Chapter 1

An Introduction to High-Throughput Sequencing Experiments: Design and Bioinformatics Analysis

Rachelly Normand and Itai Yanai

Abstract

The dramatic fall in the cost of DNA sequencing has revolutionized the experiments within reach in the life sciences. Here we provide an introduction for the domains of analyses possible using high-throughput sequencing, distinguishing between “counting” and “reading” applications. We discuss the steps in designing a high-throughput sequencing experiment, introduce the most widely used applications, and describe basic sequencing concepts. We review the various software programs available for many of the bioinformatics analysis required to make sense of the sequencing data. We hope that this introduction will be accessible to biologists with no previous background in bioinformatics, yet with a keen interest in applying the power of high-throughput sequencing in their research.

Key words RNA-Seq, ChIP-Seq, Resequencing, De novo genome assembly, Initial bioinformatics analysis

1 Introduction

High-throughput sequencing is the process of identifying the sequence of millions of short DNA fragments in parallel. In this chapter, we discuss applications and analyses of high-throughput sequencing done on the Illumina platform. The main advantage of this technology is that it allows a very high throughput; currently up to 1.6 billion DNA fragments can be sequenced in parallel in a single run, to produce a total of 320Gbp (HiSeq 2000, version three kits). One challenge with this technology, however, is that the sequenced fragments are relatively short—currently up to 250 bp (MiSeq instrument) or 150 bp (HiSeq 2500 instrument)—though double this can be produced using the paired-end option (see below).

We operate a service unit in a university setting providing high-throughput sequencing (henceforth, HTS) sample preparation, sequencing, and initial bioinformatics analysis. Based upon our experiences over the past 2 years we provide the following notes. We do not aim to provide a complete picture of all of the innumerable

Noam Shomron (ed.), *Deep Sequencing Data Analysis*, Methods in Molecular Biology, vol. 1038,
DOI 10.1007/978-1-62703-514-9_1, © Springer Science+Business Media New York 2013

resources available for any one of the described applications. Rather, our goal is to provide a basic overview of the opportunities and challenges that HTS represents. The field is clearly changing rapidly and so the details are to be taken with caution as they will surely need revision as new algorithms and technology emerge.

While many applications are supported by HTS, the actual input to the instrument is the same: libraries comprising billions of DNA strands of roughly the same length (typically 300 bp) with particular sequences (linkers) on either end. "Sample preparation" is the process by which an initial sample arrives at this highly ordered state. When genomic DNA is the starting material, it is fragmented and then size-selected for the tight size distribution. If the starting material is RNA, often times it is polyA-selected to limit the sequencing to mRNA. The RNA is reverse transcribed to DNA and then also size-selected. Irrespective of the application, linker DNA molecules of particular sequences are ligated to the ends of the strands. These consist of two fragments: adaptors and indices. The adaptors hybridize the DNA fragments to the flowcell on which they are sequenced. The indices are 6–7 bp sequences tagging different samples within the same library that will be sequenced together. Importantly there is a PCR amplification step in many of the sample preparation protocols which has implications for the structure of the data: identical sequences may be a result of the amplification or reflect recurrence in the original sample of DNA.

2 Materials

2.1 Basic Concepts in High-Throughput Sequencing

Figure 1 indicates the anatomy of an insert. The following are additional basic definitions important for HTS:

1. *Insert*—The DNA fragment that is used for sequencing.
2. *Read*—The part of the insert that is sequenced.
3. *Single Read* (SR)—A sequencing procedure by which the insert is sequenced from one end only.
4. *Paired End* (PE)—A sequencing procedure by which the insert is sequenced from both ends.
5. *Flowcell*—A small glass chip on which the DNA fragments are attached and sequenced. The flowcell is covered by probes that allow hybridization of the adaptors that were ligated to the DNA fragments.

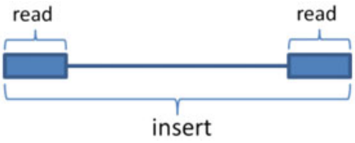


Fig. 1 Schematic of a paired-end read

6. *Lane*—The flowcell consists of eight physically separated channels called lanes. The sequencing is done in parallel on all lanes.
7. *Multiplexing/Demultiplexing*—Sequencing a few samples on the same lane is called multiplexing. The separation of reads that were sequenced on one lane to different samples is called demultiplexing and is done by a script that recognizes the index of each read and compares it to the known indices of each sample.
8. *Pipeline*—A series of computational processes.

3 Methods

3.1 High-Throughput Sequencing Applications

HTS applications can be divided into two main categories: "reading" and "counting." In reading applications the focus of the experiment is the sequence itself, for example, for finding genomic variants or assembling the sequence of an unknown genome. Counting applications are based on the ability to count amounts of reads and compare these counts, for example, to assess gene expression levels. Table 1 shows some of the main applications enabled by HTS. These represent but a sampling of the main HTS applications. It should be noted that one can invoke HTS in practically any experiment that produces DNA fragments. What should be considered and planned before the sequencing however is the method by which the analysis of the sequenced fragments will be done to extract the meaning from the experiment. As an example of a unique HTS experiment, chromatin interactions can be identified by PE sequencing [1]. This procedure includes capturing interacting loci in the genome by immune-precipitating cross-linked fragments of DNA and proteins from fixed cells. There are many others, published at a rate of about 1 per day.

3.2 Sequence Coverage

In reading applications, coverage corresponds to the number of reads that cover each base in the genome on average. Coverage can be calculated as

$$\text{Average coverage} = \frac{\text{read length} \times \text{number of reads}}{\text{genome size}}$$

Note that only the number of mapped reads should be included in the above calculation. In general, 30× coverage is considered a minimum for identifying genomic variants, while de novo assembly usually requires a much higher coverage. Furthermore, the needed coverage depends on the experiment design. For example, if resequencing is done on a population and the sample includes pooling of heterogenic genomes, the coverage must be higher for the robust detection of rare variants.

Table 1
HTS applications

	Application	Goal	Experiment details	Basic analysis summary
Reading	Resequencing	Find variants in a given sample relative to reference genome	Extract DNA from the relevant cells, conduct sample preparation consisting of DNA fragmentation and sequencing	Mapping of the sequenced fragment to the reference genome and identifying variants relative to the reference genome by summarizing the differences of the fragments from the genomic loci to which they map
	Target-enriched sequencing	Target enrichment sequencing is a specific form of resequencing that is focused only on certain genomic loci. This is useful for organisms with large genomes where enrichment increases the coverage on the loci of interest, thereby reducing costs	After the DNA is extracted from the cells and undergoes sample preparation, an enrichment process is done to capture the relevant loci Target enrichment can be done on specific regions of the genome using "tailored" target-enrichment probes, or by using available kits such as exome-enrichment kits	Same as in resequencing
	De novo assembly	Identify a genomic sequence without any additional reference	Same as in resequencing	The assembly process relies on overlaps of DNA fragments. These overlaps are merged into consensus sequences called contigs and scaffolds
Counting	ChIP-Seq/ RIP-Seq	Find the binding locations of RNA- or DNA-binding proteins	First, the ChIP/RIP experiment is done: Proteins are bound to the DNA/RNA and are cross-linked to it. The DNA/RNA is then fragmented. The proteins are pulled down by an	The sequenced fragments are mapped to the genome. The enriched locations in the genome are found by detecting "peaks" of mapped fragments along the genome. These peaks should be significantly higher than the

			immunoprecipitation process and then the cross-linking is reversed The DNA/RNA fragments that are enriched in the protein-binding sites are then sequenced	mapped fragments in the surrounding loci, and significantly higher compared to a control sample—usually the input DNA of the ChIP experiments or another sample of immunoprecipitation done by a nonspecific antibody
	RNA-Seq	Detecting and comparing gene expression levels	Total RNA is extracted from the cells. In a sample preparation process the mRNA is pulled down and fragmented. The mRNA fragments are then reverse transcribed to cDNA. The cDNA fragments are sequenced	The cDNA fragments are mapped to the reference genome. The fragments that map to each gene are counted and normalized to allow comparisons between different genes and different samples Un-annotated genes and transcripts can be found in an RNA-Seq experiment by detecting bundles of fragments that are mapped to the genome in an un-annotated region
Reading/ counting	microRNA-Seq	Detect and count microRNAs	Total RNA is extracted from the cells, and the microRNA is isolated by recognizing the natural structure common to most known microRNA molecules. The microRNA fragments are then reverse transcribed and sequenced	The sequenced fragments are mapped to the genome. The microRNA can then be detected and counted

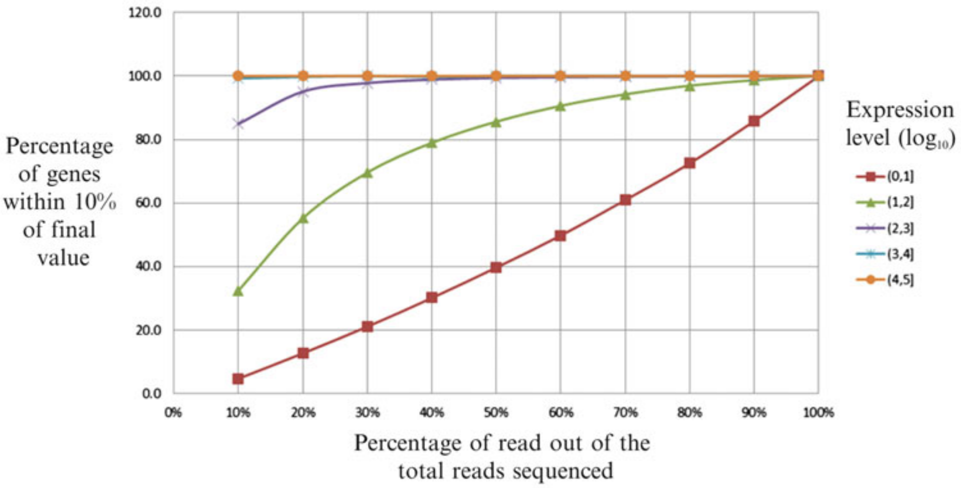


Fig. 2 Saturation report. The different series are sets of genes that differ in their final expression values using the complete dataset (in this case, 32 million reads). Highly expressed genes are saturated with even 10 % of the reads, whereas lowly expressed genes require a higher amount of reads, while very lowly expressed genes remain unsaturated even with the complete dataset

Contaminations may not pose a great difficulty for "reading" applications with a known reference genome, since they will not map to the reference genome. However contaminations "steal" coverage from the sample, and should be taken into account when estimating the expected coverage. If it is not possible to assess what percentage of contaminations the sample will contain, a pilot experiment may again prove useful: sequencing of just one or two samples in low coverage, and then assessing by mapping the percentage of contaminants. In de novo assembly, contaminations may be a lot more difficult to detect and thus attempts to eliminate contamination should be made when extracting the DNA, before sequencing and analysis.

