Bioinformatics Advance Access published October 11, 2012

Tools for mapping high-throughput sequencing data

Nuno A. Fonseca*, Johan Rung, Alvis Brazma, John C. Marioni

EMBL-EBI
European Molecular Biology Laboratory - European Bioinformatics Institute
Cambridge CB10 1SD, UK

Associate Editor: Dr. Jonathan Wren

ABSTRACT

Motivation: A ubiquitous and fundamental step in high-throughput sequencing analysis is the alignment (mapping) of the generated reads to a reference sequence. To accomplish this task numerous software tools have been proposed. Determining the mappers that are most suitable for a specific application is not trivial.

Results: This survey focuses on classifying mappers through a wide number of characteristics. The goal is to allow practitioners to compare the mappers more easily and find those that are most suitable for their specific problem.

Availability: A regularly updated compendium of mappers can be found at `http://wwwdev.ebi.ac.uk/fg/hts_mappers/`.

Contact: nf@ebi.ac.uk

Supplementary information: Supplementary information on this manuscript is available online.

1 INTRODUCTION

In the last decade high throughput sequencing (HTS) has changed the way life sciences research is done. The decreasing costs have made HTS technology more mainstream and it is now exploited in a growing number of biological applications, the so called -seq experiments: DNA-seq (Mardis, 2008); ChIP-seq (Park, 2009); RNA-seq (Wang *et al.*, 2009; Ozsolak and Milos, 2010; Marioni *et al.*, 2008); BS-seq (Meissner *et al.*, 2008; Cokus *et al.*, 2008); as well as numerous other applications, such as investigating the spatial organization of the genome inside the cell nucleus (Lieberman-Aiden *et al.*, 2009). We refer to Metzker (2009) for an overview of sequencing technologies and related applications.

A common feature of all HTS technologies and applications is the generation of relatively short reads (fragments of DNA sequences), which have to be aligned (mapped) to a reference sequence. The primary challenge is to efficiently find the true location of each read from a potentially large quantity of reference data while distinguishing between technical sequencing errors and true genetic variation within the sample.

Presently, more than 60 mappers are available (see Table 1 for a list of mappers and Figure 1 for a time line), most of them proposed after 2008, concurrent with developments in sequencing technologies. Mappers have had to adapt to: i) handle growing quantities of data generated by HTS; ii) exploit technological developments (Li and Durbin, 2010; Langmead and Salzberg, 2012; Weese *et al.*, 2009); and, iii) tackle protocol developments. For instance, paired-end library protocols motivated the development of mappers that exploit read pairing information (Ning *et al.*, 2001; Langmead *et al.*, 2009; Li *et al.*, 2008a). Furthermore, the appearance of novel protocols may result in specific biases (Ondov *et al.*, 2008; Li *et al.*, 2008a; Malhis and Jones, 2010). One consequence of the increasing number of mappers is that making a suitable choice for a specific application is not easy. Resources such as the SeqAnswers forum Wiki pages (Li *et al.*, 2012) have collated information about different mappers, such as the operating system supported and different technologies that the mappers have been designed to handle. However, information about other equally important features/characteristics of the mappers is difficult to find, being still scattered through publications, source code (when available), manuals and other documentation.

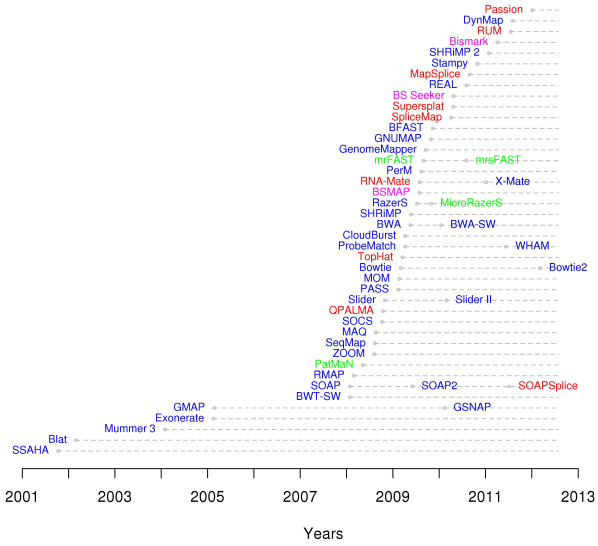

Fig. 1. Mappers time line (since 2001). DNA mappers are plotted in blue, RNA mappers in red, miRNA mappers in green, and bisulfite mappers in purple. Gray dotted lines connect related mappers (extensions or new versions). The time line only includes mappers with peer-reviewed publications and the date corresponds to the earliest date of publication (e.g., advanced publication date as opposed to the date of publication).

*to whom correspondence should be addressed

This survey aims to help overcome these challenges by allowing practitioners to compare mappers more easily and, thus, find those that are most suitable for their specific problem. It does not evaluate mappers in terms of their accuracy, but instead it presents an overview of their characteristics. It complements previous studies that focused mainly on a reduced subset of mappers and/or empirically compared the performance of a small number of mappers (Li and Homer, 2010; Flicek and Birney, 2009; Trapnell and Salzberg, 2009).

