J Appl Genetics (2016) 57:71–79
DOI 10.1007/s13353-015-0292-7

ANIMAL GENETICS • REVIEW

Review of alignment and SNP calling algorithms for next-generation sequencing data

M. Mielczarek[1] · **J. Szyda**[1]

Received: 6 November 2014 / Revised: 27 February 2015 / Accepted: 15 May 2015 / Published online: 9 June 2015

Abstract Application of the massive parallel sequencing technology has become one of the most important issues in life sciences. Therefore, it was crucial to develop bioinformatics tools for next-generation sequencing (NGS) data processing. Currently, two of the most significant tasks include alignment to a reference genome and detection of single nucleotide polymorphisms (SNPs). In many types of genomic analyses, great numbers of reads need to be mapped to the reference genome; therefore, selection of the aligner is an essential step in NGS pipelines. Two main algorithms—suffix tries and hash tables—have been introduced for this purpose. Suffix array-based aligners are memory-efficient and work faster than hash-based aligners, but they are less accurate. In contrast, hash table algorithms tend to be slower, but more sensitive. SNP and genotype callers may also be divided into two main different approaches: heuristic and probabilistic methods. A variety of software has been subsequently developed over the past several years. In this paper, we briefly review the current development of NGS data processing algorithms and present the available software.

Keywords Alignment · Genotype calling · NGS · SNP calling · Review · Software

Communicated by: Maciej Szydlowski

✉ M. Mielczarek
magda.mielczarek@up.wroc.pl

[1] Biostatistics Group, Department of Genetics, Wroclaw University of Environmental and Life Sciences, Kożuchowska 7, 51-631 Wroclaw, Poland

Introduction

The impact of next-generation sequencing (NGS) on modern biological sciences is second to none. It is a revolution that could be compared with the one that followed Sanger sequencing 40 years ago. Existing NGS technologies provide us with a throughput that is at least 100 times larger than that obtained by classical Sanger sequencing (Mardis 2008; Ansorge 2009; Metzker 2010), and the technologies are still improving. NGS led to a tremendous increase in sequencing speed; therefore, complete data of any species may be obtained within a period of days. Importantly, NGS provides not only cost-effective de novo sequencing, but also resequencing of known genomes and providing whole genome sequence information for multiple individuals representing the same species or even the same population. A major advantage of whole genome sequencing stems from the potential to detect genetic variants which mark individual deviations from the reference genome (Van Tassell et al. 2008; Alkan et al. 2009; Medvedev et al. 2009). These are not only single nucleotide polymorphisms (SNPs), by far the most common type of polymorphisms, but also small insertions/deletions (InDels), as well as larger structural variants (SV), e.g. copy number variations (CNVs). Beside genetics, where its applications are the most common, NGS technology supports other fields of research and industrial applications, such as metagenomics (Qin et al. 2010), DNA methylation studies (Taylor et al. 2007), mRNA expression analysis (Sultan et al. 2008), cancer genomics (Guffanti et al. 2009), medicine (Auffray et al. 2009), animal breeding (Meuwissen and Goddard 2010) and many others (Ruffalo et al. 2011). Therefore, NGS provides a near-complete picture not only of variants, but also of transcriptomes and epigenomes (Pérez-Enciso and Ferretti 2010).

On the other hand, it should be kept in mind that, since NGS represents a throughput technology, it is highly sensitive

to technological errors, consequently making the process of utilising NGS data for research highly dependent on reliable bioinformatics tools (Medvedev et al. 2009; Horner et al. 2010). Technically, two of the major bioinformatics steps in processing sequences generated by NGS include alignment of short sequence reads to a reference genome and detection of SNPs.

The aim of this article is to present a review of alignment and variant calling algorithms underlying the most commonly used software from the open source domain. This comparison is largely based on the authors' own experience with the application of the tools while dealing with NGS data from dairy cattle. Note that, unlike many NGS dedicated reviews which provide comparisons of particular programs, here we focus on describing the methodology.

Alignment to the reference genome

The first computational step in NGS analysis pipelines consists in the alignment of short sequence reads to the existing reference genome or, alternatively, a de novo assembly without a reference. Nowadays, reference genomes are available for many species. In general, alignment to the reference genome starts with indexing, following which the actual alignment process may be initiated. In the first stage, most aligners construct indices for the reference sequence and/or for the short sequenced read data set. The most current aligners, such as Bowtie2 (Langmead and Salzberg 2012) and SOAP3-dp (Luo et al. 2013), index the reference genome instead of sequenced reads, which is advantageous in terms of computational time, since indexing of the reference needs to be conducted only once, while indexing sequence reads needs to be done for each sample separately. Two main indexing algorithms based on suffix/prefix tries or hash tables are incorporated in a vast majority of software tools, but exceptions exist, e.g. the Slider software dedicated to output from the Illumina sequencer (Malhis et al. 2009) is based on merge-sorting of the reference sub-sequences and read sequences.

