Chapter 5

Bioinformatics Analysis for Cell-Free Tumor DNA Sequencing Data

Shifu Chen, Ming Liu, and Yanqing Zhou

Abstract

As a major biomarker of liquid biopsy, cell-free tumor DNA (ctDNA), which can be extracted from blood, urine, or other circulating liquids, is able to provide comprehensive genetic information of tumor and better overcome the tumor heterogeneity problem comparing to tissue biopsy. Developed in recent years, next-generation sequencing (NGS) is a widely used technology for analyzing ctDNA. Although the technologies of processing ctDNA samples are mature, the task to detect low mutated allele frequency (MAF) variations from noisy sequencing data remains challenging. In this chapter, the authors will first explain the difficulties of analyzing ctDNA sequencing data, review related technologies, and then present some novel bioinformatics methods for analyzing ctDNA NGS data in better ways.

Key words Liquid biopsy, Circulating tumor DNA, ctDNA, Gene fusion, CNV, Mutation visualization, OpenGene

1 Introduction

Cell-free DNA (cfDNA) is the extracellular DNA fragments, which is mainly derived from apoptosis, as well as part from necrosis and active cell release. For cancer patients, both the normal and tumor cells are the source of cfDNA, and the fraction from tumor cells are called ctDNA (cell-free tumor DNA), which can be isolated from secretions, excretions, and body fluids [1–3]. The existence of cfDNA was first reported by Mandel and Metais in 1948, but it was not until in 1977 that it was found to have a correlation with cancer [4].

1.1 ctDNA and Its Applications

The possibility to use cfDNA as materials for liquid biopsy for cancer bases on the fact that all cancers are a disease of DNA alterations, including genomic variations like mutations, translocations, amplifications, gene loss, etc., and epigenetic variations. Collectively, these DNA changes determine a set of tumor-specific variations of each tumor against the background of normal DNA

Tao Huang (ed.), *Computational Systems Biology: Methods and Protocols*, Methods in Molecular Biology, vol. 1754,
https://doi.org/10.1007/978-1-4939-7717-8_5, © Springer Science+Business Media, LLC, part of Springer Nature 2018

from other tissues. Since both normal and tumor cells shed DNA into the circulation, in general there is a relatively low abundance of tumor-specific cfDNA in the total cfDNA. Before a DNA fragment is sequenced, we do not know whether it is derived from a tumor cell. So we can only do cfDNA sequencing in all at this stage and have no way to merely do ctDNA sequencing. However, for applications in cancer field, ctDNA is a term much better known and more widely used by researchers. So in this chapter, we will use both ctDNA sequencing and cfDNA sequencing. When we mention ctDNA sequencing for cancer patients, be noted that it is the same as cfDNA sequencing.

In general, cfDNA dynamically reflects the overall state of the body. The tumor-specific ctDNA are good biomarkers for early detection and prevention, assessing minimal residual disease and prognosis, monitoring tumor burden, and guiding therapies. It allows a relatively noninvasive repeated serial sampling for continuous monitoring of disease. Moreover, the ability to identify specific drug-sensitizing or resistance mutations in the blood of cancer patients makes liquid biopsy a good alternative method for tissue biopsies, which is often difficult or impossible to obtain for late-stage cancer patients using the conventional methods. Although lots of studies support these concepts, the use of cfDNA in clinical practice is still in its infancy and requires rigorous prospective validation studies to demonstrate the benefit of this promising analyte to facilitate the clinical decision making. Figure 1 illustrates how ctDNA can be applied for cancer diagnosis and therapy.

1.2 How Is ctDNA Sequenced

As there is a relatively low abundance of ctDNA in the total cfDNA, highly sensitive techniques are required for ctDNA detection. Advancements in next-generation sequencing (NGS) have made it possible to detect low occurrence mutations in a heterogeneous population [5].

The main steps for ctDNA sequencing include sample processing and cfDNA extraction, NGS library construction, hybrid capture, and sequencing.

CtDNA can be extracted from bodily fluids like plasma, cerebral spinal fluid, urine, pleural effusion, etc. The choice of material is related to the clinical or research purpose. The amount of cfDNA extracted from 1 ml plasma is relatively low, and an accurate quantification of cfDNA is necessary for the subsequent library construction procedure. A classical library construction process on Illumina platform includes end repair to produce blunt-ended and 5′-phosphorylated dsDNA fragments; A-tailing, during which dAMP is added to the 3′-ends of blunt-ended dsDNA library fragments; adapter ligation, during which dsDNA adapters with 3′-dTMP overhangs are ligated to 3′-A-tailed library fragments; and library amplification, which employs PCR to amplify library fragments carrying appropriate adapter sequences on both ends.

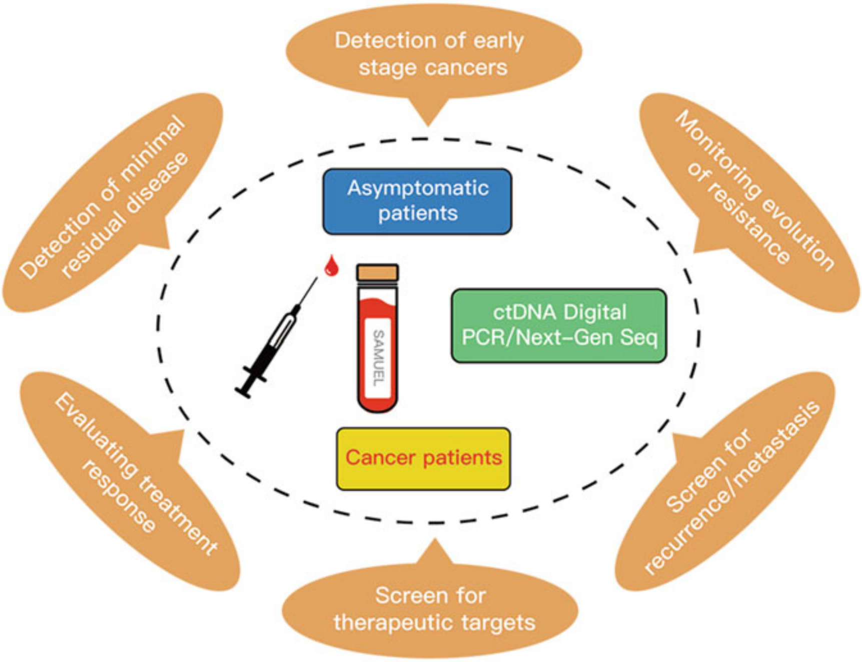

Detection of early stage cancers
Monitoring evolution of resistance
Screen for recurrence/metastasis
Screen for therapeutic targets
Evaluating treatment response
Detection of minimal residual disease
Asymptomatic patients
ctDNA Digital PCR/Next-Gen Seq
Cancer patients
SAMUEL

Fig. 1 Applications of ctDNA for cancer diagnosis and therapy. Cell-free DNA can be evaluated by different techniques like digital PCR (dPCR) or next-generation sequencing (NGS). Digital PCR can only cover a few target loci, while NGS can cover millions of loci, including the whole exome or even whole genome. CtDNA testing can be applied to search for targeted therapy drugs, evaluate treatment response, and detect early-stage cancers

The adapter ligation efficiency and the number of PCR cycles are two critical factors for this procedure. For hybridization capture, first, one should design a panel consistent with the particular application and then order the probes to a manufacturer; especially NimbleGen SeqCap EZ could be a good choice. The Illumina sequencing platform is usually applied in the last sequencing step. Figure 2 demonstrates how blood samples are processed and how the libraries are prepared and sequenced.

1.3 Difficulties of Analyzing ctDNA NGS Data

Cell-free tumor DNA is only a small fraction of cell-free DNA, especially for samples from early-stage cancer patients. This fact makes it hard to detect tumor-specific mutations. Furthermore, PCR and sequencing errors, DNA oxidative damages, and software-introduced artifacts can produce a high level of noise and introduce many false-positive mutations.

The amount of tumor-specific DNA can vary greatly from less than 0.01% to more than 90% [3]. The variability of ctDNA abundance is associated with tumor burden, stage, vascularity, cellular turnover, and response to therapy. Theoretically DNA alteration of any fraction is detectable via deep sequencing with sufficient number of molecules. However, amplification bias during PCR of

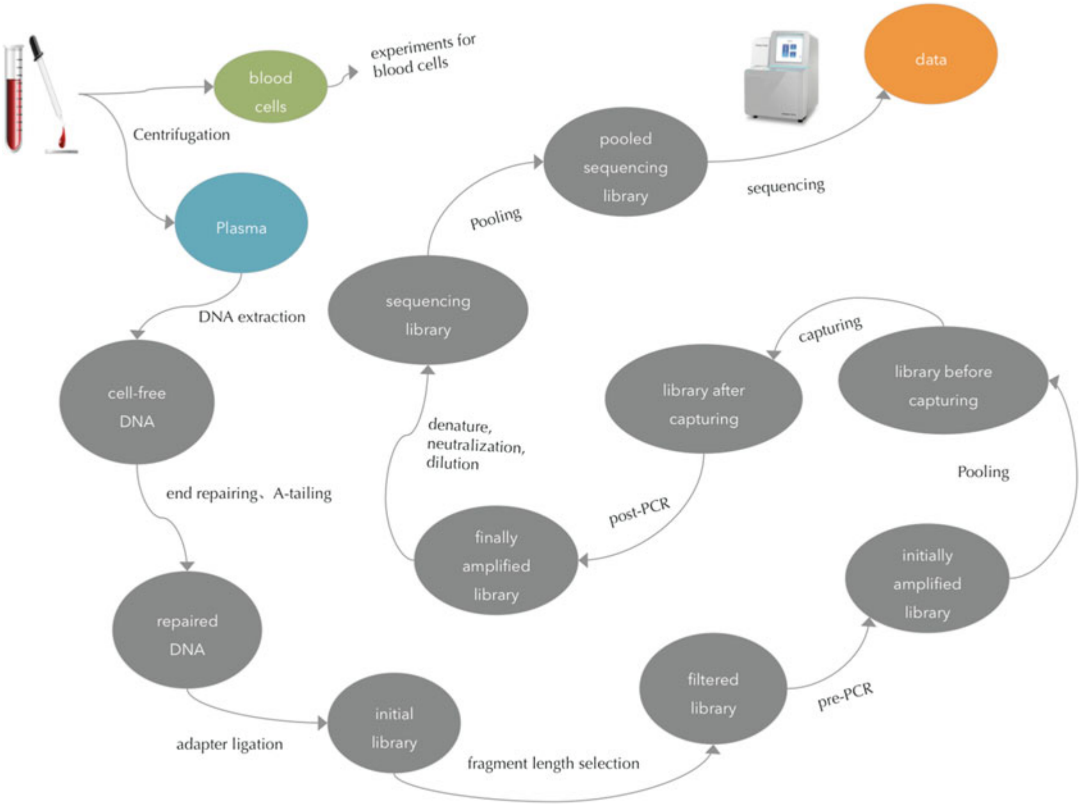


Fig. 2 The workflow of ctDNA sequencing. In this workflow, a blood sample is first processed, then DNA is extracted from the blood plasma, and an initial library is prepared. This initial library will then be amplified by PCR and be enriched with target capturing methods. The captured library can be amplified again and be processed to a sequencing library, which can be pooled and then be sequenced to generate the data

heterogeneous mixtures can result in skewed populations, and polymerase errors can result in wrong base incorporations and rearrangements. Furthermore, errors arising during sequencing process can result in approximately 0.1–1% incorrect bases calling [6], which are known as sequencing errors. Table 1 shows the error ratios of different major NGS platforms.

Library preparation can also introduce significant errors. For instance, guanine oxidation is an important source of artificial mutations because 8-oxoG tends to pair with adenine instead of cytosine [7]. Long-time heat incubations, which are common in many DNA extraction and hybrid capture protocols, can significantly increase the number of G → T substitutions. Recently a study showed that a DNA repairing process can eliminate 77% and 82% of G → T and C → A errors, respectively [8]. This study indicates DNA lesions can cause a large amount of errors.