2 OVERVIEW

The HTS data mapping problem can be generally stated as follows: given a set of sequences Q (produced by a HTS technology), a set of reference sequences R, a possible set of constraints, and a distance threshold k, find all substrings m of R that respect the constraints and that are within a distance k to a sequence q in Q, i.e., $d(q, m) \leq k$, where $d()$ is some distance function. The occurrences m in R are called *matches*. The constraints imposed can vary depending upon the HTS application and data type (e.g., whether the data generated are single- or paired-end reads).

The main goal of a mapper is to find the true location of each sequence q from a potentially large quantity of reference data while allowing for errors and structural variation. To allow for these errors/variants the matching has to be approximate. The distance measures typically used account for the number of mismatches and indels to allow for errors and structural variation, but they may also incorporate gap sizes or probabilities associated with the reads.

Table 1 provides a brief overview of the mappers considered herein. The information presented in the tables of this manuscript was collected mainly from the publications, manuals and other documentation, by direct inspection of the source code, and in some cases by contacting the developers. When we could not collect information the cells in the table were left blank. We collected, and included in the table, the number of citations of the bibliographic reference (when available) associated with each mapper, and the number of citations normalized by the lifetime (in years) of the publication in an attempt to provide an idea of the popularity of the mappers. However, we emphasize that one cannot infer that mapper A is better than mapper B simply because it is cited more often.

One obvious issue when considering the choice of a mapper is the type of data that it was designed for or is suitable to align (DNA, RNA, miRNA or bisulfite). Another dimension to consider is the sequencing platform that generated the HTS data. General mappers such as BLAT, SSAHA, Exonerate, and Mummer were designed for aligning any sequences (DNA, RNA, or Protein) and the source of data is irrelevant. However, as can be observed in Table 1, a considerable number of mappers support HTS data generated from a subset of technologies. For instance, Slider was designed specifically for Illumina data and exploits the base call probabilities in Illumina's probability output files. By contrast SOCS, RNA-Mate and MapReads are tailored for aligning SOLiD reads, which are encoded in colour space. Some mappers also try to exploit specific biases associated with a sequencing platform. For instance, for the Illumina platform, sequencing accuracy decreases with increasing number of read cycles and therefore less reliable base calls are produced towards the 3'-end of each read. Some mappers like SOAP, Bowtie, or Novoalign can therefore trim several bases off the 3'-end of reads in an attempt to overcome this problem.

Most eukaryotic genes are composed of multiple exons, which can be spliced together in distinct combinations to generate different transcripts. Thus, when RNA-seq reads are mapped to a reference genome, reads that span multiple exons will have potentially large gaps in the alignment corresponding to intronic sequence. The Splicing column, in Table 2, indicates, for the RNA mappers, if the detection of splice junctions is made *de novo* or via user provided libraries of junction locations. *De novo* detection of splice junctions means that the mappers are able to detect splice junctions without relying on existing annotation. An alternative is to build exon junction libraries that include sequences around known or predicted splicing junctions. Some mappers construct these libraries during execution using splice junction information provided by the user while others require that the user provides the library. Finally, hybrid approaches that couple *de novo* with prior information are also possible. For example, QPALMA starts by aligning the reads to the genome to identify putative exons from clusters of mapped reads. Next potential junctions are enumerated within a certain distance around putative exons. Finally, the unmapped reads are aligned against the sequences flanking possible junctions, thus making it possible to find novel junctions. QPALMA, although similar to TopHat, differs by training a support vector machine-like algorithm using known splice junctions from the genome of interest (thus also requiring a set of known junctions from the reference).

3 FEATURE-LEVEL COMPARISON

Table 2 enables a comparison of mappers based on data centric features (e.g., read length limits, utilization of read pairing information, parallel processing), and alignment sensitivity and reporting (e.g., errors allowed, support for gaps, alignments reported, type of alignment performed, and the role of read quality information during alignment).

3.1 Input Data Features

The read length supported by a mapper is a particularly important characteristic. For instance, aligning miRNA data, which typically comprises short reads ranging between 16-30 bases in length after trimming the adapters, requires mappers that support rather short reads. Naturally, the miRNA specific mappers support reads of the mentioned length but some more general purpose mappers, such as Bowtie, BWA, GNUMAP, MapReads, Maq, Novoalign, SHRiMP, Stampy, and SOAP, may also be used for this purpose. By contrast, advances in sequencing technologies have enabled reads longer than 1000 bases to be generated (up to 10000 bases have been reported using PacBio sequencing). The trend of increasing read length has motivated the development of novel mappers (e.g., RazerS, BWA-SW, SOAP2, RUM, RMAP, SOAPSplice, and Bowtie2) that efficiently handle the longer reads.

Sequencing platforms can produce reads in pairs, which can help to detect alignment errors and to improve sensitivity and specificity compared to using single-end reads (Li and Homer, 2010). The majority of the mappers exploit read pairing information (see PE column in Table 2). To align paired reads, a strategy often followed is to independently align the two reads belonging to a pair before

searching for the pair of hits with the correct orientation relationship and proper distance.