Indexing algorithms based on suffix/prefix tries

This category of programs may be based on the following concepts: (i) a suffix/prefix trie, (ii) an enhanced suffix array and (iii) the FM-index. The *suffix/prefix trie* is defined as a data structure representing the set of suffixes of a given string of characters, enabling fast matching of the string, which, in our case, represents a sequence read, to the reference genome. However, for very large NGS data sets, this solution is not efficient enough, resulting in a very long computing time (Li and Homer 2010). The *enhanced suffix array* approach is more efficient and enables storing large genomes in computer memory (Abouelhoda et al. 2004). It is composed of a suffix array and some auxiliary arrays. It is adopted e.g. in Segemehl (Hoffmann et al. 2009). The *FM-index* is the method combining the Burrows–Wheeler transform (BWT) algorithm with auxiliary data structures. The main advantage of this concept is that a suffix array created from BWT altered sequences is significantly more efficient than that from primary sequences. The most popular software tools implementing the FM-index algorithm with the original BWT are BWA (Li and Durbin 2009, 2010), Bowtie (Langmead et al. 2009) and Bowtie2. Moreover, SOAP2 (Li et al. 2009c), SOAP3 (Liu et al. 2012) and SOAP3-dp use a BWT modification termed 2-way Burrows–Wheeler transform (2-BWT), which, in contrast to BWT, facilitates searching for a string pattern simultaneously in both directions and, thus, speeds up the alignment process.

Indexing algorithms based on hash tables

The basic algorithm underlying hash table programs is the seed-and-extend method. It divides a query sequence into words called k-mers and the position of each k-mer is kept in the hash table. k-mers are sought in another sequence and their exact matches are termed seeds. Afterwards, seeds are extended and joined without gaps and then their alignment is refined using the Smith–Waterman alignment algorithm. Recently, this basic algorithm has undergone the following improvements: (i) spaced seed, (ii) q-gram filter and multiple seed hits, and (iii) seed extension. A *spaced seed* algorithm assumes that non-exact matching improves alignment sensitivity (Ma et al. 2002) by accounting for the phenomenon of natural genetic variability between the sequenced organism and the reference. In this method, the seed is relatively long. This solution is implemented e.g. in BFAST (Homer et al. 2009), MAQ (Li et al. 2008a), RMAP (Smith et al. 2008, 2009) and SOAP (Li et al. 2008b). *Q-gram filter* and *multiple seed hits* algorithms were incorporated into SSAHA2 (Ning et al. 2001). They require multiple seed matches and allow for gaps within a seed, but, in contrast to spaced seeds, they extend multiple, short seed matches. It is a prevailing technique to speed up alignment of long sequence reads. Both methods mentioned above rely on the fast lookup in a hash table, while the *seed extension* algorithm is based on accelerating the standard Smith–Waterman algorithm by parallelisation of the alignment. Software such as Novoalign (Novocraft Technologies Sdn Bhd), SHRiMP (Rumble et al. 2009) and SHRiMP2 (David et al. 2011) apply the concept.

A complete technical discussion of the above-mentioned algorithms was provided by Li and Homer (2010).

Alignment

Regardless of the indexing method, the actual alignment is performed using either the Smith–Waterman (Smith and Waterman 1981) or the Needle–Wunsch (Needleman and

Wunsch 1970) algorithms and, depending on the software used, a resulting alignment is either gapped or ungapped. Alignment gaps usually result from small-scale genome rearrangements, such as insertions or deletions. Thus, allowing for gaps in the alignment is a preferred feature and most of the aligner software tools provide this option. Moreover, historically, the first alignment programs, such as SHRiMP, performed a single-end mode alignment, which considered only short reads sequenced in one direction and, consequently, aligned only a short part of the sequence at a time. Currently, all of the commonly used alignment programs support a paired-end alignment mode, in which reads sequenced in both the forward and reverse orders are considered simultaneously as a pair, providing a longer piece of sequence to be aligned and, thus, significantly improving the quality of alignment.

Post-alignment processing

In the literature on the subject, a variety of post-alignment data processing methods is usually suggested (Altmann 2012) in order to facilitate further analytical steps.

The most commonly applied post-alignment tasks include e.g. output file format converting, creating reports from the alignment process or removing polymerase chain reaction (PCR) artefacts. Each of them may be carried out using the SAMtools package (Li et al. 2009a). Converting formats with the samtools view tool allows not only for a reduction of the output data set size, but also guarantees downstream compatibility with variant callers, such as UnifiedGenotyper from the Genome Analysis Toolkit package (McKenna et al. 2010) and many other software tools. Aligners usually generate a simple summary describing the alignment process (MOSAIK, Bowtie2, SOAP2). The description typically includes the total number of aligned reads, the number of properly aligned pairs of reads or the number of reads aligned exactly once. Such summary statistics are essential for assessing the overall quality and correctness of alignment. If the summary report is not available, such as in BWA, it is possible to generate it using the samtools flagstat tool. Another element of alignment post-processing is removing PCR duplicates, which represent the same read pairs occurring many times in the raw data and changing the estimate of the true coverage of the reference genome. The samtools rmdup tool is an example tool available for removing those artefacts.