Besides errors introduced during sample preparation and sequencing, software and analysis tools can also introduce errors. Particularly, false-positive variants can be called in the reference

Table 1
A comparison of sequencing error ratios of different sequencing platforms

Platform	Most frequent error types	Error ratio
Capillary sequencing	Single nucleotide substitutions	10^{-1}
454 GS Junior	Deletions	10^{-2}
PacBio RS	CG deletions	10^{-2}
Ion Torrent PGM	Short deletions	10^{-2}
Solid	A-T bias	2×10^{-2}
Illumina MiSeq	Single nucleotide substitutions	10^{-3}
Illumina HiSeq	Single nucleotide substitutions	10^{-3}
Illumina NextSeq	Single nucleotide substitutions	10^{-3}

genome regions with homologous sequences and repetitive sequences.

Cell-free DNA fragments are usually short and have a compact peak near 167 bp [9]. This fact increases the possibility that two different original cfDNA fragments share an identical sequence and consequently increases the difficulty to remove these duplications since the deduplication algorithms will not be able to differentiate such identical and duplicated reads caused by amplification.

In summary, detecting low-frequency mutations from the noisy ctDNA sequencing data is challenging. Conventional tools cannot handle well the ctDNA analysis tasks, and more specialized tools are therefore needed.

1.4 ctDNA Sequencing Data Analysis Pipeline

To analyze ctDNA sequencing data, a series of software tools needs to be involved. For example, the raw sequencing data from Illumina sequencers are obtained in a base calling (BCL) format. This BCL file needs to be de-multiplexed to separate FASTQ files according to sample barcodes. Then the FASTQ files would be measured with quality control tools to guarantee they fulfill the quality requirement and be filtered to remove low-quality and wrongly represented reads. Next, the filtered FASTQ files would be aligned to the reference genome with aligners, and the output should be SAM/BAM files. Then the BAM files need to be sorted and duplications removed. Then variant callers are required to process the BAM file and generate a VCF with raw variant records. Next, this VCF file should be annotated with databases like dbSNP and COSMIC. A baseline technology will be applied to mark some false-positive mutations, and then the unique reads supporting each mutation will be counted to make a complete VCF. This VCF file will then be filtered to generate a clean one and visualized with tools for interactive analysis. Finally the target mutations will

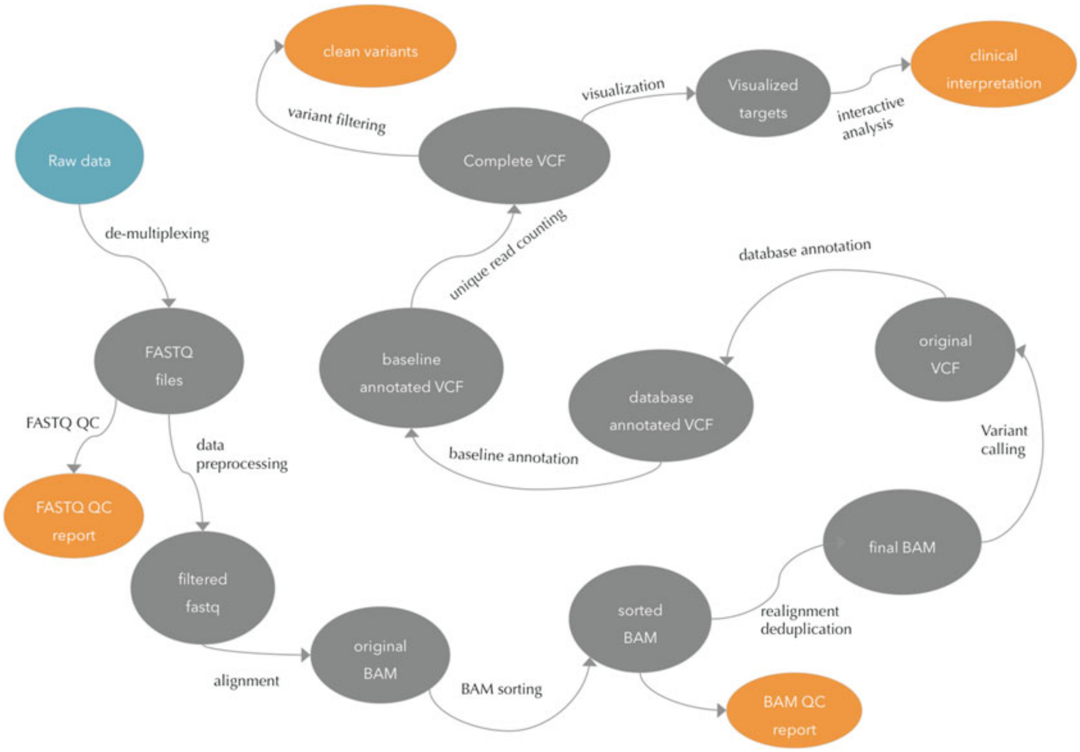


Fig. 3 A typical pipeline for analyzing ctDNA sequencing data. The raw data will be de-multiplexed to separate FASTQ files by sample indexes and then be filtered to remove bad reads. The FASTQ reads will be aligned to reference genome to generate BAM files. Variant callers will scan the sorted and processed BAM files to generate original VCF files, which can then be annotated with databases and baseline data. After a complete VCF is generated, it can be filtered to create clean variants or be visualized for variant validation

be interpreted and reported. These tools can be arranged into a pipeline. Figure 3 demonstrates a ctDNA sequencing data analysis pipeline regularly used by the authors.

For Illumina platforms, the tool bcl2fastq is used to convert BCL format files to FASTQ files. Illumina platforms support multiplexing by using different barcodes for different samples, so de-multiplexing is performed along with the conversion.

Some additional tools can perform quality control and data filtering over FASTQ files, e.g., FastQC and Trimmomatic [10]. The authors suggest using AfterQC [11], which is highly optimized for ctDNA sequencing data processing. AfterQC will be introduced in the next section.

A lot of aligners can be used to map DNA sequencing reads to reference genome, such as bowtie2 [12] and BWA [13]. According to our practice, BWA provides a better performance both in alignment quality and speed. BWA is a software package for mapping low-divergent sequences against a large reference genome. It consists of three algorithms: BWA-backtrack, BWA-SW, and

BWA-MEM. BWA-MEM is generally recommended for high-quality queries, as it is faster and more accurate. But be aware that BWA and any other aligners may still introduce misalignments, especially in reference genome regions with repetitive or homologous sequences.

The alignment process will generate a SAM file containing the alignment information that can be immediately converted to BAM, which is the binary identity of SAM. This BAM file is usually disordered and should be sorted and then indexed. The most commonly used tool to sort and index BAM files is Samtools [14], and there exist some other tools that can sort BAM faster. For example, Sambamba [15] is a high performance tool working with SAM/BAM data. Sambamba is written in D language, and its source is available at: https://github.com/lomereiter/sambamba.

After BAM file is sorted and indexed, an optional process is to apply realignment to improve the detection of insertions and deletions (INDELs). Some tools like ABRA [16] can perform assembly-based realignment to output cleaner INDELs, but these tools are usually slow. Quality control of BAM files can be applied now to evaluate the data's alignment quality and detect unwanted biases. This process can be done with tools like Qualimap [17].

The subsequent process is deduplication. Samtools rmdup and Picard MarkDuplicates (http://picard.sourceforge.net) are commonly used to identify and collapse read duplication based on reads' mapping coordinates and quality scores. Since cfDNA fragments are short and their length distribution is compactly close to 167 bp, lots of reads derived from different original DNA fragments may share identical mapping coordinates, and they should not be considered as duplication. So we do not suggest using Samtools rmdup or Picard MarkDuplicates for deduplication, and we will discuss new methods and strategies in the next section.

Variant calling is the key process following the BAM operations (sort, realign, dedup). Cancer genomes are known to harbor a wide range of mutations, including single nucleotide variants (SNVs), multiple nucleotide variants (MNVs), small insertions and deletions (INDELs), and complex variants, such as copy number variants (CNVs) and gene fusions. A number of variant callers, such as GATK HaplotypeCaller [18], FreeBayes (https://github.com/ekg/freebayes), MuTect2 [19], and VarScan2 [20], can be used to call SNV, MNV, and small INDELs. According to our experience, GATK HaplotypeCaller and FreeBayes are not good at calling ctDNA's low-frequency somatic mutations from ultra-deep sequencing data, since they are originally designed for genotyping and discovering genetic polymorphism. MuTect2 is much better in calling somatic mutations, especially with tumor-normal paired data. However, it just works well with tissue sequencing data but is not sensitive enough to detect low-frequency mutations in ctDNA sequencing data. VarScan2 is very sensitive in detecting low-frequency mutations but likely to report a large amount of false-positive mutations. So we could not find a perfect variant caller for detecting low-frequency mutations in such ultra-deep NGS data like ctDNA sequencing data. Currently we suggest VarScan2, combined with strict variant filtering. Be aware that some variant callers, like GATK HaplotypeCaller, cannot scale well with depth and typically downsample (randomly remove portions of data) to improve their computational performance. However downsampling can significantly reduce the sensitivity to detect low allele frequency mutations and is not suggested for ctDNA sequencing data analysis.

After the variant calling process is done, the original VCF file is obtained. This VCF file can be annotated with annotation tools like ANNOVAR [21] to obtain coding sequence and protein changes and compare with databases like dbSNP, ClinVar, and COSMIC.

A mutation baseline will be used to annotate each variant for how many times this variant was recorded in the past data. This information can be used to filter false-positive mutations caused by software artifacts and other regular systematic errors. Baseline technology will be introduced in next section.

To calculate the supporting read number for each mutation more accurately, we can consider the reads with the same mapping coordinate as a single unique read. A tool called MrBam (https://github.com/OpenGene/MrBam) is used to count each mutation's unique reference support and unique alternative support.

After the unique read counting is done, we obtain a complete VCF file. The records in this VCF file can be added into the mutation baseline. This VCF file can be filtered according to different conditions to remove as many false-positive mutations as possible. A white list, which consists of the important clinical targets (i.e., cancer druggable mutation targets), is usually used in this filtering process to avoid the important target mutations being filtered out unexpectedly.

On another track, the called variants can be visualized with tools like MutScan (https://github.com/OpenGene/MutScan) to produce mutation visualization for interactive analysis. Mutations that are important for cancer diagnosis and therapy will be manually interpreted.

Besides SNVs and INDELs, another two important kinds of variants for cancer diagnosis are gene fusions and copy number variants (CNV). Most of these tools can only work with sorted BAM files. For example, DELLY [22] and Factera [23] can be used to detect gene fusions, and CNVkit (https://github.com/etal/cnvkit) can be used to detect gene amplifications from targeted DNA sequencing. One exception is that FusionDirect, a tool developed by the authors, can work with FASTQ files directly to detect target fusions.

The authors have created an open source project to demonstrate this pipeline, which is available at GitHub (https://github.com/OpenGene/ctdna-pipeline). By studying it, the readers can learn how to install the tools, prepare required databases and reference data, and try the pipeline with FASTQ files for testing.

In the pipeline presented above, more than a half of the tools are commonly used software (i.e., BWA, Samtools, and VarScan2), while the rest ones are developed by the authors (i.e., MutScan, AfterQC, and MrBam). These newly developed tools are highly optimized for ctDNA sequencing data analysis. Most of these tools are open source projects under the GitHub organization OpenGene (https://github.com/OpenGene). We will introduce some of them in the next section.