In counting applications, such as RNA-Seq, the notion of coverage is not straightforward since the number of reads along the genome is not expected to be uniform. For example, most RNA-Seq reads will correspond to highly expressed transcripts, whereas lowly expressed transcripts will be less represented. This notion presents the question of how many reads are required for a particular application. In general, this is a trial-and-error process, and consequently we have found it useful to begin with a pilot experiment of a few samples to provide an estimate of the transcriptomic complexity.

An analysis that can help assess whether enough reads have been sequenced is a "saturation report" (Fig. 2, [2]). In this "jack-knifing" method, the expression levels are determined using all of the reads. The expression levels are then compared to those recalculated using only a fraction of the reads. Examining the expression levels at each cut of the data informs at which point the expression level remains unchanged despite additional data. As expected, additional data is most helpful in resolving the expression levels of the lowly expressed genes. After deciding how many reads are required per sample, the samples are divided into lanes according to the number of sequenced reads per lane, which is a fixed amount.

3.3 Sequencing Recipe: Single-Read vs. Paired-End, Insert Size, and Read Length

The sequencing recipe is influenced by several factors:

3.3.1 The Repetitive Nature of the Genome

Human and mouse genomes have ~20 % repetitive sequences [3]. Consequently, to uniquely score a read mapping to a repetitive region it must be longer than the repetitive region or border the neighboring non-repetitive sequence. Longer reads or PE reads allows "rescue" of the nonunique end and also mapping to nonunique regions in the genome (Fig. 3).

3.3.2 Differentially Spliced Variants

When assessing gene expression levels in RNA-Seq, it is potentially informative to discover the differential expression levels of different transcripts of the same gene. Reads that map to an exon shared by more than one transcript pose a difficulty in assessing the transcript of origin. PE reads may solve this problem if one end of the sequenced fragment maps to an exon that is unique to one of the transcripts. Figure 4 shows an example in which one cannot determine with certainty from which transcripts the SR originated. Sequencing it as PE resolves this problem.

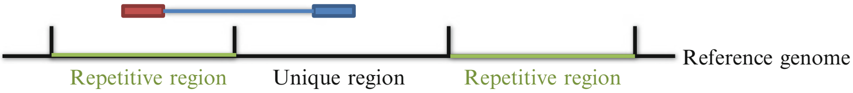


Fig. 3 The *red* end would not have been uniquely mapped if sequenced as a single read as opposed to a paired-end read

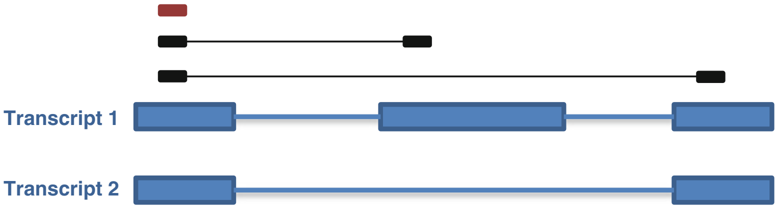


Fig. 4 The single read maps to the gene, however, cannot distinguish among the transcripts. Paired-end reads provide a better chance at identifying splice variants

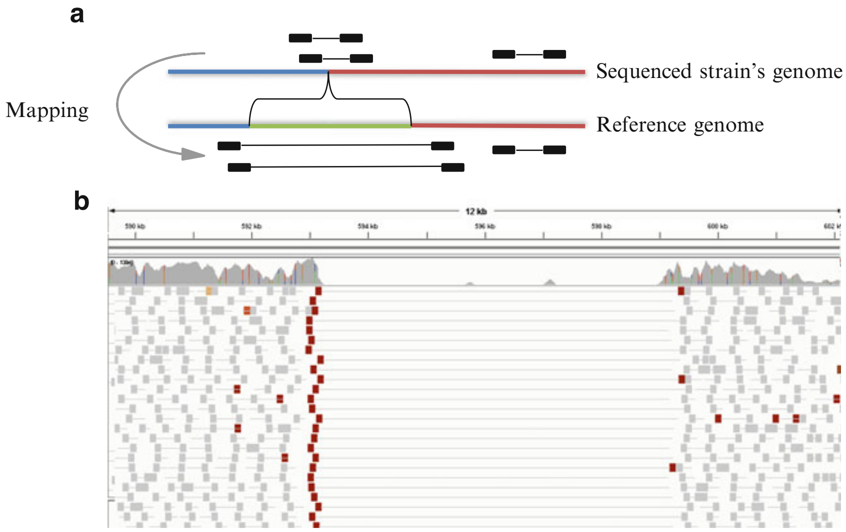


Fig. 5 (**a**) The sequenced strain contains a deletion in comparison with the reference genome. Consequently, paired-end reads mapped to the reference genome will have a bigger distance between them than the expected insert size. (**b**) An example of a genomic deletion in the IGV browser [4]

3.3.3 Genetic Distance of the Sequenced Sample from the Reference Genome

If the sequenced samples are genetically distant from the reference genome, longer reads may be required to determine the origin of each read in the genome. The mappings of each read will contain more mismatches, thus making it difficult to unambiguously determine its correct location, thereby increasing the probability that more than one location may be possible. Thus, the longer the read, the more likely a unique mapping becomes.

3.3.4 Finding Structural Variations

Structural variations in the genome, such as long insertions or deletions, inversions, and translocations, can be found using PE information. For example, if a large deletion is present in the sequenced strain, the insert lengths will be longer than expected (Fig. 5).

3.3.5 De Novo Assembly

Assembling a new genome from short sequenced reads consists of overcoming many challenges, such as sequencing errors, low-complexity regions, and repetitive regions among others [5, 6]. De novo assembly remains a notoriously difficult problem and often the genome of a metazoan remains in thousands of contigs. Obviously, longer PE reads lead to better assemblies. It has also been shown that using a few sequencing libraries with different insert length may improve the assembly process [5].

3.4 Number of Samples for Sequencing

3.4.1 Resequencing

If the reference genome to which the sequenced reads are mapped is genetically distant, sequencing the actual strain in its baseline state (before the mutagenesis, without the phenotypic change, etc.) will be beneficial for interpreting the data. This will help in distinguishing the variations that are due to evolutionary distance from those that cause the actual phenotypic trait under study.

3.4.2 RNA-Seq

It is highly recommended to sequence a few biological replicates to control for biological noise. Technical replicates will also inevitably show variation [7]. Some gene expression software programs, such as Cufflinks [8], can use the data from different replicates and merge it into one value with a higher statistical significance.

3.4.3 ChIP-Seq

A ChIP-Seq experiment should include the IP DNA and one more sample that will serve as a control. The control sample may be the input DNA, before the IP process, or an IP done on the same DNA with a nonspecific antibody, such as IgG [9, 10]. Sequencing a control sample enables detection of enriched regions that are also significantly enriched compared to the control sample, and not only enriched compared to the area surrounding them in the IP sample. This may reduce false-positive peaks detected solely because of areas in the genome that have a higher coverage due to better DNA fragmentation compared to the surrounding area.

3.5 Analysis Pipelines

Figure 6 shows the bioinformatics pipelines involved in four main applications: resequencing, de novo assembly, RNA-Seq, and ChIP-Seq. Several processes are common to all or multiple applications.