Pre-variant calling processing

In order to obtain reliable results of polymorphic variant calling, additional processing steps specific for SNP calling are recommended before the actual variant detection (Li et al. 2009b; McKenna et al. 2010; Altmann 2012). The most important ones involve alignment artefact correction and sequence quality score recalibration.

Some *alignment artefacts* occur during the alignment step, which may result in mismatching of many bases near the misalignment site. Such mismatches may then be easily mistaken as SNPs. Local realignment tools are, thus, designed to realign reads in the vicinity of an identified alignment artefact in order to minimise the number of mismatching bases. This step is, for example, implemented in GATK (McKenna et al. 2010). In general, a large amount of genomic regions requiring a local realignment is due to the presence of InDels in the individual's genome with respect to the reference genome. Moreover, it is assumed that a sequencer may miscalculate base quality expressed as the Phred score. Tools provided by the GATK package *recalibrate* base quality scores so that they become more accurate estimates of the actual probability of mismatching the reference genome.

Since realignment is computationally demanding, SAMtools uses a different approach based on combining both artefact correction and recalibration steps. The software assigns a base alignment quality (BAQ) score to each base, which is calculated as Phred-scaled probability of the base being misaligned. If a base is aligned to a different reference base in a sub-optimal alignment, its BAQ is low. BAQ estimation is implemented in SAMtools as a default option.

It is noteworthy that not all programs provide both pre-SNP calling processing steps. For example, SOAPsnp (Li et al. 2009b) supplies recalibration of raw sequencing quality scores, but does not support realignment. Atlas-SNP2 (Shen et al. 2010) provides none of those options.

SNP calling

Thus far, genotypes of SNPs were typically identified in microarrays. Recent advances of NGS and complementary analysis programs have provided a new possibility to identify a higher number of variants. While array density is limited to thousands of polymorphisms, sequencing entire genomes potentially allows for the discovery of all existing polymorphisms (in particular, for the genomic regions covered by the reference sequence). Consequently, it allows for identifying not only common, but also rare SNPs, the importance of which for the determination of complex phenotypes has been recently stressed (Handel et al. 2013). Moreover, thanks to the availability of all existing polymorphisms, we no longer need to rely on the linkage disequilibrium or recombination rate in candidate gene mapping, since causal mutations are present in SNP panels originating from NGS. However, for a reliable variant detection, a high depth of coverage is a prerequisite, since a high number of reads aligned to each base is used to differentiate between sequencing errors and true polymorphisms.

Many algorithms and software packages have been dedicated to identify SNPs in NGS data. When only a single genome is analysed, SNP calling and genotype calling are similar, because heterozygous or homozygous non-reference genotypes imply the presence of an SNP. In the simultaneous analysis of multiple samples, an SNP is identified if at least one individual is heterozygous or homozygous for a non-reference allele. Therefore, SNP calling may be defined as the process of identifying sites differing from a reference sequence, while genotype calling refers to the estimation of genotypes (Nielsen et al. 2011). SNP and genotype callers are based on either heuristic or probabilistic methods.

Heuristic methods

Heuristic methods call variants based on multiple information sources associated with the structure and quality of data. Due to their computational demands, they are less commonly used than probabilistic methods. The heuristic approach is partly adopted in VarScan2 (Koboldt et al. 2012). VarScan2 identifies variants using a heuristic method, as well as a statistical test based on the number of mapped reads covering each allele. For variant detection, a heuristic algorithm determines the genotype based on thresholds for coverage (minimum 33), base quality (minimum 20) and variant allele frequency (minimum 0.08). After that, Fisher's exact test of read counts supporting each allele is applied compared to the expected distribution based on sequencing error alone.

Probabilistic methods

Probabilistic methods provide measures of statistical uncertainty for called genotypes, making it possible to monitor the accuracy of genotype calling. Moreover, additional information concerning allele frequencies and linkage disequilibrium patterns may be included in the analysis (Nielsen et al. 2011). Genotype calling is based on genotype likelihood calculations and adopts Bayes' theorem. After pre-processing steps comprising realignment and quality score recalibration, the next step involves likelihood calculation for each possible genotype at each base (a homozygote for the reference allele, a homozygote for the alternative allele or a heterozygote). It is based on quality scores and allele counts from the reads at the SNP site. In the Bayesian framework, the computed likelihood is combined with a prior genotype probability, which leads to a posterior probability of a genotype. As a result, the genotype with the highest posterior probability is chosen. The ratio between the highest and the second highest probabilities may be used as a measure of confidence. A uniform prior probability may be selected as equal for all genotypes; alternatively, a non-uniform prior can be pre-imposed based on additional information, such as dbSNP (SNP database) entries, the reference sequence structure or features provided by the analysed sample.