HTS data can consist of hundreds of millions of reads. Therefore it is useful when the mappers can (natively) be executed in parallel in distributed-memory (DM) computers (i.e., a cluster composed of multiple computers) or/and using shared-memory (SM) computers (available in computers with modern shared-memory multi-core processors). Given the large number of reads that need to be mapped, it is not surprising that the majority of the methods have been developed to exploit multiple processors/computers to speed up the mapping. A different approach is followed by CloudBurst (Schatz, 2009), a cloud aware mapper designed to run in computer clouds and in local computer clusters. Implementation details for each mapper are provided in the supplementary file.

The quality aware column in Table 2 indicates whether a mapper exploits the base quality scores (generated by the sequencer) during the alignment. It has been shown (Li and Homer, 2010) that using accurate quality scores can reduce alignment errors by giving a lower penalty for a mismatch in a position with a low quality score. Several mappers, such as Bowtie, BWA, GEM-Mapper, PASS, SHRiMP2, ZOOM, SOCS, RMAP, and GNUMAP exploit quality scores during the alignment but differ in the way that they do it. For instance, RMAP does not penalize mismatches for bases with a quality score below a predetermined cut-off value, while Novoalign uses base qualities to calculate base penalties for the Needleman-Wunsch algorithm. GNUMAP goes a step further and uses the Solexa/Illumina probability output files to construct a position weight matrix (PWM) for each read, before a modified Needleman-Wunsch alignment algorithm exploits these matrices to score and align a read against the reference sequence.

3.2 Variation and Errors

To cope with errors and variation, mappers must allow the matching of the reads to the reference sequences to be approximate. For instance, in a project to detect genome variation, the mapper should allow a small number of errors but enough to cope with the expected variation. However, more errors should be allowed when aligning reads against reference sequences from different species or when longer reads are used, e.g., 5 mismatches in a read with 36 bases (14%) is quantitatively different from 5 mismatches in a read with 150 bases (3%). The mismatches and indels columns in Table 2 indicate whether a mapper aligns reads while allowing for errors (mutations and short indels), while the Gaps column indicates whether consecutive insertions or deletions are allowed during alignment. The challenge is to distinguish between nucleotide variation caused by true genetic variation and differences from the reference due to inaccuracy in sequencing.

In an attempt to improve computational efficiency many mappers impose constraints on the number of mismatches/gaps allowed. Some mappers allow a small number of mismatches (for instance, ELAND supports up to 2 mismatches, VMATCH and WHAM support up to 5 mismatches, and BSMAP supports up to 15 mismatches), while others accept an arbitrary number of mismatches and no indels (e.g., MapReads, MicroRazerS, and mrsFast). Some mappers support indels but, again, often with some constraints: e.g., SOAP and SOAP2 support up to 3 and 2 indels respectively, MrFast up to 6 indels, and BWA up to 8 indels. Finally, some mappers impose no constraint on the number of mismatches and indels (e.g., Bowtie, Bowtie2, GNUMAP, Mosaik, RazerS, SSAHA2, VMATCH, SHRiMP and SHRiMP2). In this final case, a threshold on the score function value is often used to determine whether a read is mapped to a particular location.

Support for gaps (long indels) comes at the cost of computational efficiency but is a feature required in several contexts, namely to map longer reads (since they have a greater probability of containing gaps) or to map RNA-seq data. Li and Homer (2010) showed that gapped alignment increases the percentage of reads mapped but that it did not reduce the percentage of reads incorrectly aligned. In an attempt to minimize the computational cost incurred from allowing gaps, many mappers impose constraints: e.g., GMAP, GSNAP, SOAPSplice, SpliceMap, and WHAM allow for a single gap with different gap size constraints; SOAP2 and QPALMA allow a single gap with no constraint on size; and BLAT allows multiple gaps with a maximum size of 23k bases.

The alignment phase in RNA-seq experiments presents many challenges that arise, in general, from splicing events. These can be handled easily if reads are mapped to a pre-defined transcriptome at the cost of missing novel transcripts. Alternatively, the reads can be mapped to the genome. However, reads that span multiple exons will have potentially large gaps in the alignment corresponding to intronic sequence. Hence, RNA-seq mappers should be able to support large gaps, as is the case for MapSplice, TopHat, Supersplat, SoapSplice, SpliceMap, RNA-mate, RUM, PASS, QPALMA or MapSplice. These mappers are also termed as spliced aligners (Garber *et al.*, 2011) due to their ability to align a read to multiple exons. Some of these mappers are, in fact, wrappers to other mappers (e.g., TopHat uses Bowtie; RUM uses Bowtie or Blat; RNA-Mate uses MapReads; and SpliceMap can use Bowtie, Eland or SeqMap). Some wrappers (e.g., MapSplice or SpliceMap) use an exon-first approach that involves two main steps: i) map reads to the genome using unspliced read aligners; ii) unmapped reads are then split into shorter segments and aligned independently. The genomic regions surrounding the mapped read segments are then searched for spliced connections. This approach is efficient since a smaller proportion of the reads are used in the more computationally demanding second step. Alternative approaches are seed-and-extend variations, as exemplified by TopHat when used to align RNA-seq reads to a genome. Briefly, TopHat starts by mapping the reads using Bowtie against the whole genome, then aggregates the reads into islands of candidate exons, before generating potential donor/acceptor splice sites using neighbouring exons. Finally, unmapped reads are mapped (using Bowtie) to these splice junction sequences.