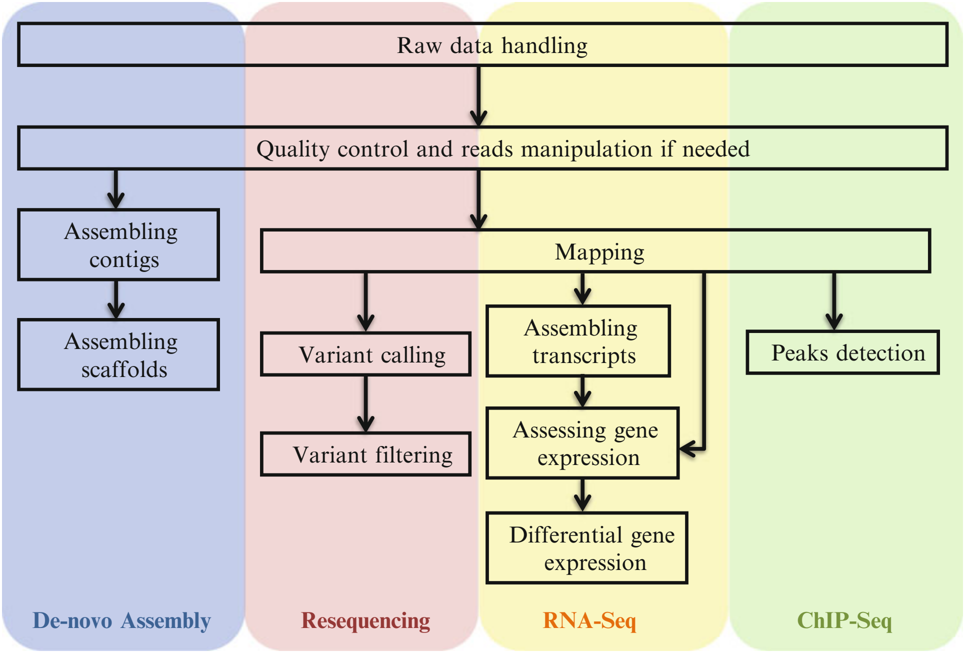


Fig. 6 Bioinformatics pipelines of the four main applications

3.5.1 Raw Data Handling

Available software for this step: Illumina's CASAVA software. The Illumina run produces "*b*ase-*cal*ling" files (*.bcl) which only become useful bioinformatically when converted to the general fastq format (see below). During this file conversion, the demultiplexing process is also carried out, which is the separation of reads from different samples that were sequenced on the same lane.

3.5.2 Quality Control and Read Manipulation

Available software for this step: CASAVA and FastQC (Babraham Bioinformatics). After a sequencing run is completed and before starting the analysis, the run's quality should be checked for the following parameters which may be telling of the quality of the sample and run.

1. Pass filter (PF) reads—The number and percentage of PF reads in each lane and for each sample should match the number of expected sequenced reads. If it is dramatically lower, this might indicate a low-quality run, and may reduce the expected coverage.
2. Control reads—Apart from the DNA libraries, control DNA from the viral PhiX genome is spiked-in at 1 % concentration with the sample onto each lane of the flowcell. Reads are automatically mapped by the Illumina software to the PhiX genome. The percentage of reads from each lane mapping to this genome and the amount of mismatches in the mapping are used as control values for the lane's quality. A good run typically has ~1 % sequencing errors, as detected by the mismatches to the PhiX genome.
3. Quality scores of the reads—As will also be explained in the next section ("Diving into the technical details") each base of each sequenced read is associated with a quality score providing the confidence in the particular base. In general, the quality scores drop toward the end of the sequenced read. These confidences should be assessed to check for the overall quality of the run. The quality scores may be automatically produced by the sequencing platform, and may also be created by programs like FastQC that provide other statistics on the sequenced reads, such as overrepresented sequences, per base GC content and more (Fig. 7).

Based upon these parameters, we found it advantageous in particular instances to further manipulate the sequences. For example, sequences may be trimmed to reduce low-quality ends, filtering reads by quality, and removal of adaptors.

3.5.3 Assembling Contigs and Scaffolds for De Novo Assembly

Available software for this step: SOAPdenovo [11], ABySS [12], Velvet [13], and ALL-PATHS [14]. De novo assembly is the most challenging application and continues to be the subject of intense algorithmic research. The process generally consists of three basic steps (Fig. 8):

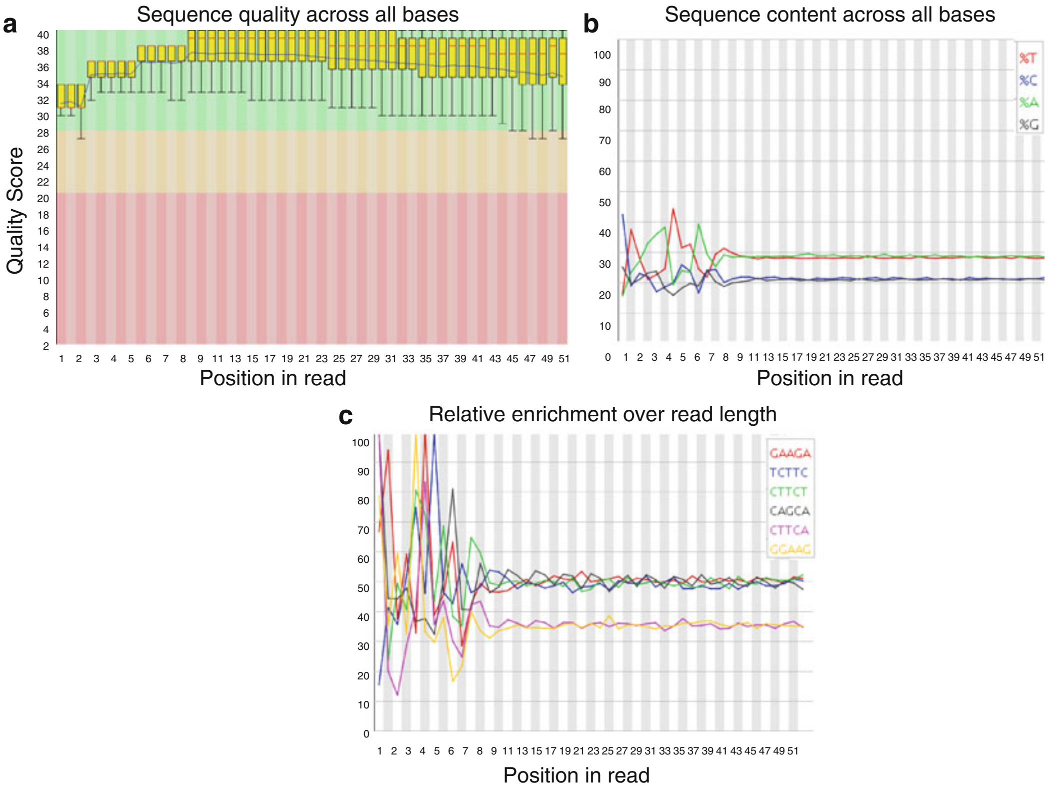


Fig. 7 Examples of statistics created by FastQC. (**a**) Quality score statistics per base. (**b**) Sequence contents per base. (**c**) Abundant Kmers across the reads

1. Contig-ing—The first step in the assembly consists of detecting overlaps between single reads. Bundle of overlapping reads are merged into a consensus sequence, called a contig. Repetitive or low-complexity regions in the assembled genomic sequence often prevent the construction of one long sequence at this initial step. This step typically results in >10,000 contigs, depending of course on the size of the genome and the number and length of sequenced reads.
2. Scaffolding—For de novo sequencing of complex genomes, it is crucial for the sequenced reads to be of paired-ends inserts. If so, the many contigs can then be merged onto longer segments called scaffolds by taking into account the paired-end information of the reads. Since the paired-end inserts contain an unknown sequence between the two reads, the scaffold may contain unknown sequence (represented as N's) of a size that can be determined by the average insert length.
3. Gap closing—After creating the scaffolds, the sequence of any remaining gaps within the scaffolds may be resolved by mapping the original paired-end reads to the scaffolds and