SAMtools is an example of software incorporating the probabilistic approach. In particular, samtools mpileup collects information stored in input BAM (binary version of the Sequence Alignment Map format) files and computes the likelihood of data given for each possible genotype. Next, bcftools applies the prior probability and performs the actual SNP calling (Li et al. 2009a). Other examples of programs based on the Bayesian principle are GATK, Atlas-SNP2, SOAPsnp and SNVer (Wei et al. 2011).

Single- vs. multi-sample SNP calling

A customary approach is to identify polymorphic variants based on a comparison of individually sequenced genomes with the reference. Although such an approach is ideal to assess individual polymorphisms usually expressed by SNP alleles, InDels and CNVs, it is difficult to draw inferences on a population-wide level and it requires custom-written programs. This is because sites homozygous for a reference allele are not stored on an individual basis, even if they are polymorphic in other sequenced individuals, which prevents the calculation of population-wide allele and genotype frequencies.

Multi-sample calling involves a simultaneous identification of variants in several individuals, but it is much more CPU time- and resource-consuming than individual variant calling, it requires all the samples to be provided at once (which may pose a challenge in large, collaborative sequencing projects) and is not available in all software tools. Multi-sample calling is provided e.g. by SAMtools and GATK. A major advantage of multi-sample calling comprises the fact that priors may be improved based on the information on allele frequency and on verification of whether the obtained genotypes follow the Hardy–Weinberg equilibrium (Nielsen et al. 2011); the latter, however, is only useful if unrelated individuals are considered.

Technical issues

Data formats

There are several data formats characteristic of raw sequences obtained via NGS analysis. FASTQ is the most popular data format for short sequence reads, while reference genomes are stored in the FASTA format. Both formats are accepted as input by all of the above-mentioned alignment programs. The standard output is usually provided in the SAM format or in the BAM format, which is a binary equivalent of SAM. The latter facilitates reduction of the output data set size and acceleration of analysis (Li et al. 2009a). Note that some aligners, such as MAQ or SOAP2, generate software-specific output formats.

SAM or BAM are usually the most widely accepted input formats by variant calling software. The standard output is usually the Variant Call Format (VCF). It comprises, among other things, called variants and their position in the genome. It may also contain genotype and haplotype information for each polymorphic position. VCF was originally used for SNPs and InDels, but it may also store structural variant information (Li et al. 2009a; Danecek et al. 2011).

Computing platforms and operating systems

With the decrease in sequencing costs, data processing costs are rising, while storing and processing very large amounts of data become a challenge.

Data storage is one of the most important and, at the same time, one of the most expensive pieces of NGS analysis infrastructure. A raw data set containing sequence reads from the NGS technology can be very large, especially when large genomes are sequenced with a high coverage rate. It should be kept in mind that the processing of raw sequence reads generates a large number of new data sets, the size of which are often larger than that of the raw data. Furthermore, different NGS operations have different memory requirements. Based on the description of the SOAP package, requirements for the following tools and genomes as large as that of the human include a 500-MB main memory for SOAPsnp, 8 GB for SOAP2 (SOAPaligner) and even 150 GB for SOAPdenovo2 (Luo et al. 2012). Thus, NGS data processing is very computationally demanding and it requires supercomputing infrastructures involving many CPUs. In order to accelerate NGS processing, most aligners, especially modern ones, may be run in the multi-threaded mode, which is often the only option to complete data processing in a reasonable time frame. Since Linux is the most common operating system used on supercomputers, all aligners are dedicated to the Linux command line and a graphical user interface is not available. Some software tools are also available for the Macintosh operating system and only a few for Windows. Technically, a stand-alone program execution is not feasible and all NGS data processing steps listed above need to be executed sequentially through an analysis pipeline, which is typically implemented as a shell script or, alternatively, a ready-to-use pipeline may be adopted, such as ngs_backbone (Blanca et al. 2011), which includes read cleaning, mapping, transcriptome assembly, annotation and SNV calling.

The characteristics of selected software tools for alignment are summarised in Table 1 (suffix array-based) and Table 2 (hash table-based), while variant calling software is recapitulated in Table 3.