3.3 Alignments

The task at hand will determine whether the exact alignments or locations are of interest. Mappers can report (semi-) global or local alignments with respect to the reads (see Alignment column in Table 2). A mapper performs a (semi-) global (or end-to-end) alignment with respect to the reads when it produces an alignment that involves all of the bases in the read. A local alignment considers only bases in part of the read (bases at the ends of the read are usually omitted in the alignment). Local alignment of the reads is often faster than global (or end-to-end) alignment since the mappers can stop the alignment process when a good quality unique match

Mapper	Data	Seq.Plat.	Input	Output	Avail.	Version	Cit.	$\frac{Citations}{Years}$	Reference
BFAST	DNA	I,So,4, Hel	(C)FAST(A/Q)	SAM TSV	OS	0.7.0	94	37.11	Homer *et al.* (2009)
Bismark	Bisulfite	I	FASTA/Q	SAM	OS	0.7.3	7	6.21	Krueger and Andrews (2011)
Blat	DNA	N	FASTA	TSV BLAST	OS	34	2844	275.67	Kent (2002)
Bowtie	DNA	I,So,4,Sa,P	(C)FAST(A/Q)	SAM TSV	OS	0.12.7	1168	363.42	Langmead *et al.* (2009)
Bowtie2	DNA	I,4,Ion	FASTA/Q	SAM TSV	OS	2.0beta5		0.00	Langmead and Salzberg (2012)
BS Seeker	Bisulfite	I	FASTA/Q	SAM	OS		19	9.26	Chen *et al.* (2010)
BSMAP	Bisulfite	I	FASTA/Q	SAM TSV	OS	2.43	31	11.06	Xi and Li (2009)
BWA	DNA	I,So,4,Sa,P	FASTA/Q	SAM	OS	0.6.2	738	224.20	Li and Durbin (2009)
BWA-SW	DNA	I,So,4,Sa,P	FASTA/Q	SAM	OS	0.6.2	160	67.69	Li and Durbin (2010)
BWT-SW	DNA	N	FASTA	TSV	OS	20070916	45	10.42	Lam *et al.* (2008)
CloudBurst	DNA	N	FASTA	TSV	OS	1.1	146	46.97	Schatz (2009)
DynMap	DNA	N	FASTA	TSV	OS	0.0.20		0.00	Flouri *et al.* (2011)
ELAND	DNA	I	FASTA	TSV	Com	2	7	1.09	Unpublished[1]
Exonerate	DNA	N	FASTA	TSV	OS	2.2	255	34.69	Slater and Birney (2005)
GEM	DNA	I, So	FASTA/Q	SAM, Counts	Bin	1.x	4	1.35	Unpublished[2]
GenomeMapper	DNA	I	FASTA/Q	BED TSV	OS	0.4.3	31	11.66	Schneeberger *et al.* (2009)
GMAP	DNA	I,4,Sa,Hel,Ion,P	FASTA/Q	SAM, GFF	OS	2012-04-27	217	29.52	Wu and Watanabe (2005)
GNUMAP	DNA	I	FASTA/Q Illumina	SAM TSV	OS	3.0.2	15	5.73	Clement *et al.* (2010)
GSNAP	DNA	I,4,Sa,Hel,Ion,P	FASTA/Q	SAM	OS	2012-04-27	72	31.61	Wu and Nacu (2010)
MapReads	DNA	So	FASTA/Q	TSV	OS	2.4.1		0.00	Unpublished[3]
MapSplice	RNA	I	FASTA/Q	SAM BED	OS	1.15.2	50	28.17	Wang *et al.* (2010)
MAQ	DNA	I,So	(C)FAST(A/Q)	TSV	OS	0.7.1	957	251.66	Li *et al.* (2008a)
MicroRazerS	miRNA	N	FASTA	SAM TSV	OS	0.1	7	2.75	Emde *et al.* (2010)
MOM	DNA	I,4	FASTA	TSV	Bin	0.6	18	5.55	Eaves and Gao (2009)
MOSAIK	DNA	I,So,4,Sa,Hel,Ion,P	(C)FAST(A/Q)	BAM	OS	2.1	4	1.18	Unpublished[4]
mrFAST	miRNA	I	FASTA/Q	SAM	OS	2.1.0.4	158	58.34	Alkan *et al.* (2009)
mrsFAST	miRNA	I,So	FASTA/Q	SAM	OS	2.3.0	32	18.03	Hach *et al.* (2010)
Mummer 3	DNA	N	FASTA	TSV	OS	3.23	683	81.58	Kurtz *et al.* (2004)
Novoalign	DNA	I,So,4,Ion,P	(C)FAST(A/Q) Illumina	SAM TSV	Bin	V2.08.01	137	34.49	Unpublished[5]
PASS	DNA	I,So,4	(C)FAST(A/Q)	SAM GFF3 BLAST	Bin	1.62	45	13.67	Campagna *et al.* (2009)
Passion	RNA	I,4,Sa,P	FASTA/Q	BED	OS	1.2.0		0.00	Zhang *et al.* (2012)
PatMaN	miRNA	N	FASTA	TSV	OS	1.2.2	38	9.36	Prüfer *et al.* (2008)