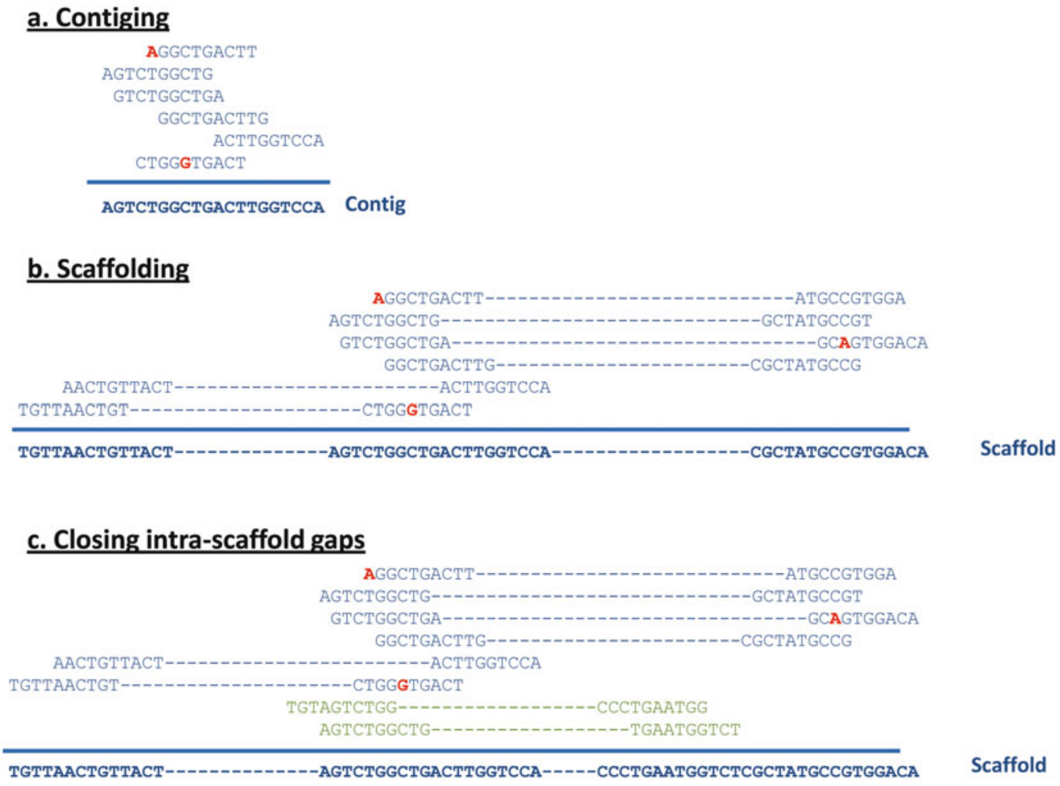

a. Contiging
AGGCTGACTT
AGTCTGGCTG
GTCTGGCTGA
GGCTGACTTG
ACTTGGTCCA
CTGGGTGACT
AGTCTGGCTGACTTGGTCCA
Contig
b. Scaffolding
AGGCTGACTT----------------------------ATGCCGTGGA
AGTCTGGCTG-----------------------------GCTATGCCGT
GTCTGGCTGA---------------------------------GCAGTGGACA
GGCTGACTTG-----------------------CGCTATGCCG
AACTGTTACT------------------------ACTTGGTCCA
TGTTAACTGT---------------------CTGGGTGACT
TGTTAACTGTTACT--------------AGTCTGGCTGACTTGGTCCA------------------CGCTATGCCGTGGACA
Scaffold
c. Closing intra-scaffold gaps
AGGCTGACTT----------------------------ATGCCGTGGA
AGTCTGGCTG-----------------------------GCTATGCCGT
GTCTGGCTGA---------------------------------GCAGTGGACA
GGCTGACTTG-----------------------CGCTATGCCG
AACTGTTACT------------------------ACTTGGTCCA
TGTTAACTGT---------------------CTGGGTGACT
TGTAGTCTGG------------------CCCTGAATGG
AGTCTGGCTG------------------TGAATGGTCT
TGTTAACTGTTACT--------------AGTCTGGCTGACTTGGTCCA-----CCCTGAATGGTCTCGCTATGCCGTGGACA
Scaffold

Fig. 8 Three basic steps of de novo assembly: (**a**) Aligning reads to find overlaps. (**b**) Connecting contigs into scaffold by using PE information. (**c**) Closing intra-scaffold gaps

searching for a read that informs the gap regions. This function may be an integrated process of some assemblers or a separate function may need to be run as in SOAPdenovo.

It should be noted that de novo assembly projects may include a reference genome of a close strain, or sequences that are known to be included in the assembly, which may help with the assembly process. In this section we discuss the basic de novo assembly process that does not rely on additional reference sequences.

In the assembly process the identification of sequencing errors is more difficult than it is when mapping reads to a reference genome. Detection of sequencing errors in the process of finding overlaps and merging them into a consensus sequence is possible if there is enough coverage. This is one of the reasons that a higher coverage is required for de novo assembly compared to application that consists of a known reference genome.

3.5.4 Mapping

Available software for this step: BWA [15], Bowtie [16], and TopHat [8]. The process of mapping is done in any application that includes a known reference genome. Each read is mapped to the reference genome separately under the conditions of the mapping software, as defined by the input parameters. PE reads are each mapped separately and only then the distance between their mappings is measured.

The main parameters inputted for a mapping software deal with the measure of difference between the read and the reference genome. As in many other bioinformatics methods, deciding on the measure of similarity between reads and the reference genome raises the dilemma between sensitivity and specificity: Allowing too much difference may result in false-positive mappings, while allowing too little difference may lead to missing true positives. From our experience the best way to decide on the parameters is to try a few values and see how they affect the results.

There are two main methods to control the measure of dissimilarity between reads and the reference genome:

1. Number of differences per read—Apply the mapping software with a value that defines the maximum number of allowed differences (mismatches, insertions, and deletions) between the read and the reference genome.
2. Seed mapping—In this method the software looks for a sequence of certain length inside the read that does not contain differences or contains a small amount of differences compared to the reference genome. The rest of the alignment is elongated without limiting the amount of differences. The parameters given to the software control the seed length, the amount of differences allowed in it, and sometimes also the intervals in the reads in which it is searched.

In general, seed mapping is a more permissive approach and is suitable for sequence strains that are distant from the reference genomes they are mapped to. The first method is more strict and is suitable for strains that are known to be close to the reference genome and when trying to avoid false positives. It should be noted that when using the first method and allowing many differences per read, the results become similar to those that are received in the second method. The sensitivity and specificity can be tuned also by the parameters of each method.

It is important to remember that the way the mapping step is done affects the rest of the analysis. Allowing a low amount of mismatches may cause regions in the reference genome that contain many variations compared to the sequenced strain to have little to no coverage. Regions in the genome with little to no coverage may be caused by a few reasons. First, region is not present in the sequenced strain—the zero coverage implies a deletion compared to the reference genome. Second, the region does exist in the sequenced strain but is not represented in the sequenced library because of a bias caused in the sample preparation process (for example, because of some regions in the genome that are not sonicated as well as others). Finally, the low coverage may also be caused by allowing too few differences per read to a region in the genome that contains many variations in the sequenced strain compared to the reference genome. Trying to map the reads again with a higher percentage of differences may cause these low-coverage regions to "fill-up."

After the mapping is done one can choose to use only a partial set of the mappings:

1. Use only uniquely mapped reads: It is very common for initial analyses to use only reads that map to one unique location in the genome. Under the mapping conditions, defined by the parameters, reads may be mapped to more than one location in the genome. In this case, one cannot surely determine where the read has originated from. There are a few approaches to deal with such reads—map them randomly to one of the possible locations, map them to all locations, apply an even amount of coverage to every possible location, etc. Each of these approaches may cause a bias in the results, and can be ignored in the initial analysis by using only the uniquely mapped reads.
2. Use mappings with a minimum mapping score: One can choose to use only mappings of higher quality in order to disregard low-quality mappings that may introduce false positives.
3. Filter mappings with certain insert sizes: PE reads are first mapped separately and only then is the distance between them measured. Long insert sizes or reads that map to different chromosomes may imply structural variations such as large deletions, inversions, and translocations (Fig. 5). One can choose to use only mappings with irregular insert size to find such structural variations or use only mappings with normal insert size for initial variant analysis. BreakDancer is an example of a program that uses PE information to find structural variations [17].
4. Removal of PCR duplicates: PCR amplification is part of the sample preparation, and may introduce bias. PCR duplicates may be identified as reads that map to the exact same location, and in PE reads have the same insert size.

3.5.5 Variant Calling and Filtering

Available software for this step: SAMtools [18], GATK [19], and MAQ [20]. Based on the mapping done in the previous step, variants can be called by finding the consensus sequence from the mapped reads. The first step in this process is to create a "pileup" file of the mapped reads. This file summarizes for each base in the reference genome the coverage of the reads that are mapped at the loci and the called bases of these reads. Depending on the software that creates the pileup file, more information can be obtained from it, such as genotype calling, mapping qualities, and *p*-values. The information in the pileup file can be used to detect and filter variants. The two basic parameters that help detect variants are the following:

(a) Coverage at the loci—The detected variants should rely on a sufficient coverage. A minimum number of reads should be set as a threshold for initial filtering.

(b) Frequency of the allele that was sequenced—The variant should have sufficient frequency out of the total reads covering the loci. If one read out of 15 reads covering a loci shows a base different from the reference genome, it may not imply a variant but rather a sequencing error. To find heterozygous variants the frequency should be ~0.5, and for homozygous variants the frequency should be ~1; if pooling was done then the frequency should match the expected percentage in the pooling. When filtering by allele frequency taking a margin of security is recommended, especially if the coverage is low. For example, for heterozygous variants filter by a frequency of 0.4 or 0.3.

The above are two basic parameters for variant filtering, but other parameters can be used for variant filtering, for example, the mapping and base qualities in the variant locations.

3.5.6 Assembling Transcripts

In strains that do not have full or sufficient gene annotations, novel annotations can be found by HTS. The idea is to sequence mRNA, map the reads to the reference genome, and infer transcripts from the detected bundles of reads in a certain loci. Based on these annotations a gene expression analysis can then be done. In principle, one can assemble the whole genome before performing RNA-Seq experiment, or assemble the transcriptome only in an application called “de novo RNA” [21] (or a combination of both).

3.5.7 Gene Expression Analysis

Available software for this step: Cufflinks [8] and Myrna [22]. After mapping the reads to the reference genome an assessment of their abundance can be made by the gene annotations. In general, the amount of reads that overlap each gene is counted. The raw count must be normalized for further analysis. A common normalization method is called Fragments per Kilobase Million (FPKM) and is calculated as follows:

$$\text{FPKM} = \frac{\text{raw count}}{\text{gene length} \times \text{number of mapped reads in millions}}$$

The normalization takes into account the gene’s length, to avoid a bias toward higher expression in longer genes. FPKM also takes into account the total number of mapped reads in each sample, to avoid a bias because of difference in number of reads in each sample.

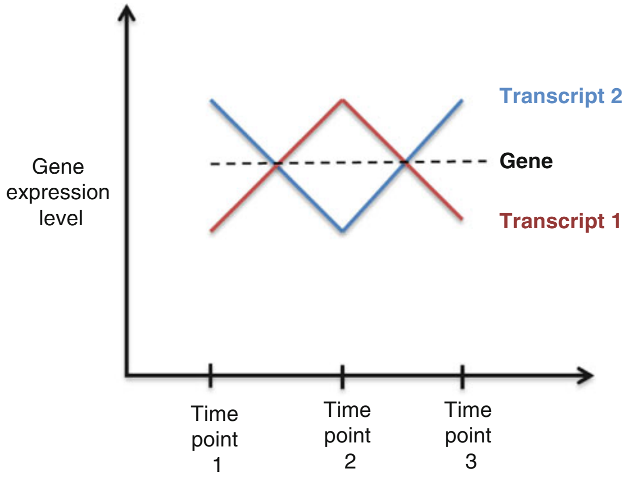


Fig. 9 An example of differential expression in the transcript abundances of a gene. If only the gene's expression level is calculated the expression does not appear to change over time; yet a separate calculation of each transcript's expression level shows a different molecular event

A basic approach to gene expression is to count all the reads that map to a gene's annotation, normalize them, and set this value as the expression level of that gene. If a gene has more than one transcript due to alternative splicing, not separating the reads that map to it to each of the transcript can cause a great bias and change the results entirely (Fig. 9). Finding the expression levels of different transcripts of the same gene is challenging, since reads that map to exons that belong to more than one transcript cannot be unambiguously correlated to one transcript [8]. The software Cufflinks [8] attempts to assess transcript expression levels by using the reads that can be unambiguously correlated with certain exons to infer the expression of all the reads (Fig. 10). Cufflinks' algorithm uses maximum likelihood to assess the abundances of each transcript.