Table 1 The characteristics of suffix array-based alignment to the reference genome software

Name	Indexing	Output formats	PE mode	Gapped alignment	Supported platforms	Operating system	Multi-threaded	Licence	Additional information
Bowtie	Genome	SAM	+	–	Illumina, ABI SOLiD	Linux, Macintosh, Windows, Solaris	+	Artistic licence	Alignments containing ambiguous characters (e.g. Ns) are not allowed
Bowtie2	Genome	SAM	+	+	Illumina, 454, Ion Torrent	Linux, Macintosh, Windows	+	GNU GPL	No upper limit on read length; alignments containing ambiguous characters (e.g. Ns) are allowed
BWA	Genome	SAM	+	+	Illumina, 454, Ion Torrent	Linux	+	GNU GPL	Three algorithms are available: BT, SW, MEM
Segemehl	Genome	SAM	+	+	Illumina, 454	Linux	+	Unlicensed (public)	Not limited to a specific read length
SOAP2 (SOAPaligner)	Genome	SOAP	+	+	Illumina	Linux	+	GNU GPL	SOAP to SAM format convertor can be downloaded separately
SOAP3	Genome	Binary/plain text, SAM	+	–	Illumina	Linux	+	GNU GPL	GPU-accelerated version
SOAP3-db	Genome	Plain text, SAM, BAM	+	+	Illumina	Linux	+	GNU GPL	GPU-accelerated version; mismatches, InDels and small gaps are allowed, read length longer than 500 bp is not recommended

SAM Sequence Alignment Map format, *BAM* binary version of SAM, *PE mode* paired-end alignment mode, *GNU GPL* GNU General Public License

Table 2 The characteristics of hash table-based alignment to the reference genome software

Name	Indexing	Output formats	PE mode	Gapped alignment	Supported platforms	Operating system	Multi-threaded	Licence	Additional information
BFAST	Genome	SAM	+	+	Illumina, 454, ABI SOLiD	Linux	+	GNU GPL	Detailed manual is available
MAQ	Reads	MAQ	+	−	Illumina, ABI SOLiD (partial)	Linux	−	GNU GPL	Upper limit on read length 63 bp
Mosaik	Genome	BAM, SAM, BED, ELAND	+	+	Illumina, ABI SOLiD, 454, Helicos Heliscope (experimental)	Linux, Macintosh, Windows, Solaris	+	MIT	Detailed manual is available; FASTA/FASTQ files compressed using gzip are acceptable
mrFAST	Genome	SAM, DIVET	+	+	Illumina	Linux	−	BSD	Designed to map short reads to reference genome with an emphasis on the discovery of structural variation and segmental duplications
mrsFAST	Genome	SAM	+	−	Illumina	Linux	−	BSD	Mismatches are allowed, InDels are not
Novoalign	Genome	SAM, TXT	+	+	Illumina, ABI SOLiD	Linux, Macintosh	−	Free basic version	Advanced options, such as multi-threading, are available with paid license
SSAHA2	Genome	SSAHA2 SAM, CIGAR, others	+	+	Illumina, ABI SOLiD, 454	Linux, Macintosh	−	Free of charge as binaries	Many output formats are supported
SHRiMP	Reads	SHRIMP	−	+	Illumina, ABI SOLiD, 454	Linux, Macintosh	−	Free	End of support in 2014
SHRiMP2	Genome	SAM, SHRIMP	+	+	Illumina, ABI SOLiD, 454	Linux, Macintosh	+	Free	Updated version of the original SHRiMP
RMAP	Reads	BED	+	+	Illumina	Linux, Macintosh	−	GNU GPL	One of the earliest short read aligners, no limitations on read widths or number of mismatches
SOAP	Genome	SOAP	+	−	Illumina	Linux	+	GNU GPL	SOAP to SAM format convertor can be downloaded separately
SMALT	Genome	SAM, BAM, CIGAR, others	+	+	Illumina, ABI SOLiD, 454, Ion Torrent, PacBio	Linux	+	GNU GPL	FASTA/FASTQ files compressed using gzip are acceptable; many output formats are supported

SAM Sequence Alignment Map Format, *BAM* binary version of SAM, *PE mode* paired-end alignment mode, *GNU GPL* GNU General Public License, *MIT* free software license originating at the Massachusetts Institute of Technology, *BSD* Berkeley Software Distribution license

Table 3 The characteristics of SNP calling software

Name	Input formats	Output formats	Realignment	Recalibration	Single-sample	Multi-sample	Called variants	Operating system	Multi-threaded	Licence	Additional information
Atlas-SNP2	BAM	VCF	–	–	+	–	SNPs	Linux	–	BSD	InDels can be called by Atlas-Indel2
GATK (UnifiedGenotyper)	BAM	VCF	+	+	+	+	SNPs, InDels	Linux	+	MIT	Member of the Genome Analysis Toolkit package
SAMtools (samtools mpileup)	BAM	VCF	+	+	+	+	SNPs, InDels	Linux	–	BSD	Member of the package containing various utilities for manipulating the SAM format
SNVer	BAM	VCF	–	–	+	–	SNPs, InDels	Linux, Macintosh, Windows	–	GNU GPL	Statistical tool for calling common and rare variants in the analysis of pooled or individual NGS data
SOAPsnp	SOAP	Text format, GLFv2	–	+	+	–	SNPs	Linux	–	GNU GPL	Member of the Short Oligonucleotide Analysis Package; input file is a result of SOAPaligner
Varscan2	Pileup/mpileup	Text format, VCF	–	–	+	+	SNPs, InDels, CNA	Linux, Macintosh, Windows	–	OSL	Filter removing sequencing- and alignment-related artifacts is applied