2 New Methods

Since tumor-specific DNA is only a small part of cfDNA, the mutated allele frequency (MAF) of somatic mutations in ctDNA is usually very low [24]. To detect mutations with such low MAF, we should apply target capturing and ultra-deep sequencing (i.e., 10,000× or deeper). However, sequencing errors and experiment errors (i.e., PCR errors) in such ultra-deep sequencing can cause high-level background noise and make it difficult to detect mutations from ctDNA NGS data with both high sensitivity and specificity. Furthermore, the detection of gene fusions is also difficult since cfDNA fragments are usually short and tumor-specific DNA fragments are too few. Since the copy number change in tumor cells only results in a slight difference of total cfDNA's copy number, detecting copy number variation (CNV) is even more challenging than detecting fusions.

In this section, we will present some new methods to partially address the problems listed above. Some of them are developed by the authors and has been used in our regular pipelines.

2.1 Better Data Preprocessing

Data preprocessing is an important step to obtain cleaner data for downstream analysis. For NGS raw data (in FASTQ format), it is necessary to discard low-quality reads, cut adapters, and apply other filters. Furthermore, quality control (QC) methods are also needed to make sure the data fulfill the quality requirements.

Some good tools can perform quality control, such like FastQC with per-base and per-sequence quality profiling functions and PRINSEQ [25] with FASTA/FASTQ statistics capability, while some other tools can perform read trimming, such like Trimmomatic [10] and SolexaQA [26]. Since the way to do data filtering depends on the QC result and the filtered data also need a post-filtering QC, a tool with both rich QC and filtering functions is still wanted.

Since cfDNA fragments are usually short (~167 bp) [9], 2 × 150 paired-end sequencing will result in overlapped read pairs. Based on this fact, we can perform overlapping analysis for paired-end sequencing data. When the DNA template length is less than twice of the sequencing length, the pair of reads will be overlapped. Note that each base in the overlapped region is actually sequenced twice, so the inconsistency of these pairs may reflect the sequencing errors.

AfterQC [11] is a tool developed by authors to address lots of practical sequencing data quality control and filtering problems. In addition to regular quality control functions like per-cycle base content and quality statistics, AfterQC also provides lots of new functions like automatic trimming and overlapping analysis. For example, we found that some sequencers (like Illumina NextSeq series) may output lots of polyX reads with high-quality scores. AfterQC can remove them using its polyX filter, whereas normal quality filters cannot. We also found that if the amplification or sequencing process has a serious strand bias, the sequence reads will show K-MER count bias (i.e., the counts of ATCGATCG and its reverse complement CGATCGAT are significantly different). Based on this finding, AfterQC provides K-MER counting based strand bias profiling. Another major contribution of this tool is overlapping analysis for paired-end sequencing data, which can be used to profile the sequencing error rate and use it for error base correction or removing. For every input of a single or pair of FASTQ files, AfterQC outputs an HTML report, which contains the quality control and data filtering summary, and a list of interactive figures. Table 2 shows feature comparison of AfterQC and other NGS quality control or filtering tools.

AfterQC is designed to process FASTQ files in batches. It goes through a folder with all FASTQ files (can be single-end or paired-end output), which are typically data of a sequencing run for different samples, and passes each FASTQ file or pair into the QC and filtering pipeline. First, AfterQC will run a bubble detection to find the bubbles raised in the sequencing process; second, a pre-filtering QC will be conducted to profile the data with per-cycle base content and quality curves; third, AfterQC will perform automatic read trimming based on data quality profiling; fourth, each read will be filtered by bubble filter, polyX filter, quality filter, and overlapping analysis filters, and the ones failed to pass these filters will be discarded as bad reads; fifth, an error correction based on overlapping analysis will be applied for paired-end sequencing data; finally, AfterQC will store the good reads, perform post-filtering QC profiling, and generate HTML reports.

AfterQC can handle automatic trimming of FASTQ data. There are two strategies for trimming, local strategy and global strategy. Some tools, like Trimmomatic, apply local strategy, which perform trimming read by read. However, local trimming strategy

Table 2
Feature comparison of FastQC, Trimmomatic, Cutadapt, and AfterQC

	FastQC	Trimmomatic	Cutadapt	AfterQC
Quality control	Rich functionality	Few functionality	Few functionality	Rich functionality
Auto trimming	None	Read by read	Read by read	Global trimming
Cutting adapter	None	Single-end/pair-end	Single-end/pair-end	Paired-end only
PolyX filtering	None	None	None	Supported
Figure plotting	Static	Static	Static	Interactive
Overlap analysis	None	Cutting adapter only	None	Supported with error correction
Sequence error profiling	None	None	None	Supported
Bubble detection	None	None	None	Supported
Programming language	Java	Java	Python	Python, C
Speed	Fast	Fast	Fast	Fast only for single-end data

has some drawbacks. The first drawback is that local trimming only uses the quality information for trimming, but it cannot utilize the global statistical information to discover the abnormal cycles. The second drawback is local trimming results in unaligned trimming, which means duplicated reads may be trimmed differently and consequently causes some deduplication tools like Picard to fail. Most of these deduplication tools detect duplications only by clustering reads with identical mapping positions. In contrast, AfterQC implements a global trimming strategy, i.e., it trims all the reads in the same manner. An algorithm is used to determine how many cycles to trim in the front and tail, which is based on the segmentation of the per-cycle base content curves and base quality curves.

A major advantage of AfterQC is the overlapping analysis. Let T denote the length of a sequenced DNA template, and S denote the length of paired-end sequencing length, then the pair of reads will totally overlap if $T \leq S$, will overlap with a length of $2S - T$, if $S < T < 2S$, and will not overlap if $2S \leq T$. AfterQC checks how does each pair of reads overlap based on edit distance optimization. For a pair of reads $R1$ and $R2$, let O be the offset, and we place $R2$ under $R1$, then we will have vertically aligned subsequences $R1o$ and $R2o$, and we can calculate their edit distance $ed.(R1o, R2o)$. The method optimizes offset O to obtain the minimal edit distance,

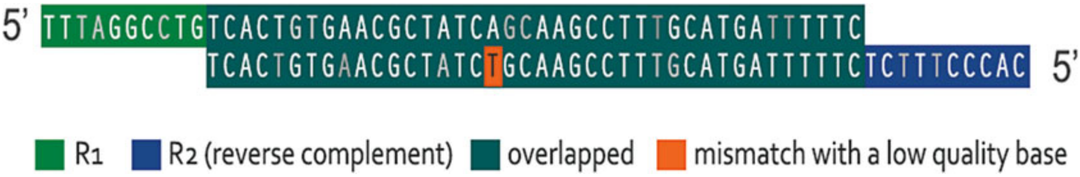

5'
TTTAGGCCTGTCACTGTGAACGCTATCAGCAAGCCTTTGCATGATTTTTC
TCACTGTGAACGCTATCTGCAAGCCTTTGCATGATTTTTCTCTTTCCCAC
5'
R1
R2 (reverse complement)
overlapped
mismatch with a low quality base

Fig. 4 How AfterQC's overlapping analysis works. The edit distance of the overlapped subsequences is 1. A mismatch pair is found with a high-quality base *A* and a very low-quality base *T*. This *T* will be recognized as wrongly represented and can be corrected

$ed.(R1o-1, R2o-1) > ed.(R1o, R2o) < ed.(R1o+1, R2o+1)$. Figure 4 shows an example of how AfterQC's overlapping analysis works.

Based on overlapping analysis, AfterQC can detect mismatches. If the mismatched pair has unbalanced quality scores, which means one base has high-quality score (i.e., >Q30) and the other has very low-quality score (i.e., <Q15), AfterQC can automatically correct the base with low quality. If the quality scores are not unbalanced, AfterQC can mask them by changing the bases to N or assigning zero quality scores to them. Based on the mismatches, AfterQC can evaluate the sequencing error rate and profile the sequencing error transform distribution (i.e., how many bases are T but sequenced as C).

Overlapping analysis can be used for automatic adapter cutting. In the overlapping analysis process, we get the optimal offset *O* for the best local alignment of each pair. The overlapping length of this pair can be directly calculated using the offset *O*. If *O* is found negative, the bases outside overlapping region will be considered as a part of adapter sequences and then be cut automatically.

AfterQC is an open source tool: https://github.com/OpenGene/AfterQC. It is implemented in Python and C++, with PyPy support enabled. AfterQC generates a standalone HTML report for each input, with figures plot by Plotly. A sample report can be found at: http://opengene.org/AfterQC/report.html.

2.2 Molecular Barcoding Sequencing and Its Data Analysis

The potential of NGS deep sequencing for ctDNA was hampered by systemic errors introduced by PCR and sequencing methods [27, 28]. Molecular indexing combined with deep sequencing holds great promise to break the limit imposed by PCR and sequencing errors and enables the detection of rare and ultra-rare mutations [29, 30].

Tagging individual templates with molecular barcodes has been proposed and reported since 2007 [31]. The molecular barcodes or molecular indexes have been given various names, such as unique identifiers (UID) [29], unique molecular identifiers (UMI) [32], primer ID [30], duplex barcodes [33], etc. They are usually designed as a string of totally random nucleotides (such as NNNNNNNN), partially degenerate nucleotides (such as

NNNRNYNN), or defined nucleotides (when template molecules are limited). UID or UMI could be introduced to targeted templates by ligation or through primers during PCR or reverse transcription.

Tagging DNA fragments with UIDs or duplex barcodes has been shown to reduce errors and improve sequencing accuracy, as true mutations could be distinguished from PCR errors or sequencing errors based on the consensus reads sharing the same UID. At present, classic tag-based methods are SafeSeq, CircleSeq, and duplex sequencing [34]. SafeSeq is a single-stranded tagging method based on "barcoding." An alternative to single-stranded tagging based on shear points is the circle sequencing methodology which utilizes the strand displacement activity of Phi29's DNA polymerase to generate multiple copies of circularized DNA molecules in tandem prior to amplification. However, both of these two methods cannot distinguish true variants from artificial variants introduced during the initial rounds of PCR amplification. In contrast duplex sequencing resolved these types of errors by tagging both strands of dsDNA, exploiting the fact that DNA naturally exists as a double-stranded entity, with one molecule reciprocally encoding the sequence information of its complement. Table 3 compares claimed error ratios of SafeSeq, CircleSeq, and duplex sequencing.

The analysis of molecular barcoding-enabled sequencing data can be divided into three steps.

The first step is extracting the UID. Be noted that the barcodes ligated to the original DNA template are usually made by DNA synthesis technology, which usually has high error ratio. For example, if 8-nt barcode is designed, we still have a chance to get 7-nt or 9-bt barcode due to synthesis error. To address this issue, a fixed sequence that consists of a few bases (usually three to five bases) is usually used to indicate the boundary of UID and original DNA sequence. Splitting algorithms should seek for this flag near the designed position, and typically the algorithm should allow one base mismatch to enable DNA synthesis or sequencing error tolerance. By using special adapters, some molecular barcoding methods place the UID on the multiplexing index positions (I7 or I5 index

Table 3
A comparison of different molecular barcoding methods.

Method	Claimed error ratio
SafeSeq	1.4×10^{-5}
CircleSeq	7.6×10^{-6}
Duplex sequencing	5×10^{-8}

for Illumina TrueSeq). UID extraction is much easier in this case since it can be taken directly from the sample index. This process is done with FASTQ data.

The second step is clustering the reads derived from the same original DNA. These reads should share very similar UID and mapping coordination. But due to the presence of PCR and sequencing errors, they are not required to be completely identical. Usually one base substitution mismatch is tolerated, and loose clustering methods can allow mismatches of INDELs or more than one substitution. This process is usually done with sorted BAM files, but it can also be done with FASTQ files based on sequence clustering algorithms.