3.5.8 Peak Detection

Available software for this step: MACS [23, 24] and SICER [25]. A ChIP-Seq experiment is done to detect enriched regions in the IP sample. These regions, called "peaks" or "islands," should be significantly higher both from their surrounding in the IP sample and from the same loci in the control sample. The peaks are found by statistical modeling of the enriched regions compared to the control. There are two important parameters for peak detection: the abundance in the genome and the width of the binding sites. We introduce two programs for peak detection, each addressing binding sites with different abundance and width characteristics. MACS is more suitable for narrow peaks that represent short and specific binding sites, for example, of transcription factors. SICER is more suitable for wide peaks that extend over thousands of base pairs;

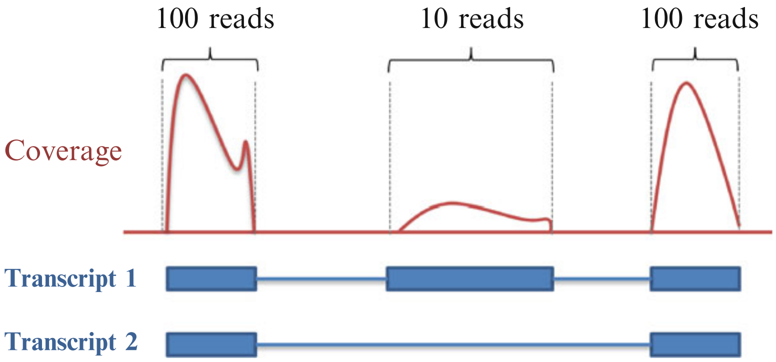


Fig. 10 Assessing transcript abundance. Since 10 reads undoubtedly originated from transcript 1, it may be inferred that 90 reads from each shared exon originated from transcript 2 while 10 reads from each shared exon originated from transcript 1

these peaks are typical for histone modification experiments, in which many close binding across the genome. Their proximity to each other makes the peaks merge to wide enriched regions rather than short and sharp peaks.

4 Notes

4.1 Tuning Up the Pipelines

The pipelines detailed above are general. It is crucial to examine each project specifically and decide what pipeline is best suited for it. Tuning up the parameters of each step in the pipeline may be vital for accurate results. Tuning up the pipeline and parameters can be done by following the general pipeline presented above and conducting quality control measurements after each step, to allow identifying a phenomenon that might infer some insight or require special action in the analysis.

Quality control measurements should be done after each step in the analysis. After the raw data handling a quality control step is done as detailed above. After the mapping step the mapping statistics should be assessed. How many reads were not mapped, uniquely mapped, and multi-mapped? High values in the first and third parameters may infer a problem. How does the coverage profile look like? What percentage of the genome is covered with sufficient coverage, and what is the average coverage? For exome projects, what is the coverage over exons? It is highly recommended to look at the mappings in a genome viewer. Some phenomena can be detected easier visually (Fig. 11).

Tuning up the parameters in each step of the analysis allows to control the balance between sensitivity and specificity. For example, if we allow one mismatch per 50 bp read in the mapping step, it will reduce the rate of incorrect mappings, but we will not be able to

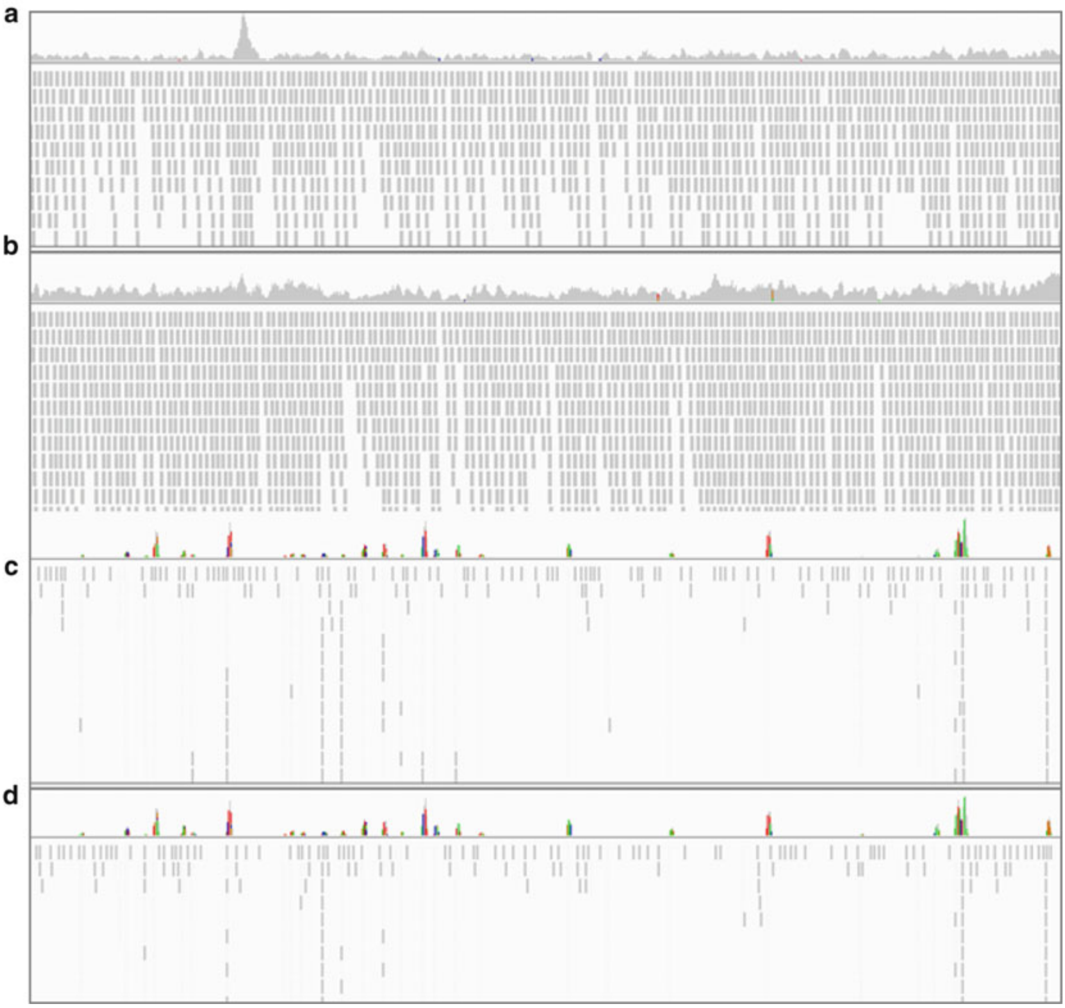


Fig. 11 Four bacteria samples were sequenced and mapped to the same reference genome. The mapping statistics of all of them showed that 96–98 % of the reads were unmapped. Viewing them on a genome viewer reveals the different phenomenon in each sample. (**a**), (**b**) Only 2–3 % of the reads are of the expected strain, while the rest are contamination. This can be seen by the high and continuous coverage and lack of variants. (**c**), (**d**) These sequenced samples seem to be evolutionarily distant from the reference genome, as can be seen by the low and segmented coverage and many variants

detect 2-base indel or areas in the genome that have more than one variant per 50 bp; the coverage in these regions will be low or zero due to incapability of mapping. Another example from gene expression analysis: when comparing gene expression between two samples one can choose to statistically test only genes that have a minimum amount of reads mapped to them in at least one sample [8]. Choosing a high threshold may cause missing interesting genes, but choosing a low threshold may include genes, the differential expression of which is not significant—a gene can be expressed in a fold change of 5 if the ration between the samples is 1 read vs. 5 reads or 1,000 reads vs. 5,000 reads.

4.2 Diving into the Technical Details: File Formats

In this section we overview the formats of some basic files used in HTS data analysis. Though not all useable formats are mentioned here, this section provides a general idea of how the files used in the analysis are constructed, as their structure is similar and the same concepts generally apply. All the files we present in this section and most of the files used for HTS analysis are plain text files and usually tab delimited, which enables easy management by various tools and scripts.