BAM Binary version of SAM, *CNA* copy number alterations, *GLFv2* Genome Likelihood Format v2, *OSL* Open Software License, *GNU GPL* GNU General Public License, *MIT* free software license originating at the Massachusetts Institute of Technology, *BSD* Berkeley Software Distribution license

Conclusions

There is a large number of programs available for processing data generated by the next-generation sequencing (NGS) technology (Li and Homer 2010; Nielsen et al. 2011; Pabinger et al. 2014).

NGS tools have been constantly improved over the past few years, with updates of existing programs and, at the same time, new software has been constantly created. Taking the SOAP mapper as an example: (i) the first version relied on a hash table for alignment, (ii) in the SOAP2, the 2-way Burrows–Wheeler transform (2-BWT) algorithm was implemented and (iii) lately, GPU-based SOAP3 and SOAP3-dp have become available. Another popular program, BWA, consists of three different algorithms: BWA-backtrack, BWA-SW and BWA-MEM. The latter is the most current and it has been recently designed for high-quality data. It also has better performance for 70–100-bp Illumina reads than algorithms used previously. Other tools, such as SAMtools or GATK, have also been frequently updated.

As mentioned above, the analysis of NGS data is a fast developing area; thus, recommendations on which software to use may change quickly. Instead, it is important to understand the differences between underlying algorithms and adjust one's own software selection to particular analytical goals, the size of the data set and available computing platforms.

A typical pipeline tested and recommended by the authors consists of the BWA, SAMtools and GATK packages. Alignment to the reference genome is carried out by BWA-MEM, since the program exhibits a good compromise between computing speed and sensitivity expressed as a percentage of aligned reads. The SAMtools package is used to manage SAM/BAM files. Since file sizes are typically very large with several dozens of GB, especially when large genomes (e.g. mammalian, plant) are sequenced with high coverage, it is very important not to store alignment in the SAM format. Instead, BAM, a binary version of SAM, needs to be generated, e.g. using the samtools view tool. It is also worth noting that sorting and indexing BAM files is a prerequisite for further steps of the analysis and it is done by the samtools sort and index tools. Furthermore, polymerase chain reaction (PCR) duplicates need to be removed, e.g. using the samtools rmdup tool, and, finally, a summary of the mapping process is generated by samtools flagstat in order to monitor the quality of alignment before proceeding to further steps of the analysis. A low percentage of aligned reads may suggest that the alignment was done incorrectly, which can be caused by e.g. low sequencing quality of reads or uncompleted reference sequence. The next step comprises pre-variant calling, including: (i) realignment computed by the RealignerTargetCreator and IndelRealigner tools of the GATK package and (ii) recalibration completed by the BaseRecalibrator and PrintReads tools of GATK. The final step of single nucleotide

polymorphism (SNP) identification is composed of using GATK's UnifiedGenotyper caller. Alternatively, for less experienced users, SAMtools' mpileup and bcftools are recommended because of their ease of use. On the other hand, GATK, which is provided with a detailed user manual, is robust towards data artifacts. Nevertheless, both programs are frequently updated, provide detailed log files describing the process of SNP calling and the agreement between their outputs is high.

Compliance with Ethical Standards This study was conducted without external funding sources and did not involve research in animals.

Conflict of interest All authors claim no potential conflicts of interest.

References

Abouelhoda MI, Kurtz S, Ohlebusch E (2004) Replacing suffix trees with enhanced suffix arrays. J Discrete Algorithms 2:53–86

Alkan C, Kidd JM, Marques-Bonet T, Aksay G, Antonacci F, Hormozdiari F, Kitzman JO, Baker C, Malig M, Mutlu O, Sahinalp SC, Gibbs RA, Eichler EE (2009) Personalized copy number and segmental duplication maps using next-generation sequencing. Nat Genet 41:1061–1067

Altmann A1, Weber P, Bader D, Preuss M, Binder EB, Müller-Myhsok B (2012) A beginners guide to SNP calling from high-throughput DNA-sequencing data. Hum Genet 131(10):1541–54

Ansorge WJ (2009) Next-generation DNA sequencing techniques. N Biotechnol 25:195–203

Auffray C, Chen Z, Hood L (2009) Systems medicine: the future of medical genomics and healthcare. Genome Med 1:2