The final step is generating consensus read for each read cluster. First, the reads in same cluster should be aligned together. This process can be done with a multiple sequence alignment tool like Clustal [35]. The complete multiple sequence alignment is usually time-consuming, and if we limit the number of mismatched substitutions and INDELs, some naive methods can run much faster. After the alignment is done, the consensus read can be generated by scanning it from front to tail. For each position, all bases in this position will be used to vote for the consensus base, according to their quality scores. For the positions with completely identical bases, the quality score of this consensus base can be adjusted a bit higher, and, vice versa, for a position that shows no consensus, the quality score of result base can be adjusted to be lower. In case when only two reads are clustered, if the two bases in the same positions are different but both have high-quality scores, this position can then be masked with N or zero quality score.

2.3 Baseline Methods

NGS data have different kinds of errors. Some errors, like sequencing error and PCR error, are random and can happen with any nucleotide at any genome position, although with some biases. Some errors are more regular, such like errors caused by misalignment usually happening in genome's high repetitive regions. These regular errors can be eliminated with baseline technologies.

Baseline technology is to combine and store all related detected mutations and other related information from as many samples as possible and then make statistics of these data and provide interfaces for querying and updating. Baseline data is usually stored in database, so it can utilize the standard SQL language for inserting, updating, deleting, and querying. Two different types of databases can be used: row-oriented database and column-oriented database. Row-oriented database is the mainstream form of relational database, like MySQL and PostgreSQL, whereas column-oriented database is less known, like Infobright and MonetDB. Row-oriented databases can support online transaction processing (OLTP) and are highly optimized for relational queries, whereas column-oriented databases can provide higher data compression ratio.

The baseline should store each mutation with its chromosome, position, reference, and alternative bases, combined with numbers of mutated reads and total depth. With this baseline, we then can count how many times a mutation of specific location with specific alteration has been detected, what its average MAF is, and what the mutated read number is.

Since some mutations can be detected in many different types of cancers, a better solution is to build a specific baseline with data sequenced from healthy people. Then this baseline can be used to filter false-positive mutations. When a mutation is called, its baseline-repeating number will be evaluated. If baseline-repeating number is too high, then this mutation can be considered as a false positive and need to be evaluated carefully.

Another usage of baseline is to detect hotspot mutations, both somatic and germline ones. By mining hot mutations from the baseline built with tumor individuals, we can find target mutations with potential to be biomarkers.

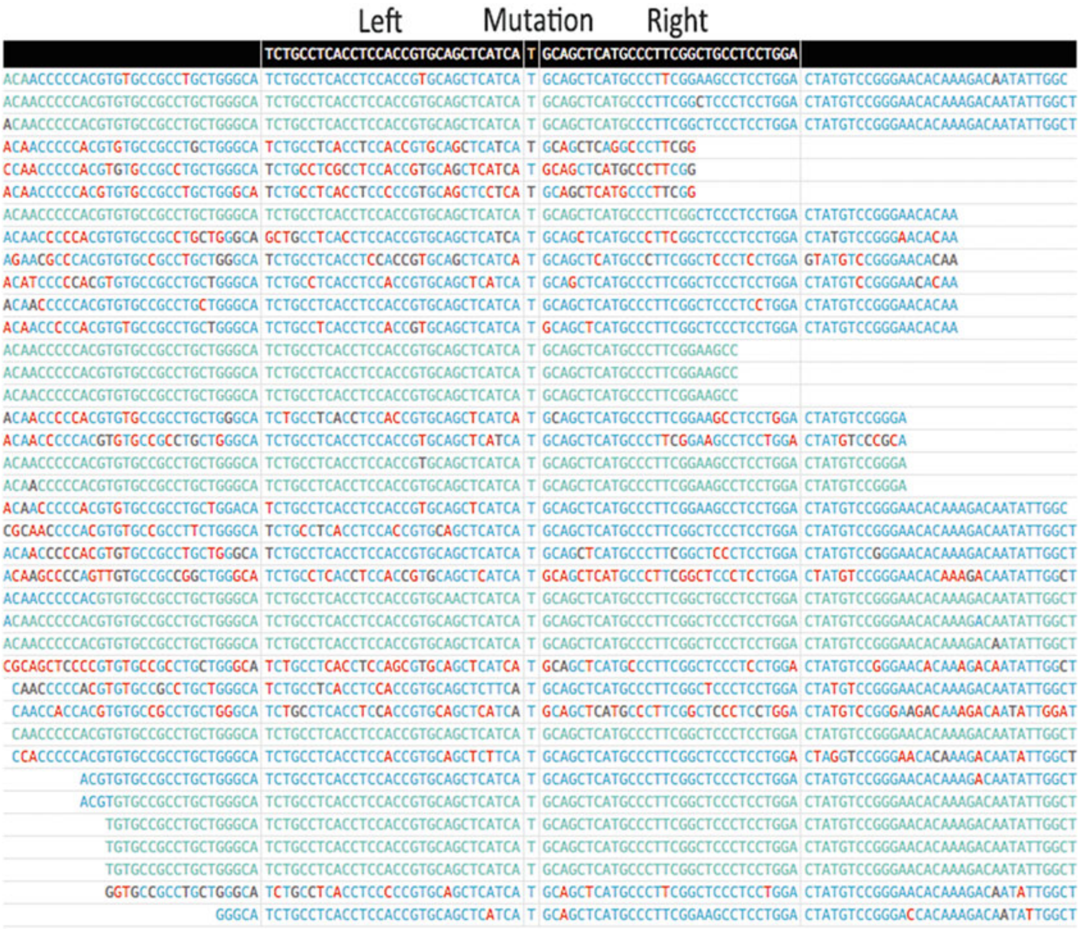
2.4 Target Variant Detection by Scanning FASTQ Data Directly

Regular mutation detection pipeline for NGS data usually involves many tools in different steps. These tools may cause information loss due to different filters applied and may finally cause miss detection of true mutations, especially the ones with low MAF. This kind of false negatives caused by data analysis is not acceptable in clinical applications, since it will make the patient miss an opportunity for better treatment.

On the contrary, false-positive detection of these key mutations should be also avoided since it can lead to an expensive but ineffective treatment and may even cause serious adverse reactions. Regular NGS pipeline can detect a lot of substitutions and INDELs and unavoidably raise false positives. Especially, caused by inaccurate reference genome mapping of aligners, a large percentage of the INDELs called in genome's high repetitive regions are false positives.

The authors have developed some tools that can detect target mutations by just scanning raw FASTQ data, without doing any alignment and variant calling. One tool is MutScan, which is built on error-tolerant string searching algorithms and is highly optimized for speed with rolling hash and bloom filters [36]. MutScan can run in reference free mode to detect target mutations, which are predefined in the program. With a VCF file and its corresponding reference FastA files provided, MutScan can scan all the variants in the VCF and visualize them by creating a HTML file for each variant.

MutScan is ultra-sensitive and ultra-fast. It can grab mutations with as few as one mutated read supported. It can run 50× faster than a regular pipeline (AfterQC + BWA + Samtools + VarScan2), if it only scans the predefined cancer druggable targets. Furthermore, the interactive HTML reports generated by MutScan can help to