PerM	DNA	I,So	(C)FAST(A/Q)	SAM TSV	OS	0.4.0	30	10.88	Chen *et al.* (2009)
ProbeMatch	DNA	I,4,Sa	FASTA	ELAND	OS		6	1.92	Kim *et al.* (2009)
QPALMA	RNA	I,4	Specific	TSV	OS	0.9.2	75	21.11	De Bona *et al.* (2008)
RazerS	DNA	I,4	FASTQ	TSV ELAND	OS	1.1	58	20.17	Weese *et al.* (2009)
REAL	DNA	I	FASTA/Q	TSV	OS	0.0.28		0.00	Frousios *et al.* (2010)
RMAP	DNA	I,So,4	(C)FAST(A/Q)	BED	OS	2.05	162	38.27	Smith *et al.* (2008)
RNA-Mate	RNA	So	CFASTA	BED Counts	OS	1.1	28	10.04	Cloonan *et al.* (2009)
RUM	RNA	I,4	FASTA/Q	SAM TSV BED	OS	1.11	2	2.36	Grant *et al.* (2011)
SeqMap	DNA	I	FASTA	ELAND	OS	1.013	142	37.34	Jiang and Wong (2008)
SHRiMP	DNA	I,So,4,Hel	(C)FAST(A/Q)	TSV	OS	1.3.2	155	50.91	Rumble *et al.* (2009)
SHRiMP 2	DNA	I,So,4	FASTA/Q	SAM	OS	2.2.2	15	11.76	David *et al.* (2011)
Slider	DNA	I	Illumina	TSV	OS	0.6	39	10.98	Malhis *et al.* (2009)
Slider II	DNA	I	Illumina	TSV	OS	1.1	16	7.25	Malhis and Jones (2010)
Smalt	DNA	I,4,Sa,Ion,P	FASTA/Q	SAM	OS	0.6.1		0.00	Unpublished[6]
SOAP	DNA	I	FASTA/Q	TSV	OS	1.11	451	104.41	Li *et al.* (2008b)
SOAP2	DNA	I	FASTA/Q	SAM TSV	OS	2.21	294	99.38	Li *et al.* (2009b)
SOAPSplice	RNA	I,4	FASTA/Q	TSV	Bin	1.8	3	3.54	Huang *et al.* (2011a)
SOCS	DNA	So	(C)FAST(A/Q)	TSV	OS	2.1.1	49	14.15	Ondov *et al.* (2008)
SpliceMap	RNA	I	FASTA/Q	SAM BED	OS	3.3.5.2	63	29.80	Au *et al.* (2010)
SSAHA	DNA	N	FASTA/Q	TSV	OS	3.1	483	42.29	Ning *et al.* (2001)
SSAHA2	DNA	I,4,Sa	FASTA/Q	SAM	Bin	2.5.5	483	44.99	Ning *et al.* (2001)
Stampy	DNA	I	FASTA/Q	SAM TSV	Bin	1.0.16	26	16.19	Lunter and Goodson (2011)
Supersplat	RNA	N	FASTA	TSV	OS	1.0	21	9.93	Bryant Jr *et al.* (2010)
TopHat	RNA	I	FASTA/Q, GFF	BAM	OS	1.4.1	389	121.04	Trapnell *et al.* (2009)
VMATCH	DNA	N	FASTA	TSV	Bin		26	2.75	Unpublished[7]
WHAM	DNA	N	FASTQ	SAM	OS	0.1.4	3	3.33	Li *et al.* (2011)
X-Mate	DNA	I,So,4	(C)FAST(A/Q)	SAM BED Counts	OS	1	1	0.74	Wood *et al.* (2011)
ZOOM	DNA	I,So,4	(C)FAST(A/Q)	SAM BED GFF	Com	1.5	109	28.66	Lin *et al.* (2008)

Table 1. List of mappers. The Data column indicates if the mapper is tailored for DNA, RNA, miRNA, or bisulfite sequences. The Seq.Plat. column indicates if the mapper natively supports reads from a specific sequencing platform (**I**llumina, ABI **So**lid, Roche **4**54, ABI **Sa**nger, **Hel**icos, **Ion** torrent, and **P**acbio) or not (**N**). The mappers are available as open-source (**OS**), in binary form (**BIN**) or commercially (**Com**). The input and output columns indicate, respectively, the file formats accepted and produced by the mappers. Input formats: FASTA, FASTQ, CFASTA, CFASTQ, and Illumina Sequence and Probability files format. Output formats: SAM (Li *et al.*, 2009a), Tab-Separated-Values (TSV), BED file format, different versions of General Feature Format (GFF), number of reads mapped to genes/exons (Counts). The version column indicates the version of the mapper considered in this study. The table also includes the number of citations per year of the associated publication. The number of citations (**Cit.** was obtained from Google Scholar on 14 Apr 2012.