4.2.1 Fastq: Raw Read Format (Fasta + Quality)

A fastq file is constructed out of quadruplet lines (Fig. 12), each quadruplet representing a read and containing the following information:

1. Read identifier—PE reads will have the same identifier. The read's identifier is unique and is constructed in the following way (CASAVA 1.8.2): @<instrument>:<run number>:<flowcell ID>:<lane>:<tile>:<x-pos on tile>:<y-pos on tile>

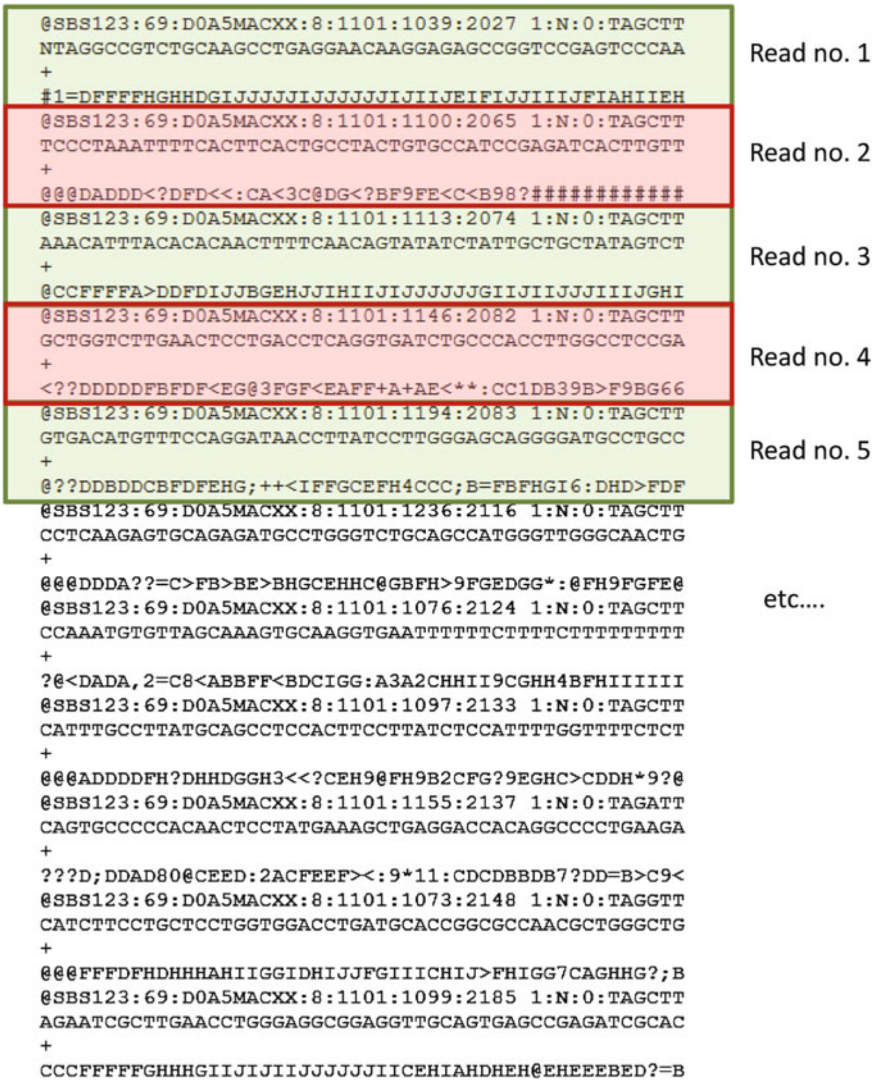
   <read (1/2)>:<is filtered>:<control number>:<index sequence>
2. Sequence.
3. Read description (optional).
4. Quality score per base. Each base is associated with a quality score that defines how reliable the base is. The score is called Phred quality score and defined as $P(\text{base is miscalled}) = \frac{1}{10^{Q/10}}$ where Q is the quality value. The quality values are typically between 2 and 50. For example, the quality scores 20, 30, and 40 refer to an error probability of 1/100, 1/1,000, and 1/10,000, respectively. In order to encode each quality score into one character in the fastq file the following procedure is done: A value is added to the quality score, either 33 or 64, and the new value is then encoded into a character using the ASCII table (Fig. 13). The number that is usually being added is 33, while some old CASAVA versions used to add 64 instead.

4.2.2 SAM: Sequence Alignment/Mapping Format

Each line in a SAM tab-delimited file contains information on a single read and its mapping as it was done on using a mapping software. A SAM file is well defined in the SAM specification document [18]; periodically this is updated. The following is a general overview of the SAM file structure (applicable for SAM specification v1.4-r985).

Figure 14 shows an example of a SAM file. It is constructed of the following:

- *Header lines*: Appear in the beginning of the file; they begin with a "@" character and define general information regarding the mapping. Among the possible header lines are @SQ that



```
@SBS123:69:D0A5MACXX:8:1101:1039:2027 1:N:0:TAGCTT
NTAGGCCGTCTGCAAGCCTGAGGAACAAGGAGAGCCGGTCCGAGTCCCAA
+
#1=DFFFFHGHHDGIJJJJJIJJJJJJIJIIJEIFIJJIIIJFIAHIIEH
@SBS123:69:D0A5MACXX:8:1101:1100:2065 1:N:0:TAGCTT
TCCCTAAATTTTCACTTCACTGCCTACTGTGCCATCCGAGATCACTTGTT
+
@@@DADDD<?DFD<<:CA<3C@DG<?BF9FE<C<B98?############
@SBS123:69:D0A5MACXX:8:1101:1113:2074 1:N:0:TAGCTT
AAACATTTACACACAACTTTTCAACAGTATATCTATTGCTGCTATAGTCT
+
@CCFFFFA>DDFDIJJBGEHJJIHIIJIJJJJJGIIJIIJJJIIIJGHI
@SBS123:69:D0A5MACXX:8:1101:1146:2082 1:N:0:TAGCTT
GCTGGTCTTGAACTCCTGACCTCAGGTGATCTGCCCACCTTGGCCTCCGA
+
<??DDDDDFBFDF<EG@3FGF<EAFF+A+AE<**:CC1DB39B>F9BG66
@SBS123:69:D0A5MACXX:8:1101:1194:2083 1:N:0:TAGCTT
GTGACATGTTTCCAGGATAACCTTATCCTTGGGAGCAGGGGATGCCTGCC
+
@??DDBDDCBFDFEHG;++<IFFGCEFH4CCC;B=FBFHGI6:DHD>FDF
@SBS123:69:D0A5MACXX:8:1101:1236:2116 1:N:0:TAGCTT
CCTCAAGAGTGCAGAGATGCCTGGGTCTGCAGCCATGGGTTGGGCAACTG
+
@@@DDDA??=C>FB>BE>BHGCEHHC@GBFH>9FGEDGG*:@FH9FGFE@
@SBS123:69:D0A5MACXX:8:1101:1076:2124 1:N:0:TAGCTT
CCAAATGTGTTAGCAAAGTGCAAGGTGAATTTTTTCTTTTCTTTTTTTTT
+
?@<DADA,2=C8<ABBFF<BDCIGG:A3A2CHHII9CGHH4BFHIIIIII
@SBS123:69:D0A5MACXX:8:1101:1097:2133 1:N:0:TAGCTT
CATTTGCCTTATGCAGCCTCCACTTCCTTATCTCCATTTTGGTTTTCTCT
+
@@@ADDDDFH?DHHDGGH3<<?CEH9@FH9B2CFG?9EGHC>CDDH*9?@
@SBS123:69:D0A5MACXX:8:1101:1155:2137 1:N:0:TAGATT
CAGTGCCCCCACAACTCCTATGAAAGCTGAGGACCACAGGCCCCTGAAGA
+
???D;DDAD80@CEED:2ACFEEF><:9*11:CDCDBBDB7?DD=B>C9<
@SBS123:69:D0A5MACXX:8:1101:1073:2148 1:N:0:TAGGTT
CATCTTCCTGCTCCTGGTGGACCTGATGCACCGGCGCCAACGCTGGGCTG
+
@@@FFFDFHDHHHAHIIGGIDHIJJFGIIICHIJ>FHIGG7CAGHHG?;B
@SBS123:69:D0A5MACXX:8:1101:1099:2185 1:N:0:TAGCTT
AGAATCGCTTGAACCTGGGAGGCGGAGGTTGCAGTGAGCCGAGATCGCAC
+
CCCFFFFFGHHHGIIJIJIIJJJJJJIICEHIAHDHEH@EHEEEBED?=B
```

Fig. 12 The anatomy of a fastq file

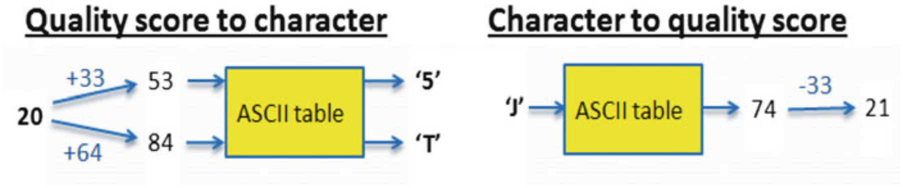
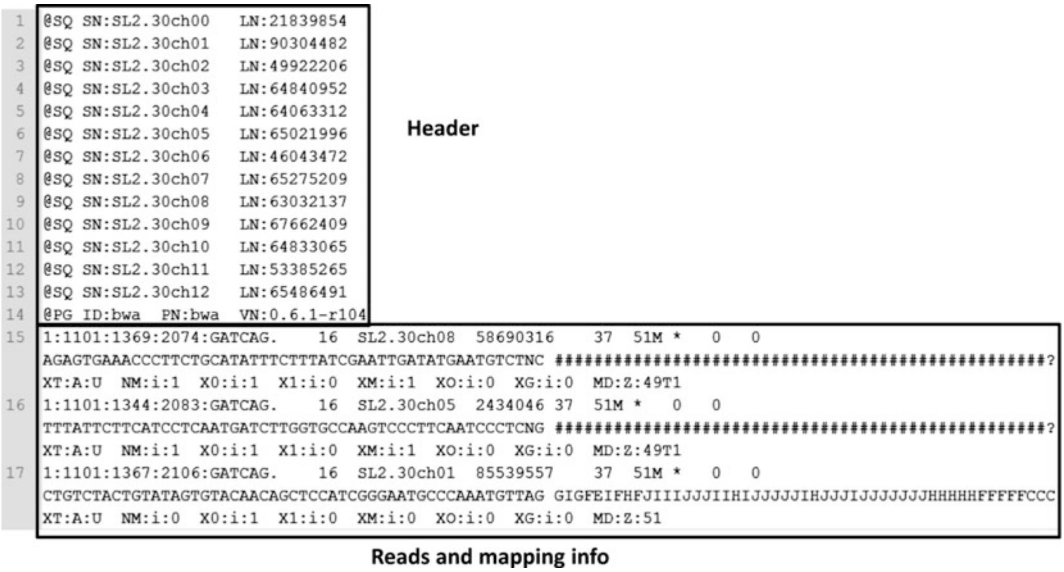


Fig. 13 The encoding and decoding of a quality score

details the reference genome's chromosomes, and @PG which contains details about the mapping program.