Left Mutation Right

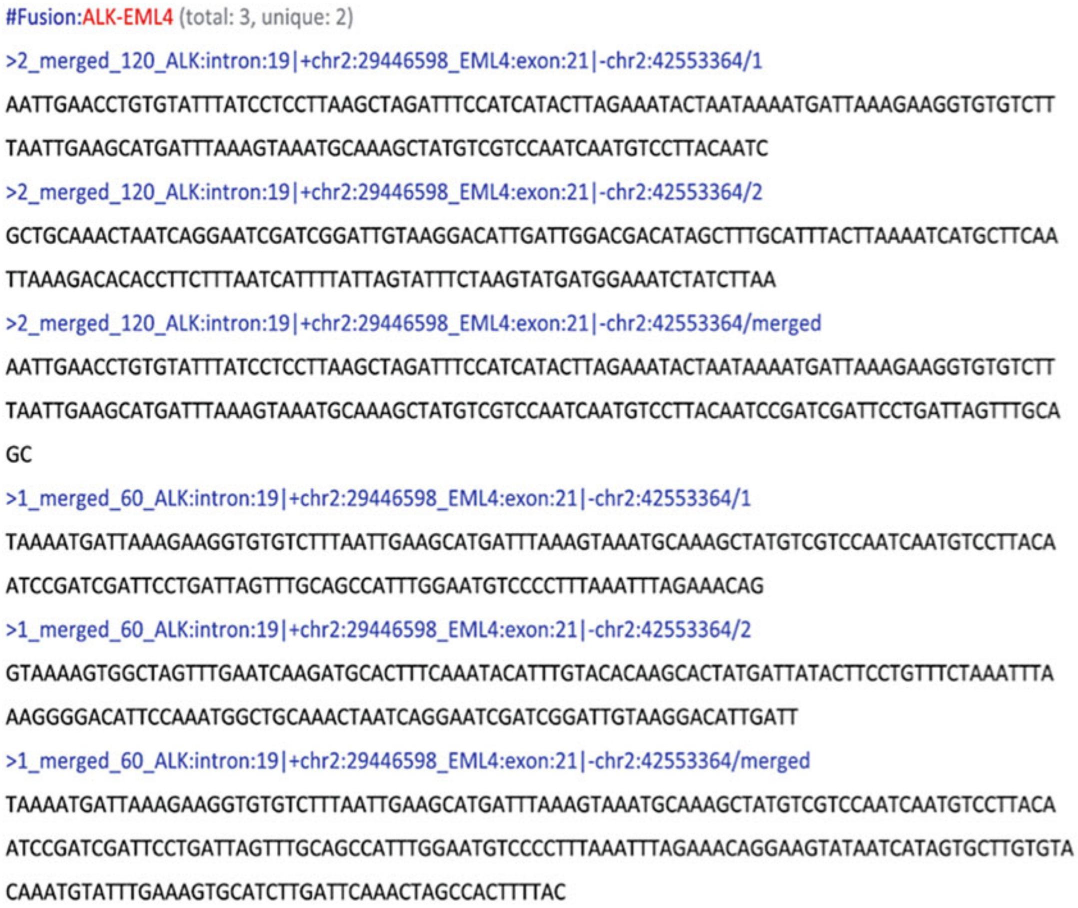
	TCTGCCTCACCTCCACCGTGCAGCTCATCA	T	GCAGCTCATGCCCTTCGGCTGCCTCCTGGA	
ACAACCCCCACGTGTGCCGCCTGCTGGGCA	TCTGCCTCACCTCCACCGTGCAGCTCATCA	T	GCAGCTCATGCCCTTCGGAAGCCTCCTGGA	CTATGTCCGGGAACACAAAGACAATATTGGC
ACAACCCCCACGTGTGCCGCCTGCTGGGCA	TCTGCCTCACCTCCACCGTGCAGCTCATCA	T	GCAGCTCATGCCCTTCGGCTCCCTCCTGGA	CTATGTCCGGGAACACAAAGACAATATTGGCT
ACAACCCCCACGTGTGCCGCCTGCTGGGCA	TCTGCCTCACCTCCACCGTGCAGCTCATCA	T	GCAGCTCATGCCCTTCGGCTCCCTCCTGGA	CTATGTCCGGGAACACAAAGACAATATTGGCT
ACAACCCCCACGTGTGCCGCCTGCTGGGCA	TCTGCCTCACCTCCACCGTGCAGCTCATCA	T	GCAGCTCAGGCCCTTCGG	
CCAACCCCCACGTGTGCCGCCTGCTGGGCA	TCTGCCTCGCCTCCACCGTGCAGCTCATCA	T	GCAGCTCATGCCCTTCGG	
ACAACCCCCACGTGTGCCGCCTGCTGGGCA	TCTGCCTCACCTCCCCCGTGCAGCTCCTCA	T	GCAGCTCATGCCCTTCGG	
ACAACCCCCACGTGTGCCGCCTGCTGGGCA	TCTGCCTCACCTCCACCGTGCAGCTCATCA	T	GCAGCTCATGCCCTTCGGCTCCCTCCTGGA	CTATGTCCGGGAACACAA
ACAACCCCCACGTGTGCCGCCTGCTGGGCA	GCTGCCTCACCTCCACCGTGCAGCTCATCA	T	GCAGCTCATGCCCTTCGGCTCCCTCCTGGA	CTATGTCCGGGAACACAA
AGAACGCCCACGTGTGCCGCCTGCTGGGCA	TCTGCCTCACCTCCACCGTGCAGCTCATCA	T	GCAGCTCATGCCCTTCGGCTCCCTCCTGGA	GTATGTCCGGGAACACAA
ACATCCCCCACGTGTGCCGCCTGCTGGGCA	TCTGCCTCACCTCCACCGTGCAGCTCATCA	T	GCAGCTCATGCCCTTCGGCTCCCTCCTGGA	CTATGTCCGGGAACACAA
ACAACCCCCACGTGTGCCGCCTGCTGGGCA	TCTGCCTCACCTCCACCGTGCAGCTCATCA	T	GCAGCTCATGCCCTTCGGCTCCCTCCTGGA	CTATGTCCGGGAACACAA
ACAACCCCCACGTGTGCCGCCTGCTGGGCA	TCTGCCTCACCTCCACCGTGCAGCTCATCA	T	GCAGCTCATGCCCTTCGGCTCCCTCCTGGA	CTATGTCCGGGAACACAA
ACAACCCCCACGTGTGCCGCCTGCTGGGCA	TCTGCCTCACCTCCACCGTGCAGCTCATCA	T	GCAGCTCATGCCCTTCGGAAGCC	
ACAACCCCCACGTGTGCCGCCTGCTGGGCA	TCTGCCTCACCTCCACCGTGCAGCTCATCA	T	GCAGCTCATGCCCTTCGGAAGCC	
ACAACCCCCACGTGTGCCGCCTGCTGGGCA	TCTGCCTCACCTCCACCGTGCAGCTCATCA	T	GCAGCTCATGCCCTTCGGAAGCC	
ACAACCCCCACGTGTGCCGCCTGCTGGGCA	TCTGCCTCACCTCCACCGTGCAGCTCATCA	T	GCAGCTCATGCCCTTCGGAAGCCTCCTGGA	CTATGTCCGGGA
ACAACCCCCACGTGTGCCGCCTGCTGGGCA	TCTGCCTCACCTCCACCGTGCAGCTCATCA	T	GCAGCTCATGCCCTTCGGAAGCCTCCTGGA	CTATGTCCCGCA
ACAACCCCCACGTGTGCCGCCTGCTGGGCA	TCTGCCTCACCTCCACCGTGCAGCTCATCA	T	GCAGCTCATGCCCTTCGGAAGCCTCCTGGA	CTATGTCCGGGA
ACAACCCCCACGTGTGCCGCCTGCTGGGCA	TCTGCCTCACCTCCACCGTGCAGCTCATCA	T	GCAGCTCATGCCCTTCGGAAGCCTCCTGGA	CTATGTCCGGGA
ACAACCCCCACGTGTGCCGCCTGCTGGACA	TCTGCCTCACCTCCACCGTGCAGCTCATCA	T	GCAGCTCATGCCCTTCGGAAGCCTCCTGGA	CTATGTCCGGGAACACAAAGACAATATTGGC
CGCAACCCCACGTGTGCCGCCTTCTGGGCA	TCTGCCTCACCTCCACCGTGCAGCTCATCA	T	GCAGCTCATGCCCTTCGGCTCCCTCCTGGA	CTATGTCCGGGAACACAAAGACAATATTGGCT
ACAACCCCCACGTGTGCCGCCTGCTGGGCA	TCTGCCTCACCTCCACCGTGCAGCTCATCA	T	GCAGCTCATGCCCTTCGGCTCCCTCCTGGA	CTATGTCCGGGAACACAAAGACAATATTGGCT
ACAAGCCCCAGTTGTGCCGCCGGCTGGGCA	TCTGCCTCACCTCCACCGTGCAGCTCATCA	T	GCAGCTCATGCCCTTCGGCTCCCTCCTGGA	CTATGTCCGGGAACACAAAGACAATATTGGCT
ACAACCCCCACGTGTGCCGCCTGCTGGGCA	TCTGCCTCACCTCCACCGTGCAACTCATCA	T	GCAGCTCATGCCCTTCGGCTGCCTCCTGGA	CTATGTCCGGGAACACAAAGACAATATTGGCT
ACAACCCCCACGTGTGCCGCCTGCTGGGCA	TCTGCCTCACCTCCACCGTGCAGCTCATCA	T	GCAGCTCATGCCCTTCGGCTCCCTCCTGGA	CTATGTCCGGGAACACAAAGACAATATTGGCT
ACAACCCCCACGTGTGCCGCCTGCTGGGCA	TCTGCCTCACCTCCACCGTGCAGCTCATCA	T	GCAGCTCATGCCCTTCGGCTCCCTCCTGGA	CTATGTCCGGGAACACAAAGACAATATTGGCT
CGCAGCTCCCCGTGTGCCGCCTGCTGGGCA	TCTGCCTCACCTCCAGCGTGCAGCTCATCA	T	GCAGCTCATGCCCTTCGGCTCCCTCCTGGA	CTATGTCCGGGAACACAAAGACAATATTGGCT
CAACCCCCACGTGTGCCGCCTGCTGGGCA	TCTGCCTCACCTCCACCGTGCAGCTCTTCA	T	GCAGCTCATGCCCTTCGGCTCCCTCCTGGA	CTATGTCCGGGAACACAAAGACAATATTGGCT
CAACCACCACGTGTGCCGCCTGCTGGGCA	TCTGCCTCACCTCCACCGTGCAGCTCATCA	T	GCAGCTCATGCCCTTCGGCTCCCTCCTGGA	CTATGTCCGGGAAGACAAAGACAATATTGGAT
CAACCCCCACGTGTGCCGCCTGCTGGGCA	TCTGCCTCACCTCCACCGTGCAGCTCATCA	T	GCAGCTCATGCCCTTCGGCTCCCTCCTGGA	CTATGTCCGGGAACACAAAGACAATATTGGCT
CCACCCCCACGTGTGCCGCCTGCTGGGCA	TCTGCCTCACCTCCACCGTGCAGCTCTTCA	T	GCAGCTCATGCCCTTCGGCTCCCTCCTGGA	CTAGGTCCGGGAACACAAAGACAATATTGGCT
ACGTGTGCCGCCTGCTGGGCA	TCTGCCTCACCTCCACCGTGCAGCTCATCA	T	GCAGCTCATGCCCTTCGGCTCCCTCCTGGA	CTATGTCCGGGAACACAAAGACAATATTGGCT
ACGTGTGCCGCCTGCTGGGCA	TCTGCCTCACCTCCACCGTGCAGCTCATCA	T	GCAGCTCATGCCCTTCGGCTCCCTCCTGGA	CTATGTCCGGGAACACAAAGACAATATTGGCT
TGTGCCGCCTGCTGGGCA	TCTGCCTCACCTCCACCGTGCAGCTCATCA	T	GCAGCTCATGCCCTTCGGCTCCCTCCTGGA	CTATGTCCGGGAACACAAAGACAATATTGGCT
TGTGCCGCCTGCTGGGCA	TCTGCCTCACCTCCACCGTGCAGCTCATCA	T	GCAGCTCATGCCCTTCGGCTCCCTCCTGGA	CTATGTCCGGGAACACAAAGACAATATTGGCT
TGTGCCGCCTGCTGGGCA	TCTGCCTCACCTCCACCGTGCAGCTCATCA	T	GCAGCTCATGCCCTTCGGCTCCCTCCTGGA	CTATGTCCGGGAACACAAAGACAATATTGGCT
GGTGCCGCCTGCTGGGCA	TCTGCCTCACCTCCCCCGTGCAGCTCATCA	T	GCAGCTCATGCCCTTCGGCTCCCTCCTGGA	CTATGTCCGGGAACACAAAGACAATATTGGCT
GGGCA	TCTGCCTCACCTCCACCGTGCAGCTCATCA	T	GCAGCTCATGCCCTTCGGAAGCCTCCTGGA	CTATGTCCGGGACCACAAAGACAATATTGGCT

Fig. 5 A demonstration of MutScan's interactive HTML. The demonstrated mutation is EGFR p.T790M (hg19 chr7:55,249,071 C > T), which is an important druggable target for lung cancer. The colors of the bases indicate the quality score (green and blue mean high quality, red means low quality). Due to page size limitation, this figure is an incomplete screenshot. The full report can be found at http:/opengene.org/MutScan/report.html

visualize and validate target mutations. Figure 5 shows a demonstration of MutScan's interactive mutation pileup.

MutScan is available at: https://github.com/OpenGene/MutScan. It is written in C++ with multi-threading support. It supports both single-end and paired-end data, and for latter one, it will try to merge each pair with quality adjustment and error correction.

Another tool developed by authors is FusionDirect, which can detect gene fusions directly from raw FASTQ data. This tool also works with FASTQ files directly and requires no alignment. It can output fusion sites (genes and positions), along with the reads supporting the fusions. Figure 6 gives an example of the output of FusionDirect.



```
#Fusion:ALK-EML4 (total: 3, unique: 2)
>2_merged_120_ALK:intron:19|+chr2:29446598_EML4:exon:21|-chr2:42553364/1
AATTGAACCTGTGTATTTATCCTCCTTAAGCTAGATTTCCATCATACTTAGAAATACTAATAAAATGATTAAAGAAGGTGTGTCTT
TAATTGAAGCATGATTTAAAGTAAATGCAAAGCTATGTCGTCCAATCAATGTCCTTACAATC
>2_merged_120_ALK:intron:19|+chr2:29446598_EML4:exon:21|-chr2:42553364/2
GCTGCAAACTAATCAGGAATCGATCGGATTGTAAGGACATTGATTGGACGACATAGCTTTGCATTTACTTAAAATCATGCTTCAA
TTAAAGACACACCTTCTTTAATCATTTTATTAGTATTTCTAAGTATGATGGAAATCTATCTTAA
>2_merged_120_ALK:intron:19|+chr2:29446598_EML4:exon:21|-chr2:42553364/merged
AATTGAACCTGTGTATTTATCCTCCTTAAGCTAGATTTCCATCATACTTAGAAATACTAATAAAATGATTAAAGAAGGTGTGTCTT
TAATTGAAGCATGATTTAAAGTAAATGCAAAGCTATGTCGTCCAATCAATGTCCTTACAATCCGATCGATTCCTGATTAGTTTGCA
GC
>1_merged_60_ALK:intron:19|+chr2:29446598_EML4:exon:21|-chr2:42553364/1
TAAAATGATTAAAGAAGGTGTGTCTTTAATTGAAGCATGATTTAAAGTAAATGCAAAGCTATGTCGTCCAATCAATGTCCTTACA
ATCCGATCGATTCCTGATTAGTTTGCAGCCATTTGGAATGTCCCCTTTAAATTTAGAAACAG
>1_merged_60_ALK:intron:19|+chr2:29446598_EML4:exon:21|-chr2:42553364/2
GTAAAAGTGGCTAGTTTGAATCAAGATGCACTTTCAAATACATTTGTACACAAGCACTATGATTATACTTCCTGTTTCTAAATTTA
AAGGGGACATTCCAAATGGCTGCAAACTAATCAGGAATCGATCGGATTGTAAGGACATTGATT
>1_merged_60_ALK:intron:19|+chr2:29446598_EML4:exon:21|-chr2:42553364/merged
TAAAATGATTAAAGAAGGTGTGTCTTTAATTGAAGCATGATTTAAAGTAAATGCAAAGCTATGTCGTCCAATCAATGTCCTTACA
ATCCGATCGATTCCTGATTAGTTTGCAGCCATTTGGAATGTCCCCTTTAAATTTAGAAACAGGAAGTATAATCATAGTGCTTGTGTA
CAAATGTATTTGAAAGTGCATCTTGATTCAAACTAGCCACTTTTAC
```

Fig. 6 FusionDirect result example. In the result, an EML4-ALK fusion is detected and reported with three supporting read pairs, while two of them are unique. The reads of each pair are overlapped so they are merged by pair before detection applied

FusionDirect needs a BED file containing four columns (chromosome, start position, end position, gene name). If this file is not provided, FusionDirect will use the built-in BED file, which contains most fusion genes of high clinical importance.