Unpublished[1] ELAND: Efficient local alignment of nucleotide data.

Unpublished[2] MapReads: SOLiD System Color Space Mapping Tool.

Unpublished[3] The Vmatch large scale sequence analysis software.

Unpublished[4] Mosaik 1.0 documentation.

Unpublished[5] `www.novocraft.com`.

Unpublished[6] SMALT Manual

Unpublished[7] GEM-GEnomic Multi-tool.

Mapper	Min. RL	Max. RL	Mismatches	Indels	Gaps	Align. Reported	Alignment	Parallel	QA	PE	Splicing	Data
BFAST		*	Y	Y	Y	B,R,U	G	SM	N	Y	N	DNA
Bismark	16	10K	Score	Score	N	U	-	SM	Y	Y	N	Bisulfite
Blat	11	5000K	Score	Score	Y	B	L	N	N	N	De novo	DNA
Bowtie	4	1K	Score	Score	N	A,B,R,S	G L	SM	Y	Y	N	DNA
Bowtie2	4	5000K	Score	Score	Y	A,B,R,S	G L	SM	Y	Y	N	DNA
BS Seeker	-	-	3	0	N	U	-	SM	Y	N	N	Bisulfite
BSMAP	8	144	15	0	N	B,S,U		SM	N	Y	N	Bisulfite
BWA	4	200	Y	8	Y	R,S	G	SM	Y	Y	N	DNA
BWA-SW	4	1000K	0.1	0.1	Y	R,S	L	SM	Y	N	N	DNA
BWT-SW		1K	Score	Score	Y	A		N	N	N	N	DNA
CloudBurst		1K	Y	Y	Y	A,B	G	Cloud	N	N	N	DNA
DynMap	18	8K	5	0	N	B	L	N	N	N	N	DNA
ELAND		32	2	0	N	B		N	N	N	N	DNA
Exonerate	20	*	Score	Score	Y	B,S	G L	N	N	N	De novo	DNA
GEM	0	4294M	1.0	1.0	Y	A, S	G	SM	Y	Y	Lib and de novo	DNA
GenomeMapper	12	2K	10	10	Y	A,B,R	G	SM	N	N	N	DNA
GMAP	8	*	Y	Y	Y	B	G L	SM	N	N	De novo	DNA
GNUMAP	16	1K	Score	Score	Y	B	G	SM/DM	Y	N	N	DNA
GSNAP	8	250	Y	Y	Y	A,B,U,S	G L	SM	N	Y	Lib and de novo	DNA
MapReads	10	120	Score	0	N	S		N	Y	N	N	DNA
MapSplice	-	-	3		Y	B	-	SM	N	Y	De novo	RNA
MAQ	8	63	Y	Y	N			N	Y	Y	N	DNA
MicroRazerS	10	*	Score	0	N	S	G	N	N	N	N	miRNA
MOM			Y	0	N	A	L	SM	N	Y	N	DNA
MOSAIK	15	1000	Y	Y	Y	A,B	G	SM	Y	Y	N	DNA
mrFAST	25	300	Score	6	N	A,B	G	N	N	Y	N	miRNA
mrsFAST	25	200	Y	0	N	A	G	N	N	Y	N	miRNA
Mummer 3	10	*	Y	Y	Y	A,B	G	N	N	N	N	DNA
Novoalign	30	300	8	2	N	A, B, R, U, S	G	SM/DM/Cloud	Y	Y	Lib	DNA
PASS	23	1K	Y	Y	Y	A,B	G	SM	Y	Y	De novo	DNA
Passion	-	-	Y	Y	Y	U	-	SM	Y	Y	De novo	RNA
PatMaN	1	*	Y	Y	N	A	G	N	N	N	N	miRNA
PerM	20	128	9	0	Y	A,U	G	DM	Y	Y	N	DNA
ProbeMatch	36	50	3	Y	N	A,B		N	N	N	N	DNA
QPALMA	-	-	Y	Y	Y	B	L	N	Y	N	Lib and de novo	RNA
RazerS	11	*	Score	Score	Y	A,B,S	G	N	N	Y	N	DNA
REAL	4	*	Score	N	N	B, U	G	SM	Y	N	N	DNA
RMAP	11	10K	Y	0	N	B,S		N	Y	Y	N	DNA
RNA-Mate	-	-	Y	0	N	S	-	DM	Y	N	Lib	RNA
RUM	-	-	Y	Y	Y	B	-	SM	N	Y	De novo	RNA
SeqMap	15	500	5	3	N	A		SM	N	N	N	DNA
SHRiMP	14	1K	Score	Score	Y	B,S	G	SM	N	Y	N	DNA
SHRiMP 2	30	1K	Y	Score	N	B,U,S	G	SM	Y	Y	N	DNA
Slider		62	3	0	N	B,S		N	Y	Y	N	DNA
Slider II		93	Y		N	B,S		N	N	Y	N	DNA
Smalt	4	2048M	Score	Score	N	A,B,R,U,S	L	SM	Y	Y	N	DNA
SOAP	7	60	5	3	N	B,R S		SM	N	Y	N	DNA
SOAP2	27	1K	2	0	Y	A,B,R	L	SM	N	Y	N	DNA
SOAPSplice	13	3K	5	2	Y	U	-	SM	Y	Y	De novo	RNA
SOCS		64	Y	0	N	A,B		SM	Y	N	N	DNA
SpliceMap	-	-	0.1		Y	A	-	SM	N	Y	Lib and/or de novo	RNA
SSAHA	15	*	Y	Y	Y	B,S	G L	N	N	N	N	DNA
SSAHA2	15	48K	Score	Score	N	B,S	L	N	N	Y	N	DNA
Stampy	4	4K	0.15	30	N	B,R,S	G	N	Y	Y	N	DNA
Supersplat			0	0	Y	A,U	G	N	N	N	De novo	RNA
TopHat	-	-	2	0	N	B,S	-	SM	Y	Y	De novo	RNA
VMATCH			Score	Score	Y	A,B,S	G L	N	N	N	N	DNA
WHAM	5	128	5	3	N	A,B,R,U,S	G	N	Y	Y	De novo	DNA
X-Mate	-	-	Y	0	N	S	-	DM	Y	N	Lib	DNA
ZOOM	12	240	Y	Y	N	B,S,U	G	SM/DM	Y	Y	N	DNA

