_{fi}}{l_{fi}^{(eff)}} \left(\frac{1}{\sum_f \frac{Y_{fi}}{l_{fi}^{(eff)}}} \right) 10^6.$$

Note that the first part of the right-hand-side (RHS), $\frac{Y_{fi}}{l_{fi}^{(eff)}}$ is the expression counts normalized for the length of the feature. This measure, however, is still affected by the sequencing depth, which is then alleviated by dividing by the sum of the length-normalized counts across all features, i.e., $\sum_f \frac{Y_{fi}}{l_{fi}^{(eff)}}$. TPMs hence normalize for the feature length as well as sequencing depth.

3.3.2 The final countdown

Once abundances have been quantified, the (estimated) counts are typically stored in a count matrix, with genes spanning the rows and samples spanning the columns. This count matrix forms the basis of most downstream analyses to interpret RNA-seq data, and it will be the main object we will be working with in the following lectures.

4 References

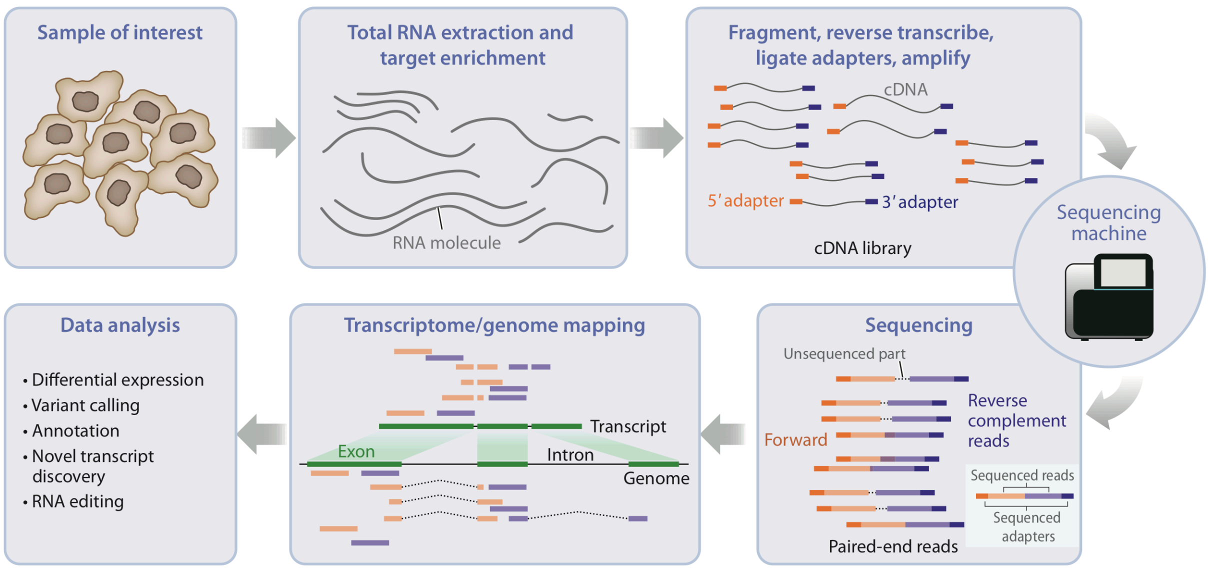

Figure 10: Figure: An updated sequencing workflow. Image adapted from Van den Berge et al. (2019).