FusionDirect is available at: https://github.com/OpenGene/FusionDirect.jl. It is written in Julia, which is a fresh language allowing high performance technical computing. FusionDirect is built upon the OpenGene Julia library (https://github.com/OpenGene/OpenGene.jl), which provides basic sequence and variant representations and I/O functions of regular NGS-related file formats (i.e., FASTQ/FastA/VCF).

2.5 Deduplication and Unique Supporting Read Counting

When it comes to determine the confidence of a called variant, the most important evidence is the number and quality of its supporting reads. To calculate numbers of supporting reads, we need to identify and collapse duplicated reads.

There exist some tools to remove PCR duplication. Picard MarkDuplicates compares sequences in the five primary positions of both reads and read pairs in a SAM/BAM file. After duplicated reads are marked, this tool differentiates the primary and duplicated reads using an algorithm ranking reads by the summation of their base quality scores. However, this tool can result in unwanted removal of tumor-derived mutated reads, when it shares mapping coordination with some wild-type reads.

Another approach was introduced by CAPP-seq [37]. It collapses those reads with completely identical sequences except the reads with ultralow-quality scores. This method is less lossy since it removes fewer reads comparing with Picard MarkDuplicates. However, it is usually affected by sequencing errors, so the duplication level of processed data can still be very high.

Molecular barcoding sequencing, which has been introduced above, is a new approach that appears to be effective for removing PCR duplication. Since the UID ligation is performed before any amplification happens, the reads derived from the same original DNA will share the same UID. Based on the clustering of UID and read sequence, the PCR duplication can be detected and the consensus read generation process will remove the duplicated reads. Table 4 compares existing deduplication tools.

The methods described above detect duplication before calling variants. An alternative strategy is to detect duplication after variant calling is done, which collapses the reads with same mapping positions (start and end) as a unique read and gives the numbers of reads supporting reference and alternative base for each mutation. This unique read counting method can provide more accurate supporting read calculation. With this strategy applied, we can apply less lossy deduplication methods like CAPP-seq method to keep more information for variant calling. We can even skip deduplication before variant calling if the variant caller is able to handle the data with duplication.

MrBam is a tool designed for such unique read counting task. It differentiates the result reads generated by one single read or multiple reads sharing same mapping coordination. For paired-end sequencing data, it differentiates the cases where mutation is located in read pair's overlapped or non-overlapped region.

Table 4
Feature comparison of existing deduplication tools

	Information loss	Background noise	Error correction
Picard MarkDuplicates	High	Low	None
CAPP-seq	Low	High	None
Molecular barcodes	Low	Low	Yes

MrBam will give numbers of unique reads for a combination of following conditions: supporting reference or alternative, clustered by single or multiple reads, and locating in overlapped or non-overlapped region.

The result of MrBam can be used to filter variants called from ctDNA sequencing data. According to our experience, to report a mutation, we need at least two unique read pairs supporting it, and each pair should either have this mutation in its overlapped region or be a consensus pair generated by multiple pairs. Due to the high ratio sequencing error and extreme depth of ctDNA sequencing data, the mutations only supported by a few single reads at their non-overlapped regions are usually false positive.

MrBam is an open source project. It is developed in Python with its source available at: http://githubs.com/OpenGene/MrBam.

2.6 Methylation Analysis of Cell-Free DNA

Methylation changes are common for different cancer types and usually occur early in cancer development, typically repressing the expression of tumor suppressor genes [38]. Aberrant DNA methylation may offer a more consistent and hence broadly applicable marker of tumor DNA in blood than mutations [39].

There is a very large amount of published information describing DNA methylation patterns in tumor tissue and their impact on patient prognosis. When tumor DNA is shed into the blood stream, these patterns are also detectable in plasma and serum [40].

Tumor-specific ctDNA methylation can be used to quantitate tumor DNA, providing information about the level of tumor burden, as well as revealing the methylation patterns in the tumor. DNA methylation-based biomarkers could be incorporated into patient care and management with only very minor changes to clinical practice, such as recent applications of methylated ctDNA in determining cancer prognosis and in disease monitoring following surgery or during chemotherapy treatment. Methylated ctDNA assays are also developed to meet the stringent criteria required for cancer screening.

Next-generation sequencing platforms allow the construction of genomic maps of DNA methylation at a single-base resolution [41]. Treating genomic DNA with sodium bisulfite deaminates unmethylated cytosine (C) to uracil (U), while methylated C residues remain unaffected [42]. The U eventually converts to thymine (T) in a subsequent polymerase chain reaction (PCR). Whole-genome bisulfite sequencing (WGBS) and reduced representation bisulfite sequencing (RRBS) are two classic methods for genome-wide methylation study.

WGBS (BS-seq; MethylC-seq) theoretically covers all the C information [43]. In this method, genomic DNA is purified and sheared into fragments. The fragmented DNAs are end-repaired; adenine bases are added to the 3′ end (A-tailing) of the DNA fragments, and methylated adapters are ligated to the DNA fragments. The DNA fragments are size-selected before sodium bisulfite treatment and PCR amplification, and the resulting library is sequenced. The major advantage of WGBS is its ability to assess the methylation state of nearly every CpG site, including low CpG-density regions, such as intergenic "gene deserts," partially methylated domains, and distal regulatory elements. It can also determine absolute DNA methylation level and reveal methylation sequence context.

RRBS was developed cheaper than WGBS, which integrates Msp1 restriction enzyme digestion, bisulfite conversion, and next-generation sequencing for the analysis of methylation patterns of specific fragments. A size selection of MspI-digested fragments between 40 and 220 bps was found to cover 85% of CGIs, mostly in promoters, which compose only 1–3% of the mammalian genome, thereby significantly decreasing the amount of sequencing [44]. RRBS-based protocols are more cost-effective than WGBS because these methods focus on the enrichment of CpG-rich regions in close proximity to the restriction enzyme's recognition sequence. However, these protocols may exhibit a lack of coverage at intergenic and distal regulatory elements that are relatively less studied.

Recently, target capturing-based bisulfite sequencing methods have also been developed, and some kits like NimbleGen SeqCap Epi have been commercialized to provide targeted methylation analysis. Since ultra-deep sequencing is usually needed due to low fraction of tumor DNA in cfDNA, the ability of doing target capturing bisulfite sequencing is very important for analyzing methylation information of ctDNA samples.

One of the major applications of ctDNA methylation analysis is to detect early-stage cancers. Circulating methylated SEPT9 DNA in plasma was developed as a biomarker of colorectal cancer [45], and methylation at the SHP-1 promoter 2 (SHP1P2) was reported as a biomarker of non-small cell lung cancer (NSCLC). These biomarkers are usually more sensitive than protein biomarkers (i.e., carcinoembryonic antigen, CEA) and have the potential to be applied in cancer screening or early-stage cancer detection.

Another major application of ctDNA methylation analysis is identifying tissue of origin for carcinoma of unknown primary (CUP). This application is based on the fact that different human tissues and cells have different DNA methylation patterns. Recently, a method of identifying methylation haplotype blocks was developed to perform tumor tissue-of-origin mapping from plasma DNA [46].

The bioinformatics pipeline to analyze bisulfite sequencing (BS-seq) data is different from analyzing normal sequencing data. The key steps of analyzing BS-seq data are quality control, mapping, methylation scoring, differential methylation assessment, etc.

The QA process for BS-seq data is like the same process for normal sequencing data, including quality profiling, adapter trimming, and low-quality reads filtering. However, be aware that bisulfite treatment will result in overrepresentation of T and underrepresentation of C, which may be considered biased by conventional QC tools. Therefore conventional QC tools, like FastQC, are not a good choice to handle quality control for BS-seq data. BseQC [47] and MethyQA [48] are a better choice since they are specialized for BS-seq data.

Mapping BS-seq reads to reference genome is challenging since the sequences do not exactly match the reference, and the library complexity is reduced due to bisulfite treatment [49]. Furthermore, every given T could either be a genuine genomic T or a converted unmethylated C. Due to these reasons, conventional alignment tools such as BWA and Bowtie are unsuitable for mapping BS-seq reads to reference [50]. Some BS-seq specialized aligners have been developed, and typically they can be categorized into two wildcard aligners and three-letter aligners. Wild-card aligners like BSMAP [51] operate by replacing C with Y (IUPAC code for cytosine or thymine), while three-letter aligners like Bismark [52] convert C to T in both sequenced reads and reference.

Once alignment is done, methylation scores can be calculated for cytosines or genomic regions to find differentially methylated cytosines (DMCs) and differentially methylated regions (DMRs). Cytosine methylation scores are calculated by aggregating overlapping reads and calculating the proportion of C or T, which is called β-score. This process can be achieved by tools like Bismark and GBSA [53]. Software like Methylkit [54] provides a strategy of dividing the genome into small bins, and the mean β-score is taken as bin score. Then statistical tests like Fisher's exact test (FET) can be applied to assess the statistical relevance of DMCs/DMRs between samples. This part of work can also be done with Methylkit, which is a comprehensive R package for analyzing DNA methylation (https://code.google.com/p/methylkit).

Recently some novel methylation analysis methods for BS-seq data have been published. For instance, Gao et al. presented a method to search for genomic regions with highly coordinated methylation. This method is based on blocks of tightly coupled CpG sites, which is called methylation haplotype block (MHB). Then methylation analysis can be done in block level (MHL), and the results based on MHL analysis are much better than those based on analyzing single-CpG sites, which means this method can be applied for identifying tissue of origin [46].

Bisulfite sequencing, as the golden method for analyzing DNA methylation, has been studied for many years, and lots of methods and tools have been developed. Due to the urgent needs of establishing methylation analysis for cancer screening and tissue-of-origin identification, BS-seq data analysis will draw more attention of researchers. We cannot discuss all the aspects of BS-seq in this chapter. A collection of BS-seq data analysis tools and pipelines can be found in OMIC tools online (https://omictools.com/bs-seq-category).

2.7 Machine Learning Methods

Machine learning (ML) technologies are very popular for creating data models in lots of domains, and it can also be applied into ctDNA data analysis. Most applicable methods are supervised learning methods, which build classifiers based on training from labeled data. In this subsection, we will show how to use ML technology to build classifiers with ctDNA sequencing data.

One ML application is to classify cfDNA data and non-cfDNA data. CfDNA has certain fragmentation patterns, which can bring nonrandom base content curves of the sequencing data's beginning cycles. The cfDNA fragmentation patterns were first reported by Chandrananda et al. at one nucleotide resolution in 2014 [55]. They found some high frequency 10-nucleotide motifs on either side of cfDNA fragments, and the first two bases of the cfDNA at cleavage site could determine most of the other eight bases. His further study in 2015 indicated that these fragmentation patterns were related to the nonrandom biological cleavage over chromosomes. The ten positions on either side of the DNA cleavage site show consistent patterns with preference of specific nucleotides for nucleosomal cores and linker regions. Figure 7 shows the fragmentation pattern of plasma cfDNA sequencing data.

Since this fragmentation pattern of cfDNA is stable and unique, it can be used to differentiate data of cfDNA and data of other kinds of samples. The authors have developed an open source tool, called CfdnaPattern, to train classifiers like SVM, KNN, or random forest to predict whether a FASTQ is sequenced from cfDNA or not. Cross validation using 0.632+ bootstrapping [56] with more than 3000 FASTQ files gave a result of 99.8% average accuracy, obtained with random forest, linear SVM, or KNN classifiers. This tool is written in Python, with the widely used Python machine learning package scikit-learn. This tool is available at: https://github.com/OpenGene/CfdnaPattern.