Table 2. Features comparison. Read length limits are shown in the first two data columns: minimum read length (Min. RL) and maximum read length (Max. RL.). Unless otherwise stated the unit is base pairs, K denotes kilobases (1000 bases), M denotes megabases (1000K bases), and * denotes a (unknown) large number. The support for mismatches and short indels is presented in the 4th and 5th columns respectively, including when possible the maximum number of allowed mismatches and indels: by default the value is in bases; in some cases the value is presented as a proportion of the read size; or as **score**, meaning that mapper uses a score function. The **Gaps** column indicates whether consecutive insertions or deletions are allowed during alignment. The alignments reported column indicates the alignments reported when a read maps to multiple locations: **A**-all, **B**-best, **R**-random, **U**-unique alignments only (no multimaps), and **S**-user defined number of matches. The alignment column indicates if the reads are aligned end-to-end (**G**lobally) or not (**L**ocally). The Parallel column indicates if the mapper can be run in parallel and, if yes, how: using a shared-memory (**SM**) or/and a distributed memory (**DM**) computer. The QA (quality awareness) column indicates if the mapper uses read quality information during the mapping. The support for paired reads is indicated in the PE column. The Splicing column indicates, for the RNA mappers, if the detection of splice junctions is made *de novo* or through user provided libraries (*Lib*). Yes is abbreviated as **Y** and No is abbreviated as **N**. A cell in the table is filled with '-' when a third-party mapper is used to perform the alignment.

is found. This is useful for cases where the number of hits are of interest, as opposed to the alignments *per se*.

Multimap reads (also known as multireads) are those that align to multiple locations with very similar alignment scores, due to the reads originating from repetitive regions and/or due to the short length of the reads. Having identified a multimap read a mapper has several reporting options. For instance, in RNA-seq data analysis when the reference is a transcriptome, one may want to consider reads with *many* possible alignments (and then perform some post-processing). TopHat, on the other hand, uses Bowtie to map and report reads with up to 10 possible locations and excludes the reads that have more than this number of alignments: the aim is to include multimap reads from paralogous genes but to exclude reads aligned to low-complexity sequences. Although several mappers have an option to report all possible mapping locations of a read, they are less efficient than mappers specifically designed for this purpose, such as mrFast, mrsFAST, and PatMaN.

4 EXECUTION TIME AND MEMORY REQUIREMENTS

The computational time required by a mapper to align a given set of sequences and the computer memory required are critical characteristics. If a mapper is extremely fast but the computer hardware available for performing a given analysis does not have enough memory to run it, then the mapper is not very useful. A mapper is also not useful if it has a very low memory requirement but is very slow. Hence, ideally, a mapper should be able to balance speed and memory usage whilst reporting the desired mappings.

We measured the computational speed and memory requirements of the mappers empirically. The human genome (Homo sapiens, Assembly GRCh37), obtained from Ensembl (build 66), was used as a reference. Two further reference sets (subsets of chromosomes of the human genome, with a size of 130MB and 1GB) were used to assess the impact of reference size on computational speed and memory usage. Samples of one million high-quality reads were mapped against each reference. Table 3 presents the total time (in minutes), maximum memory usage, time spent in pre-processing/preparing the data (which includes indexing), the time spent on mapping, and the number of reads aligned. Due to space constraints we only show, for each type of data, the five mappers with lowest mapping time, total time, memory usage, or highest number of reads mapped when aligning to the whole human genome. It should be stressed that the focus of the evaluation is on the speed and memory of the mappers - the accuracy of the mappings produced was not evaluated. All the values presented are averaged across multiple runs. The default parameter values of the mappers were used whenever possible and no parameter optimization was attempted. The mappers were configured to use a single processor and up to a maximum of 32 GB of RAM. Further information about the data, methods, and more results are presented in Section 2 in the Supplementary file.