Blanca JM, Pascual L, Ziarsolo P, Nuez F, Cañizares J (2011) ngs_backbone: a pipeline for read cleaning, mapping and SNP calling using next generation sequence. BMC Genomics 12:285

Danecek P, Auton A, Abecasis G, Albers CA, Banks E, DePristo MA, Handsaker RE, Lunter G, Marth GT, Sherry ST, McVean G, Durbin R; 1000 Genomes Project Analysis Group (2011) The variant call format and VCFtools. Bioinformatics 27(15):2156–2158

David M, Dzamba M, Lister D, Ilie L, Brudno M (2011) SHRiMP2: sensitive yet practical SHort Read Mapping. Bioinformatics 27(7): 1011–1012

Guffanti A, Iacono M, Pelucchi P, Kim N, Soldà G, Croft LJ, Taft RJ, Rizzi E, Askarian-Amiri M, Bonnal RJ, Callari M, Mignone F, Pesole G, Bertalot G, Bernardi LR, Albertini A, Lee C, Mattick JS, Zucchi I, De Bellis G (2009) A transcriptional sketch of a primary human breast cancer by 454 deep sequencing. BMC Genomics 10:163–179

Handel AE, Disanto G, Ramagopalan SV (2013) Next-generation sequencing in understanding complex neurological disease. Expert Rev Neurother 13(2):215–227

Hoffmann S, Otto C, Kurtz S, Sharma CM, Khaitovich P, Vogel J, Stadler PF, Hackermüller J (2009) Fast mapping of short sequences with mismatches, insertions and deletions using index structures. PLoS Comput Biol 5(9):e1000502

Homer N, Merriman B, Nelson SF (2009) BFAST: an alignment tool for large scale genome resequencing. PLoS One 4(11):e7767

Horner DS, Pavesi G, Castrignanò T, De Meo PD, Liuni S, Sammeth M, Picardi E, Pesole G (2010) Bioinformatics approaches for genomics and post genomics applications of next-generation sequencing. Brief Bioinform 11:181–197

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

Langmead B, Salzberg S (2012) Fast gapped-read alignment with Bowtie 2. Nat Methods 9:357–359

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

Li H, Durbin R (2009) Fast and accurate short read alignment with Burrows–Wheeler transform. Bioinformatics 25:1754–1760

Li H, Durbin R (2010) Fast and accurate long-read alignment with Burrows–Wheeler transform. Bioinformatics 26(5):589–595

Li H, Homer N (2010) A survey of sequence alignment algorithms for next-generation sequencing. Brief Bioinform 11:473–483

Li H, Ruan J, Durbin R (2008a) Mapping short DNA sequencing reads and calling variants using mapping quality scores. Genome Res 18: 1851–1858

Li R, Li Y, Kristiansen K, Wang J (2008b) SOAP: short oligonucleotide alignment program. Bioinformatics 24:713–714

Li H, Handsaker B, Wysoker A, Fennell T, Ruan J, Homer N, Marth G, Abecasis G, Durbin R; 1000 Genome Project Data Processing Subgroup (2009a) The Sequence Alignment/Map format and SAMtools. Bioinformatics 25:2078–2079

Li R, Li Y, Fang X, Yang H, Wang J, Kristiansen K, Wang J (2009b) SNP detection for massively parallel whole-genome resequencing. Genome Res 19(6):1124–1132

Li R, Yu C, Li Y, Lam TW, Yiu SM, Kristiansen K, Wang J (2009c) SOAP2: an improved ultrafast tool for short read alignment. Bioinformatics 25:1966–1967

Liu C-M, Wong T, Wu E, Luo R, Yiu S-M, Li Y, Wang B, Yu C, Chu X, Zhao K, Li R, Lam T-W (2012) SOAP3: ultra-fast GPU-based parallel alignment tool for short reads. Bioinformatics 28(6):878–879

Luo R, Liu B, Xie Y, Li Z, Huang W, Yuan J, He G, Chen Y, Pan Q, Liu Y, Tang J, Wu G, Zhang H, Shi Y, Liu Y, Yu C, Wang B, Lu Y, Han C, Cheung DW, Yiu SM, Peng S, Xiaoqian Z, Liu G, Liao X, Li Y, Yang H, Wang J, Lam TW, Wang J (2012) SOAPdenovo2: an empirically improved memory-efficient short-read de novo assembler. Gigascience 1:18

Luo R, Wong T, Zhu J, Liu C-M, Zhu X, Wu E, Lee L-K, Lin H, Zhu W, Cheung DW, Ting H-F, Yiu S-M, Peng S, Yu C, Li Y, Li R, Lam T-W (2013) SOAP3-dp: fast, accurate and sensitive GPU-based short read aligner. PLoS One 8(5):e65632

Ma B, Tromp J, Li M (2002) PatternHunter: faster and more sensitive homology search. Bioinformatics 18:440–445