Another ML application is to predict whether a mutation is somatic or germline. Typically, tumor and normal samples are both

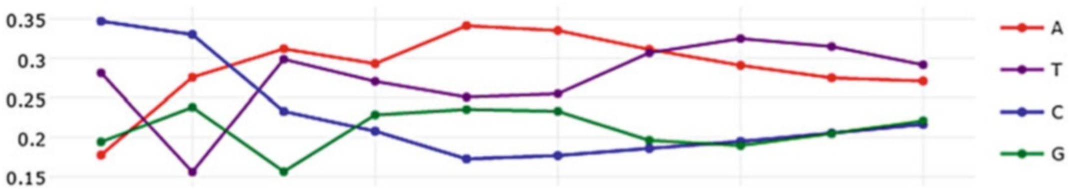


Fig. 7 The cfDNA fragmentation pattern. This figure shows content curves at the first ten cycles of plasma cfDNA sequencing data. This pattern is found stable and can be repeated by different plasma cfDNA samples

sequenced, and the normal sample can be used as a reference to determine the mutations called in tumor sample to be germline or somatic mutations. But for some cases, we may not have matched normal samples for tumor samples, and then we can apply an ML method to classify mutations based on the reads supporting references and the mutations.

DeepSomatic is a tool providing such functions. It can classify somatic and germline mutations with deep neural networks. All reads covering the mutation are extracted and sampled to 256 reads if the read number is greater than 256. Then these reads' bases around the mutation site are coded as a 2D image, with each pixel containing following channels: the read base and its quality score, the reference base, and the lengths of insertion or deletion. Then a deep convolutional neural network (CNN) is constructed with five conventional layers. The model was trained and validated with the tumor-normal paired data, and then cross validation evaluation suggested that this model has an average accuracy higher than 99.9%. DeepSomatic is also an open source tool available at: https://github.com/OpenGene/DeepSomatic.

2.8 Data Simulation

Tuning bioinformatics pipelines and training software parameters require sequencing data with known ground truth, which are actually difficult to get from real sequencing data. Particularly, for ctDNA sequencing applications, which aim to detect low-frequency variations from ultra-deep sequencing data, it is hard to tell whether a called variation is a true positive or a false positive caused by errors from sequencing or other processes. In these cases, simulated data with configured variations can be used to troubleshoot and validate bioinformatics programs.

Although many next-generation sequencing simulators have already been developed, most of them lack of capability to simulate some practical features, such as target capturing sequencing, copy number variations, gene fusions, amplification bias, and sequencing errors. The authors developed SeqMaker, a modern NGS simulator with capability to simulate different kinds of variations, with amplification bias and sequencing errors integrated. Target capturing sequencing is simply supported by using a capturing panel description file, other characteristics like sequencing error rate, average duplication level, DNA template length distribution, and quality distribution can be easily configured with a simple JSON format profile file. With the integration sequencing errors and amplification bias, SeqMaker is able to simulate more real next-generation sequencing data. The configurable variants and capturing regions make SeqMaker very useful to generate data for training bioinformatics pipelines for applications like somatic mutation calling. Table 5 compares the features of SeqMaker and other NGS simulators.

Table 5
A comparison of SeqMaker and other NGS simulators

	SNV	INDEL	INV	TRA	CNV	UMI
SeqMaker	Yes	Yes	Yes	Yes	Yes	Yes
BEAR	No	No	No	No	No	No
dwgsim	Yes	Yes	Yes	Yes	No	No
GemSIM	Yes	No	No	No	No	No
Grinder	Yes	Yes	No	No	No	No
Mason	Yes	Yes	No	No	No	No
pIRS	Yes	Yes	Yes	No	No	No
SInC	Yes	Yes	No	No	Yes	No
wgsim	Yes	Yes	No	No	No	No

SeqMaker is a tool which generates sequencing reads with SNV, INDEL, CNV, and gene fusion enabled, with sequencing error and PCR bias integrated. This tool uses a JSON format profile file to describe the sequencing simulation settings, and a BED format like TSV file to configure the target regions of capturing. First, the simulator samples DNA fragments from whole genome or the target regions configured by the panel file, and CNVs are simulated in this process. Second, the DNA fragments will be altered to simulate SNVs, INDELs, and gene fusions according to the variation list configured in the profile file. Third, a sequencing process will be simulated on each DNA fragment to generate NGS reads, and sequencing errors and amplification bias are also simulated in this process. Finally, generated reads are written into FASTQ files.

SeqMaker is written in Julia, and the source code is available at GitHub: https://github.com/OpenGene/SeqMaker.jl. Currently, it only supports Illumina platforms. More efforts are needed to build simulators for other platforms, especially the new generations of sequencers like PacBio and Nanopore platforms.

3 Discussion

As an innovative method in cancer field, liquid biopsy has current or potential applications in cancer diagnosis, monitoring, and screening. Cell-free tumor DNA, as a major component of liquid biopsy, has been widely used in personalized drug guidance for tumor patients. For those patients not suitable for taking tissue samples by surgery or needle puncture, ctDNA sequencing gives them new opportunities for diagnosis of tumors.

Since ctDNA should be sequenced very deeply, typically target capturing with small gene panels is applied with cost consideration. However, small panels have some disadvantages. Small panels do not allow to detect mutations out of the target regions, difficult to detect large-scale copy number variations, and hard to calculate total mutation burden (TMB) which usually require large panels or whole exome sequencing. As the sequencing cost goes down, it is not difficult to speculate that the whole exome or even whole genome deep sequencing will become affordable and more widely adopted for ctDNA sequencing. Then very big sequencing data will be acquired, and data processing and analysis for such data would be very challenging.

3.1 Conclusion

In this chapter, we introduced the concept and applications of ctDNA, explained the difficulties of analyzing ctDNA NGS data, reviewed some related tools and presented some new methods or tools. One should realize that somatic mutations in cfDNA usually have very low MAF since tumor-specific DNA fragments are usually a small fraction of whole cfDNA. One should be also aware that errors may happen during the experiments and sequencing steps, and software can also introduce artifacts like misalignment or false-positive variant calling.

3.2 Future Work

Although we have discussed so many aspects of bioinformatics for ctDNA NGS data analysis, there still exist topics that have not been discussed above.

Data compression is a key topic we have not discussed in this chapter. Since ctDNA usually requires ultra-deep sequencing, it usually produces very big data. Imagine that if 10,000× WES is applied, we would obtain more than 500 Gb data for a single sample, giving an uncompressed raw file bigger than 1 TB. Storing or transferring such big files will be very challenging, and methods offering high compress ratio will be urgently needed. From signal processing's perspective, the ctDNA sequencing data is highly redundant since it is very deep and has the potential to be compressed with high ratio. However, it is still not easy to compress such kind of data due to three reasons: inconsistent reads due to sequencing errors, varying quality scores, and the requirement of lossless compression. Current methods like DSRC have shown better performance comparing to universal compressors like gzip and bzip2, but the compression ratio improvement is still not satisfactory. Some new compressors like gtz (https://github.com/Genetalks/gtz) have been developed, but they are still not optimized for deep sequencing data. In our opinion, the perfect deep sequencing data compressor should implement local de novo assembly or apply reference-based strategies to achieve much higher compression ratio.

Another topic that remains to be discussed is CNV detection. Since tumor-specific DNA is only a small part of cfDNA, copy number change in tumor cells only leads to slight copy number difference in the ctDNA sequencing data. For instance, if tumor-specific DNA is 1% of the whole cfDNA, and copy number fold in the tumor cells is five, the copy number in whole cfDNA data will be 104%, which is just slightly higher than average level. Current CNV detectors, like CNVkit, are not designed to deal with ctDNA sequencing data and are not sensitive enough to detect such subtle changes in CNV. Better CNV detectors remain to be developed, which should provide better normalization for deep and target-captured ctDNA sequencing data.

Some new methods targeting for cancer immunology are drawing attraction recently. One topic is to predict the outcome of cancer immunotherapies, especially PD-1/PD-L1 checkpoint inhibitors. Tumor mutation burden (TMB) has been shown to be associated with the response of cancer immunotherapies. However, TMB is usually calculated with tissue whole exome sequencing data, and calculating TMB with ctDNA is still challenging due to the low MAF and high level of noises. Methods optimized for ctDNA-based TMB calculation are needed, and this topic can be discussed in future. Another topic related to cancer immunotherapy is neoantigen discovery. In December 2016, Parker Institute for Cancer Immunotherapy and others announced the formation of the Tumor Neoantigen Selection Alliance. This alliance involves researchers from 30 nonprofit institutions and aims to identify software that can best predict neoantigens from patient tumor DNA. For now, computational prediction of neoantigens capable of eliciting efficacious antitumor responses in patients remains a hit-or-miss affair. It is even much more challenging to do the same prediction from patient's ctDNA. The neoantigen prediction study will be a hot topic in both academic and industrial communities, and the progress and outcome can be discussed in the future.

References

1. Kohler CBZ, Radpour R et al (2011) Cell-free DNA in the circulation as a potential cancer biomarker. Anticancer Res 31:2623–2628
2. Diehl F, Schmidt K, Choti MA, Romans K, Goodman S, Li M, Thornton K, Agrawal N, Sokoll L, Szabo SA, Kinzler KW, Vogelstein B, Diaz LA Jr (2008) Circulating mutant DNA to assess tumor dynamics. Nat Med 14 (9):985–990. https://doi.org/10.1038/nm.1789
3. Heitzer E, Ulz P, Geigl JB (2015) Circulating tumor DNA as a liquid biopsy for cancer. Clin Chem 61(1):112–123. https://doi.org/10.1373/clinchem.2014.222679
4. Leon SASB, Sklaroff DM et al (1977) Free DNA in the serum of cancer patients and the effect of therapy. Cancer Res 37:646–650
5. Beaver JA, Jelovac D, Balukrishna S, Cochran RL, Croessmann S, Zabransky DJ, Wong HY, Valda Toro P, Cidado J, Blair BG, Chu D, Burns T, Higgins MJ, Stearns V, Jacobs L, Habibi M, Lange J, Hurley PJ, Lauring J, VanDenBerg DA, Kessler J, Jeter S, Samuels ML, Maar D, Cope L, Cimino-Mathews A, Argani P, Wolff AC, Park BH (2014) Detection of cancer DNA in plasma of patients with early-stage breast cancer. Clin Cancer Res 20 (10):2643–2650. https://doi.org/10.1158/1078-0432.CCR-13-2933