- *Read lines*: One line per read, containing information about the read and its mapping.



```
@SQ SN:SL2.30ch00   LN:21839854
@SQ SN:SL2.30ch01   LN:90304482
@SQ SN:SL2.30ch02   LN:49922206
@SQ SN:SL2.30ch03   LN:64840952
@SQ SN:SL2.30ch04   LN:64063312
@SQ SN:SL2.30ch05   LN:65021996
@SQ SN:SL2.30ch06   LN:46043472
@SQ SN:SL2.30ch07   LN:65275209
@SQ SN:SL2.30ch08   LN:63032137
@SQ SN:SL2.30ch09   LN:67662409
@SQ SN:SL2.30ch10   LN:64833065
@SQ SN:SL2.30ch11   LN:53385265
@SQ SN:SL2.30ch12   LN:65486491
@PG ID:bwa  PN:bwa  VN:0.6.1-r104
1:1101:1369:2074:GATCAG.    16  SL2.30ch08  58690316    37  51M *   0   0
AGAGTGAAACCCTTCTGCATATTTCTTTATCGAATTGATATGAATGTCTNC ##################################################?
XT:A:U  NM:i:1  X0:i:1  X1:i:0  XM:i:1  XO:i:0  XG:i:0  MD:Z:49T1
1:1101:1344:2083:GATCAG.    16  SL2.30ch05  2434046 37  51M *   0   0
TTTATTCTTCATCCTCAATGATCTTGGTGCCAAGTCCCTTCAATCCCTCNG ##################################################?
XT:A:U  NM:i:1  X0:i:1  X1:i:0  XM:i:1  XO:i:0  XG:i:0  MD:Z:49T1
1:1101:1367:2106:GATCAG.    16  SL2.30ch01  85539557    37  51M *   0   0
CTGTCTACTGTATAGTGTACAACAGCTCCATCGGGAATGCCCAAATGTTAG GIGFEIFHFJIIIJJJIIHIJJJJJIHJJJIJJJJJJHHHHHFFFFFCCC
XT:A:U  NM:i:0  X0:i:1  X1:i:0  XM:i:0  XO:i:0  XG:i:0  MD:Z:51
```

Fig. 14 The anatomy of a SAM file

The reads' lines are constructed with the attributes shown in Table 2.

The flag is one number that contains answers to the following 11 YES/NO questions regarding the read's mapping:

1. Is the read paired?
2. Is the read mapped in proper pair (in the expected insert length)?
3. Is the read unmapped?
4. Is the mate unmapped?
5. Is the read mapped to the reverse strand?
6. Is the mate mapped to the reverse strand?
7. Is the read the first in the pair?
8. Is the read the second in the pair?
9. Is the mapping not a primary alignment?
10. Did the read fail platform/vendor quality checks?
11. Is the read a PCR or an optical duplicate?

To encode the answers to these questions to one number, a NO answer is encoded as "0," and a YES answer is encoded as "1". The binary number resulting from the series of answers is then converted to a decimal number (*see* Fig. 15 for an example).

CIGAR: Compact Idiosyncratic Gapped Alignment Report

This details the mapping structure between the read and reference genome, according to a specific encoding. As an example, "M" corresponds to a matched alignment. Thus, 101 M means that

Table 2
Description of the columns of a SAM format file

Column	Field	Description
1	Read ID	The read's identifier as it appears in the fastq file
2	Flag	See text
3	Chromosome	The chromosome to which the read was mapped to. "*" if the read is not mapped
4	Position	The position on the chromosome to which the read was mapped to. "*" if the read is not mapped
5	Mapping quality	The mapping quality score that was specified by the mapping software. "255" if the mapping quality is not available
6	CIGAR	See text
7	Mate chromosome	The chromosome to which the read's mate was mapped to. "*" if the mate is not mapped. "=" if it is identical to the chromosome of the read
8	Mate position	The position on the chromosome to which the read's mate was mapped to. "*" if the mate is not mapped
9	Insert length	The distance between the mappings of the two reads, inferring the insert size
10	Sequence	The read's sequence, as it appears in the fastq file
11	Quality scores	The read's quality scores, as it appears in the fastq file
12 (optional)	Program-specific attributes	See text

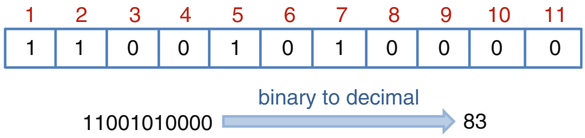
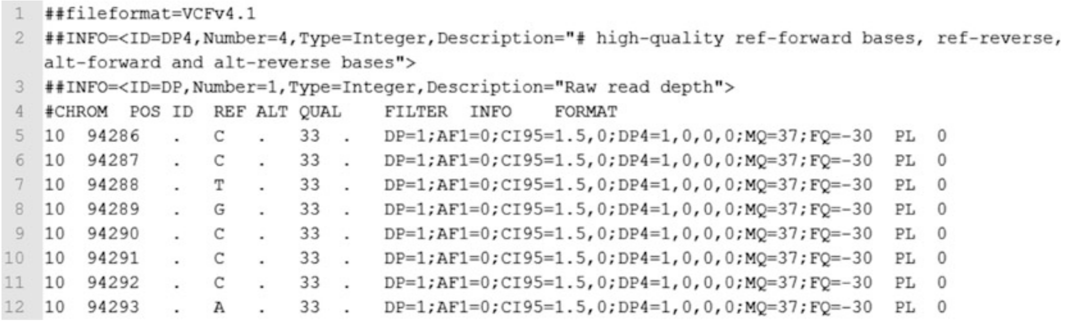


Fig. 15 The flag in the SAM file is binary encoded with the following 11 bits of information: read is paired, read is mapped in a proper pair, read is on reverse strand, read is the first in the pair

there were 101 matches or mismatches without gaps opened, and 73M1I27M means that the first 73 bases were a match or a mismatch compared to the reference genome, then there was one base insertion, and then another 27 matches/mismatches.

The optional attributes detail more information about the mapping. Some of the options are the edit distance in the mapping, mismatch positions, and number of gap openings. These attributes are tab delimited and will be of the form<Tag>:<Type>:<Details>



```
##fileformat=VCFv4.1
##INFO=<ID=DP4,Number=4,Type=Integer,Description="# high-quality ref-forward bases, ref-reverse,
alt-forward and alt-reverse bases">
##INFO=<ID=DP,Number=1,Type=Integer,Description="Raw read depth">
#CHROM  POS  ID  REF ALT QUAL    FILTER  INFO    FORMAT
10  94286   .   C   .   33  .   DP=1;AF1=0;CI95=1.5,0;DP4=1,0,0,0;MQ=37;FQ=-30  PL  0
10  94287   .   C   .   33  .   DP=1;AF1=0;CI95=1.5,0;DP4=1,0,0,0;MQ=37;FQ=-30  PL  0
10  94288   .   T   .   33  .   DP=1;AF1=0;CI95=1.5,0;DP4=1,0,0,0;MQ=37;FQ=-30  PL  0
10  94289   .   G   .   33  .   DP=1;AF1=0;CI95=1.5,0;DP4=1,0,0,0;MQ=37;FQ=-30  PL  0
10  94290   .   C   .   33  .   DP=1;AF1=0;CI95=1.5,0;DP4=1,0,0,0;MQ=37;FQ=-30  PL  0
10  94291   .   C   .   33  .   DP=1;AF1=0;CI95=1.5,0;DP4=1,0,0,0;MQ=37;FQ=-30  PL  0
10  94292   .   C   .   33  .   DP=1;AF1=0;CI95=1.5,0;DP4=1,0,0,0;MQ=37;FQ=-30  PL  0
10  94293   .   A   .   33  .   DP=1;AF1=0;CI95=1.5,0;DP4=1,0,0,0;MQ=37;FQ=-30  PL  0
```

Fig. 16 The anatomy of a VCF file

Tag—identifies what kind of information is detailed, according to the SAM specification.
Type—I for integer, Z for string.
Details—the details themselves.

For example, NM:i:3 means an edit distance of 3 and MD: Z:74G26 means that there is a mismatch in the 75th position of the reads, a "G" instead of the reference base.