Malhis N, Butterfield YSN, Ester M, Jones SJM (2009) Slider—maximum use of probability information for alignment of short sequence reads and SNP detection. Bioinformatics 25:6–13

Mardis ER (2008) The impact of next-generation sequencing technology on genetics. Trends Genet 24:133–141

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 20:1297–1303

Medvedev P, Stanciu M, Brudno M (2009) Computational methods for discovering structural variation with next-generation sequencing. Nat Methods 6:S13–S20

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

Meuwissen T, Goddard M (2010) Accurate prediction of genetic values for complex traits by whole-genome resequencing. Genetics 185: 623–631

Needleman SB, Wunsch CD (1970) A general method applicable to the search for similarities in the amino acid sequence of two proteins. J Mol Biol 48(3):443–453

Nielsen R, Paul JS, Albrechtsen A, Song YS (2011) Genotype and SNP calling from next-generation sequencing data. Nat Rev Genet 12(6): 443–451

Ning Z, Cox AJ, Mullikin JC (2001) SSAHA: a fast search method for large DNA databases. Genome Res 11(10):1725–1729

Pabinger S, Dander A, Fischer M, Snajder R, Sperk M, Efremova M, Krabichler B, Speicher MR, Zschocke J, Trajanoski Z (2014) A survey of tools for variant analysis of next-generation genome sequencing data. Brief Bioinform 15(2):256–278

Pérez-Enciso M, Ferretti L (2010) Massive parallel sequencing in animal genetics: wherefroms and wheretos. Anim Genet 41(6):561–569

Qin J, Li R, Raes J, Arumugam M, Burgdorf KS, Manichanh C, Nielsen T, Pons N, Levenez F, Yamada T, Mende DR, Li J, Xu J, Li S, Li D, Cao J, Wang B, Liang H, Zheng H, Xie Y, Tap J, Lepage P, Bertalan M, Batto JM, Hansen T, Le Paslier D, Linneberg A, Nielsen HB, Pelletier E, Renault P, Sicheritz-Ponten T, Turner K, Zhu H, Yu C, Li S, Jian M, Zhou Y, Li Y, Zhang X, Li S, Qin N, Yang H, Wang J, Brunak S, Doré J, Guarner F, Kristiansen K, Pedersen O, Parkhill J, Weissenbach J; MetaHIT Consortium, Bork P, Ehrlich SD, Wang J (2010) A human gut microbial gene catalogue established by metagenomic sequencing. Nature 464:59–65

Ruffalo M, LaFramboise T, Koyutürk M (2011) Comparative analysis of algorithms for next-generation sequencing read alignment. Bioinformatics 27(20):2790–2796

Rumble SM, Lacroute P, Dalca AV, Fiume M, Sidow A, Brudno M (2009) SHRiMP: accurate mapping of short color-space reads. PLoS Comput Biol 5:e1000386

Shen Y, Wan Z, Coarfa C, Drabek R, Chen L, Ostrowski EA, Liu Y, Weinstock GM, Wheeler DA, Gibbs RA, Yu F (2010) A SNP discovery method to assess variant allele probability from next-generation resequencing data. Genome Res 20(2):273–280

Smith TF, Waterman MS (1981) Identification of common molecular subsequences. J Mol Biol 147:195–197

Smith AD, Xuan Z, Zhang MQ (2008) Using quality scores and longer reads improves accuracy of Solexa read mapping. BMC Bioinformatics 9:128

Smith AD, Chung WY, Hodges E, Kendall J, Hannon G, Hicks J, Xuan Z, Zhang MQ (2009) Updates to the RMAP short-read mapping software. Bioinformatics 25:2841–2842

Sultan M, Schulz MH, Richard H, Magen A, Klingenhoff A, Scherf M, Seifert M, Borodina T, Soldatov A, Parkhomchuk D, Schmidt D, O'Keeffe S, Haas S, Vingron M, Lehrach H, Yaspo ML (2008) A global view of gene activity and alternative splicing by deep sequencing of the human transcriptome. Science 321:956–960

Taylor KH, Kramer RS, Davis JW, Guo J, Duff DJ, Xu D, Caldwell CW, Shi H (2007) Ultradeep bisulfite sequencing analysis of DNA methylation patterns in multiple gene promoters by 454 sequencing. Cancer Res 67:8511–8518

Van Tassell CP, Smith TP, Matukumalli LK, Taylor JF, Schnabel RD, Lawley CT, Haudenschild CD, Moore SS, Warren WC, Sonstegard TS (2008) SNP discovery and allele frequency estimation by deep sequencing of reduced representation libraries. Nat Methods 5(3):247–252

Wei Z, Wang W, Hu P, Lyon GJ, Hakonarson H (2011) SNVer: a statistical tool for variant calling in analysis of pooled or individual next-generation sequencing data. Nucleic Acids Res 39(19):e132