6. Fox EJ, Reid-Bayliss KS, Emond MJ, Loeb LA (2014) Accuracy of next generation sequencing platforms. Next Gener Seq Appl 1. https://doi.org/10.4172/jngsa.1000106
7. Arbeithuber B, Makova KD, Tiemann-Boege I (2016) Artifactual mutations resulting from DNA lesions limit detection levels in ultrasensitive sequencing applications. DNA Res 23 (6):547–559. https://doi.org/10.1093/dnares/dsw038
8. Lixin Chen PL (2017) DNA damage is a pervasive cause of sequencing errors, directly confounding variant identification. Science 355 (6326):752–756
9. Underhill HR, Kitzman JO, Hellwig S, Welker NC, Daza R, Baker DN, Gligorich KM, Rostomily RC, Bronner MP, Shendure J (2016) Fragment length of circulating tumor DNA. PLoS Genet 12(7):e1006162. https://doi.org/10.1371/journal.pgen.1006162
10. Bolger AM, Lohse M, Usadel B (2014) Trimmomatic: a flexible trimmer for Illumina sequence data. Bioinformatics 30 (15):2114–2120. https://doi.org/10.1093/bioinformatics/btu170
11. Chen S, Huang T, Zhou Y, Han Y, Xu M, Gu J (2017) AfterQC: automatic filtering, trimming, error removing and quality control for fastq data. BMC Bioinformatics 18(Suppl 3; 80):91–100. https://doi.org/10.1186/s12859-017-1469-3
12. Langmead B, Salzberg SL (2012) Fast gapped-read alignment with Bowtie 2. Nat Methods 9 (4):357–359. https://doi.org/10.1038/nmeth.1923
13. Li H, Durbin R (2010) Fast and accurate long-read alignment with Burrows-Wheeler transform. Bioinformatics 26(5):589–595. https://doi.org/10.1093/bioinformatics/btp698
14. Li H, Handsaker B, Wysoker A, Fennell T, Ruan J, Homer N, Marth G, Abecasis G, Durbin R, Genome Project Data Processing S (2009) The sequence alignment/map format and SAMtools. Bioinformatics 25 (16):2078–2079. https://doi.org/10.1093/bioinformatics/btp352
15. Tarasov A, Viella AJ, Cuppen E, Nijman IJ, Prins P (2015) Sambamba: fast processing of NGS alignment formats. Bioinformatics. https://doi.org/10.5281/zenodo.13200
16. Mose LE, Wilkerson MD, Hayes DN, Perou CM, Parker JS (2014) ABRA: improved coding indel detection via assembly-based realignment. Bioinformatics 30(19):2813–2815. https://doi.org/10.1093/bioinformatics/btu376
17. Garcia-Alcalde F, Okonechnikov K, Carbonell J, Cruz LM, Gotz S, Tarazona S, Dopazo J, Meyer TF, Conesa A (2012) Qualimap: evaluating next-generation sequencing alignment data. Bioinformatics 28 (20):2678–2679. https://doi.org/10.1093/bioinformatics/bts503
18. McKenna A, Hanna M, Banks E, Sivachenko A, Cibulskis K, Kernytsky A, Garimella K, Altshuler D, Gabriel S, Daly M, DePristo MA (2010) The genome analysis toolkit: a MapReduce framework for analyzing next-generation DNA sequencing data. Genome Res 20 (9):1297–1303. https://doi.org/10.1101/gr.107524.110
19. Cibulskis K, Lawrence MS, Carter SL, Sivachenko A, Jaffe D, Sougnez C, Gabriel S, Meyerson M, Lander ES, Getz G (2013) Sensitive detection of somatic point mutations in impure and heterogeneous cancer samples. Nat Biotechnol 31(3):213–219. https://doi.org/10.1038/nbt.2514
20. Koboldt DC, Zhang Q, Larson DE, Shen D, McLellan MD, Lin L, Miller CA, Mardis ER, Ding L, Wilson RK (2012) VarScan 2: somatic mutation and copy number alteration discovery in cancer by exome sequencing. Genome Res 22(3):568–576. https://doi.org/10.1101/gr.129684.111
21. Wang K, Li M, Hakonarson H (2010) ANNOVAR: functional annotation of genetic variants from high-throughput sequencing data. Nucleic Acids Res 38(16):e164. https://doi.org/10.1093/nar/gkq603
22. Rausch T, Zichner T, Schlattl A, Stutz AM, Benes V, Korbel JO (2012) DELLY: structural variant discovery by integrated paired-end and split-read analysis. Bioinformatics 28(18): i333–i339. https://doi.org/10.1093/bioinformatics/bts378
23. Newman AM, Bratman SV, Stehr H, Lee LJ, Liu CL, Diehn M, Alizadeh AA (2014) FACTERA: a practical method for the discovery of genomic rearrangements at breakpoint resolution. Bioinformatics 30(23):3390–3393. https://doi.org/10.1093/bioinformatics/btu549
24. Wang K, Ma Q, Jiang L, Lai S, Lu X, Hou Y, Wu CI, Ruan J (2016) Ultra-precise detection of mutations by droplet-based amplification of circularized DNA. BMC Genomics 17:214. https://doi.org/10.1186/s12864-016-2480-1
25. Schmieder R, Edwards R (2011) Quality control and preprocessing of metagenomic datasets. Bioinformatics 27(6):863–864. https://doi.org/10.1093/bioinformatics/btr026

26. Cox MP, Peterson DA, Biggs PJ (2010) SolexaQA: at-a-glance quality assessment of Illumina second-generation sequencing data. BMC Bioinformatics 11(1):485. https://doi.org/10.1186/1471-2105-11-485
27. Meldrum C, Doyle MA, Tothill RW (2011) Next-generation sequencing for cancer diagnostics a practical perspective. Clin Biochem Rev 32(4):177–195
28. Tindall KRKT (1988) Fidelity of DNA synthesis by the Thermus aquaticus DNA polymerase. Biochemistry 27:6008–6013
29. Kinde IWJ, Papadopoulos N, Kinzler KW, Vogelstein B (2011) Detection and quantification of rare mutations with. Proc Natl Acad Sci U S A 108(23):9530–9535
30. Liang RH, Mo T, Dong W, Lee GQ, Swenson LC, McCloskey RM, Woods CK, Brumme CJ, Ho CK, Schinkel J, Joy JB, Harrigan PR, Poon AF (2014) Theoretical and experimental assessment of degenerate primer tagging in ultra-deep applications of next-generation sequencing. Nucleic Acids Res 42(12):e98. https://doi.org/10.1093/nar/gku355
31. Hoffmann C, Minkah N, Leipzig J, Wang G, Arens MQ, Tebas P, Bushman FD (2007) DNA bar coding and pyrosequencing to identify rare HIV drug resistance mutations. Nucleic Acids Res 35(13):e91. https://doi.org/10.1093/nar/gkm435
32. Kivioja T, Vaharautio A, Karlsson K, Bonke M, Enge M, Linnarsson S, Taipale J (2011) Counting absolute numbers of molecules using unique molecular identifiers. Nat Methods 9(1):72–74. https://doi.org/10.1038/nmeth.1778
33. Michael W, Schmitta SRK, Salka JJ, Foxa EJ, Hiattb JB, Loeba LA (2012) Detection of ultra-rare mutations by next-generation sequencing. Proc Natl Acad Sci U S A 109:14508–14513
34. Kennedy SR, Schmitt MW, Fox EJ, Kohrn BF, Salk JJ, Ahn EH, Prindle MJ, Kuong KJ, Shen JC, Risques RA, Loeb LA (2014) Detecting ultralow-frequency mutations by Duplex Sequencing. Nat Protoc 9(11):2586–2606. https://doi.org/10.1038/nprot.2014.170
35. Larkin MA, Blackshields G, Brown NP, Chenna R, McGettigan PA, McWilliam H, Valentin F, Wallace IM, Wilm A, Lopez R, Thompson JD, Gibson TJ, Higgins DG (2007) Clustal W and Clustal X version 2.0. Bioinformatics 23(21):2947–2948. https://doi.org/10.1093/bioinformatics/btm404
36. Kirsch A, Mitzenmacher M (2008) Less hashing, same performance: building a better bloom filter. Random Struct Algor 33(2):187–218. https://doi.org/10.1002/rsa.20208
37. Newman AM, Bratman SV, To J, Wynne JF, Eclov NC, Modlin LA, Liu CL, Neal JW, Wakelee HA, Merritt RE, Shrager JB, Loo BW Jr, Alizadeh AA, Diehn M (2014) An ultrasensitive method for quantitating circulating tumor DNA with broad patient coverage. Nat Med 20(5):548–554. https://doi.org/10.1038/nm.3519
38. Jones SBBPA (2011) A decade of exploring the cancer epigenome – biological and translational implications. Nat Rev Cancer 11(10):726–734. https://doi.org/10.1038/nrc3130
39. Warton K, Samimi G (2015) Methylation of cell-free circulating DNA in the diagnosis of cancer. Front Mol Biosci 2:13. https://doi.org/10.3389/fmolb.2015.00013
40. Heyn H, Esteller M (2012) DNA methylation profiling in the clinic: applications and challenges. Nat Rev Genet 13(10):679–692. https://doi.org/10.1038/nrg3270
41. Laird PW (2010) Principles and challenges of genomewide DNA methylation analysis. Nat Rev Genet 11(3):191–203. https://doi.org/10.1038/nrg2732
42. Frommer MML, Millar DS, Collis CM, Watt F, Grigg GW et al (1992) A genomic sequencing protocol that yields a positive display of 5-methylcytosine residues in individual DNA strands. Proc Natl Acad Sci U S A 89(18):27–31
43. Urich MA, Nery JR, Lister R, Schmitz RJ, Ecker JR (2015) MethylC-seq library preparation for base-resolution whole-genome bisulfite sequencing. Nat Protoc 10(3):475–483. https://doi.org/10.1038/nprot.2014.114
44. Gu H, Smith ZD, Bock C, Boyle P, Gnirke A, Meissner A (2011) Preparation of reduced representation bisulfite sequencing libraries for genome-scale DNA methylation profiling. Nat Protoc 6(4):468–481. https://doi.org/10.1038/nprot.2010.190
45. deVos T, Tetzner R, Model F, Weiss G, Schuster M, Distler J, Steiger KV, Grutzmann R, Pilarsky C, Habermann JK, Fleshner PR, Oubre BM, Day R, Sledziewski AZ, Lofton-Day C (2009) Circulating methylated SEPT9 DNA in plasma is a biomarker for colorectal cancer. Clin Chem 55(7):1337–1346. https://doi.org/10.1373/clinchem.2008.115808
46. Guo S, Diep D, Plongthongkum N, Fung HL, Zhang K, Zhang K (2017) Identification of methylation haplotype blocks aids in deconvolution of heterogeneous tissue samples and

tumor tissue-of-origin mapping from plasma DNA. Nat Genet 49(4):635–642. https://doi.org/10.1038/ng.3805

47. Lin X, Sun D, Rodriguez B, Zhao Q, Sun H, Zhang Y, Li W (2013) BSeQC: quality control of bisulfite sequencing experiments. Bioinformatics 29(24):3227–3229. https://doi.org/10.1093/bioinformatics/btt548
48. Sun S, Noviski A, Yu X (2013) MethyQA: a pipeline for bisulfite-treated methylation sequencing quality assessment. BMC Bioinformatics 14:259
49. Krueger F, Kreck B, Franke A, Andrews SR (2012) DNA methylome analysis using short bisulfite sequencing data. Nat Methods 9 (2):145–151
50. Adusumalli S, Mohd Omar MF, Soong R, Benoukraf T (2014) Methodological aspects of whole-genome bisulfite sequencing analysis. Brief Bioinform 16(3):369–379. https://doi.org/10.1093/bib/bbu016
51. Xi Y, Li W (2009) BSMAP: whole genome bisulfite sequence MAPping program. BMC Bioinformatics 10:232. https://doi.org/10.1186/1471-2105-10-232
52. Krueger F, Andrews SR (2011) Bismark: a flexible aligner and methylation caller for Bisulfite-seq applications. Bioinformatics 27 (11):1571–1572. https://doi.org/10.1093/bioinformatics/btr167
53. Benoukraf T, Wongphayak S, Hadi LH, Wu M, Soong R (2013) GBSA: a comprehensive software for analysing whole genome bisulfite sequencing data. Nucleic Acids Res 41(4): e55. https://doi.org/10.1093/nar/gks1281
54. Akalin A, Kormaksson M, Li S, Garrett-Bakelman FE, Figueroa ME, Melnick A, Mason CE (2012) methylKit: a comprehensive R package for the analysis of genome-wide DNA methylation profiles. Genome Biol 13: R87
55. Chandrananda D, Thorne NP, Bahlo M (2015) High-resolution characterization of sequence signatures due to non-random cleavage of cell-free DNA. BMC Med Genet 8:29. https://doi.org/10.1186/s12920-015-0107-z
56. Efron B, Tibshirani R (1997) Improvements on cross-validation: the .632+ bootstrap method. J Am Stat Assoc 92(438):548–560