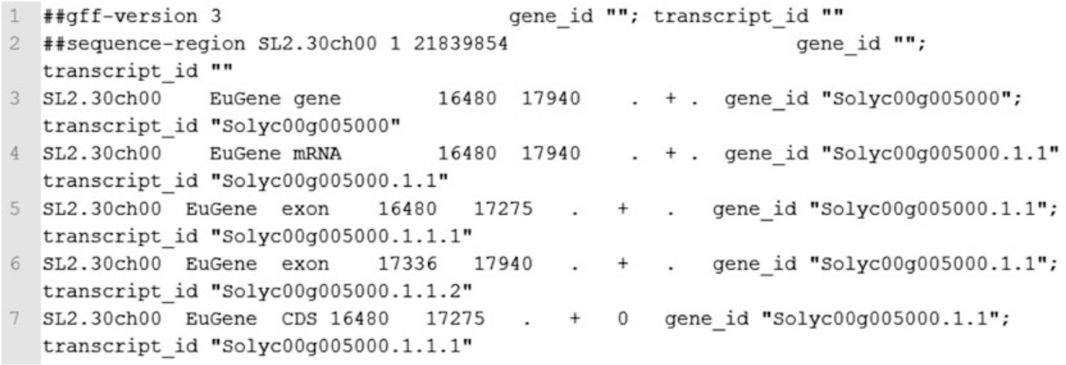
The SAM specification defines some of these options, and reserves attribute for program-specific needs; the reserved options start with an "X". These attributes should be defined in the mapper's documentations. SAMtools [18] is a program that enables manipulation, conversion, and data retrieval from SAM files.

4.2.3 VCF: Variant Call Format

A VCF file details information per base of the reference genome, accumulated from the mappings in a SAM file (Fig. 16). A VCF file is a tab-delimited file, also constructed from header lines and variant lines. The header lines begin with "#" character and detail general information on the file, such as the program used for the variant calling process, and the attributes that appear in each variant line. Each line in the rest of the document contains information about a specific base in the genome. Only bases that have a coverage of at least one appear in a VCF file. A "raw" VCF file contains information about every base with coverage in the genome. It can then be filtered to contain bases that define a variant compared to the reference genome. VCFtools [26] is a program package designed to working with VCF files.

4.2.4 GFF: General Feature Format

A GFF file contains details about annotations of a specific genome sequence (Fig. 17). A GFF file should be of the same build or version of the genome sequence it refers to. A GFF file is constructed of header lines that begin with a "#" character and feature lines. The feature lines are tab delimited and contain the attributes shown in Table 3.



```
##gff-version 3                        gene_id ""; transcript_id ""
##sequence-region SL2.30ch00 1 21839854                        gene_id "";
transcript_id ""
SL2.30ch00    EuGene gene       16480  17940    . + .  gene_id "Solyc00g005000";
transcript_id "Solyc00g005000"
SL2.30ch00    EuGene mRNA       16480  17940    . + .  gene_id "Solyc00g005000.1.1"
transcript_id "Solyc00g005000.1.1"
SL2.30ch00  EuGene  exon    16480   17275   .   +   .  gene_id "Solyc00g005000.1.1";
transcript_id "Solyc00g005000.1.1.1"
SL2.30ch00  EuGene  exon    17336   17940   .   +   .  gene_id "Solyc00g005000.1.1";
transcript_id "Solyc00g005000.1.1.2"
SL2.30ch00  EuGene  CDS 16480   17275   .   +   0   gene_id "Solyc00g005000.1.1";
transcript_id "Solyc00g005000.1.1.1"
```

Fig. 17 The anatomy of a GFF file

Table 3
Description of the columns of a GFF format file

Column	Field	Description
1	Chromosome	The chromosome on which the feature is located
2	Source	The source of this feature, usually the prediction software or a public DB
3	Feature	The feature type
4	Start	The start position on the chromosome on which the feature is located
5	End	The end position on the chromosome on which the feature is located
6	Score	A floating point value
7	Strand	The strand the feature is originated from ("+", "-", or "." if the strand is not relevant)
8	Frame	The position of the feature in the ORF ("0", "1", "2", or ".")
9	Attributes	More details about the feature, separated by ";". For example gene ID, gene description, exon number, description

4.2.5 File Formats: Summary

The file formats detailed above, similarly to other files used in HTS data analysis, enable easy retrieval of information using simple scripts or public programs. Knowing how the data is stored and where enables to ask questions such as the following:

- Which sequences were not mapped (look for lines with bit 4 in the flag equals to 0 in a SAM file)?
- What is the average coverage in a certain region in the genome (calculate the average of the DP values in a region in a VCF file)?
- What kind of annotations are known in a reference genome (find the possible options in column 3 in a GFF file)?

References

1. Goh Y, Fullwood MJ, Poh HM, Peh SQ, Ong CT, Zhang J, Ruan X, Ruan Y (2012) Chromatin interaction analysis with paired-end tag sequencing (ChIA-PET) for mapping chromatin interactions and understanding transcription regulation. J Vis Exp e3770
2. Toung JM, Morley M, Li M, Cheung VG (2011) RNA-sequence analysis of human B-cells. Genome Res 21:991–998
3. Koehler R, Issac H, Cloonan N, Grimmond SM (2011) The uniqueome: a mappability resource for short-tag sequencing. Bioinformatics 27:272–274
4. Robinson JT, Thorvaldsdottir H, Winckler W, Guttman M, Lander ES, Getz G, Mesirov JP (2011) Integrative genomics viewer. Nat Biotechnol 29:24–26
5. Paszkiewicz K, Studholme DJ (2010) De novo assembly of short sequence reads. Brief Bioinform 11:457–472
6. Schatz MC, Witkowski J, McCombie WR (2012) Current challenges in de novo plant genome sequencing and assembly. Genome Biol 13:243
7. McIntyre LM, Lopiano KK, Morse AM, Amin V, Oberg AL, Young LJ, Nuzhdin SV (2011) RNA-seq: technical variability and sampling. BMC Genomics 12:293
8. Trapnell C, Williams BA, Pertea G, Mortazavi A, Kwan G, van Baren MJ, Salzberg SL, Wold BJ, Pachter L (2010) Transcript assembly and quantification by RNA-Seq reveals unannotated transcripts and isoform switching during cell differentiation. Nat Biotechnol 28:511–515
9. Kharchenko PV, Tolstorukov MY, Park PJ (2008) Design and analysis of ChIP-seq experiments for DNA-binding proteins. Nat Biotechnol 26:1351–1359
10. Auerbach RK, Euskirchen G, Rozowsky J, Lamarre-Vincent N, Moqtaderi Z, Lefrancois P, Struhl K, Gerstein M, Snyder M (2009) Mapping accessible chromatin regions using Sono-Seq. Proc Natl Acad Sci USA 106:14926–14931
11. Li R, Zhu H, Ruan J, Qian W, Fang X, Shi Z, Li Y, Li S, Shan G, Kristiansen K et al (2010) De novo assembly of human genomes with massively parallel short read sequencing. Genome Res 20:265–272
12. Simpson JT, Wong K, Jackman SD, Schein JE, Jones SJ, Birol I (2009) ABySS: a parallel assembler for short read sequence data. Genome Res 19:1117–1123
13. Zerbino DR, Birney E (2008) Velvet: algorithms for de novo short read assembly using de Bruijn graphs. Genome Res 18:821–829
14. Butler J, MacCallum I, Kleber M, Shlyakhter IA, Belmonte MK, Lander ES, Nusbaum C, Jaffe DB (2008) ALLPATHS: de novo assembly of whole-genome shotgun microreads. Genome Res 18:810–820
15. Li H, Durbin R (2009) Fast and accurate short read alignment with Burrows-Wheeler transform. Bioinformatics 25:1754–1760
16. 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:R25
17. Chen K, Wallis JW, McLellan MD, Larson DE, Kalicki JM, Pohl CS, McGrath SD, Wendl MC, Zhang Q, Locke DP et al (2009) BreakDancer: an algorithm for high-resolution mapping of genomic structural variation. Nat Methods 6:677–681
18. Li H, Handsaker B, Wysoker A, Fennell T, Ruan J, Homer N, Marth G, Abecasis G, Durbin R (2009) The sequence alignment/map format and SAMtools. Bioinformatics 25:2078–2079
19. McKenna A, Hanna M, Banks E, Sivachenko A, Cibulskis K, Kernytsky A, Garimella K, Altshuler D, Gabriel S, Daly M et al (2010) The genome analysis toolkit: a MapReduce framework for analyzing next-generation DNA sequencing data. Genome Res 20:1297–1303
20. Li H, Ruan J, Durbin R (2008) Mapping short DNA sequencing reads and calling variants using mapping quality scores. Genome Res 18:1851–1858
21. Grabherr MG, Haas BJ, Yassour M, Levin JZ, Thompson DA, Amit I, Adiconis X, Fan L, Raychowdhury R, Zeng Q et al (2010) Full-length transcriptome assembly from RNA-Seq data without a reference genome. Nat Biotechnol 29:644–652
22. Langmead B, Hansen KD, Leek JT (2010) Cloud-scale RNA-sequencing differential expression analysis with Myrna. Genome Biol 11:R83
23. Zhang Y, Liu T, Meyer CA, Eeckhoute J, Johnson DS, Bernstein BE, Nusbaum C, Myers RM, Brown M, Li W et al (2008) Model-based analysis of ChIP-Seq (MACS). Genome Biol 9:R137

24. Feng J, Liu T, Zhang Y (2011) Using MACS to identify peaks from ChIP-Seq data. Curr Protoc Bioinformatics. Chapter 2: Unit 2 14

25. Zang C, Schones DE, Zeng C, Cui K, Zhao K, Peng W (2009) A clustering approach for identification of enriched domains from histone modification ChIP-Seq data. Bioinformatics 25:1952–1958

26. Danecek P, Auton A, Abecasis G, Albers CA, Banks E, DePristo MA, Handsaker RE, Lunter G, Marth GT, Sherry ST et al (2011) The variant call format and VCFtools. Bioinformatics 27:2156–2158