Mapper	Time	Pre.Time	Map.Time	Mem	R.Aligned
BS (30 bp)					
Bismark	188 $\pm$13	164 $\pm$12	23 $\pm$1	10.2	713,938
BS_Seeker	1,151 $\pm$110	1,137 $\pm$110	14 $\pm$1	26	71,050
BSMAP	9 $\pm$1	0 $\pm$0	9 $\pm$1	8.2	855,086
GSNAP	477 $\pm$83	26 $\pm$1	451 $\pm$82	15	998,005
Novoalign	45 $\pm$8	17 $\pm$2	28 $\pm$7	7.8	531,944
RMAP	158 $\pm$14	0 $\pm$0	158 $\pm$14	3.3	691,414
DNA (100 bp)					
BFAST	39 $\pm$0	20 $\pm$0	20 $\pm$0	21.4	561,348
Blat	93 $\pm$7	2 $\pm$0	90 $\pm$7	3.8	950,220
Bowtie	169 $\pm$39	166 $\pm$38	3 $\pm$1	5	798,566
Bowtie2	176 $\pm$40	168 $\pm$39	8 $\pm$1	5.1	991,880
BSMAP	25 $\pm$5	0 $\pm$0	25 $\pm$5	8.3	802,430
BWA	97 $\pm$6	83 $\pm$5	13 $\pm$1	7.6	928,093
GEM	380 $\pm$65	373 $\pm$64	6 $\pm$1	5.5	855,313
GMAP	2,887 $\pm$95	17 $\pm$2	2,870 $\pm$94	7.6	998,454
GSNAP	40 $\pm$6	22 $\pm$6	18 $\pm$1	7.6	926,371
MicroRazerS	453 $\pm$22	0 $\pm$0	453 $\pm$22	1.8	989,089
MOSAIK	22 $\pm$2	4 $\pm$1	18 $\pm$1	15.6	267,173
Novoalign	48 $\pm$4	12 $\pm$1	36 $\pm$3	7.8	940,428
Soap2	82 $\pm$7	78 $\pm$7	4 $\pm$0	5.3	798,565
SSAHA2	207 $\pm$27	13 $\pm$2	194 $\pm$25	9.5	1e+06
Stampy	189 $\pm$19	33 $\pm$6	156 $\pm$14	4	986,593
miRNA (20 bp)					
Bowtie	119 $\pm$2	118 $\pm$2	1 $\pm$0	5	983,951
Bowtie2	124 $\pm$1	123 $\pm$1	1 $\pm$0	5.1	983,951
BWA	91 $\pm$4	87 $\pm$4	4 $\pm$1	7.4	996,470
GEM	228 $\pm$4	226 $\pm$4	3 $\pm$0	5.6	980,071
GSNAP	19 $\pm$1	17 $\pm$1	3 $\pm$0	7.6	966,802
MicroRazerS	49 $\pm$1	0 $\pm$0	49 $\pm$1	1.6	979,464
mrFAST	48 $\pm$3	33 $\pm$2	15 $\pm$1	0.8	982,049
mrsFAST	42 $\pm$1	33 $\pm$1	9 $\pm$0	0.5	979,389
PASS	28 $\pm$2	0 $\pm$0	28 $\pm$2	15.8	999,989
PERM	57 $\pm$1	0 $\pm$0	57 $\pm$1	13.4	982,545
SHRiMP	137 $\pm$22	7 $\pm$1	130 $\pm$22	2.2	962,980
RNA (75 bp)					
BFAST	35 $\pm$2	17 $\pm$1	17 $\pm$1	21.4	726,601
Blat	1,741 $\pm$131	3 $\pm$1	1,738 $\pm$131	3.8	974,710
GMAP	931 $\pm$43	18 $\pm$1	913 $\pm$43	7.6	992,079
GSNAP	68 $\pm$5	20 $\pm$1	48 $\pm$4	7.6	924,216
Novoalign	90 $\pm$3	13 $\pm$1	77 $\pm$3	7.8	795,480
Smalt	48 $\pm$2	5 $\pm$0	43 $\pm$2	5.2	996,443
SoapSplice	104 $\pm$7	76 $\pm$5	29 $\pm$3	5.4	877,911
SSAHA2	91 $\pm$5	16 $\pm$1	76 $\pm$4	9.5	999,945
TopHat	99 $\pm$13	71 $\pm$13	28 $\pm$0	5.1	807,811

Table 3. The five mappers with lowest average mapping time (Map.Time), total time (Time), memory usage (Mem) or higher number of reads aligned when mapping 1 million (BS/DNA,miRNA,RNA)-seq single-end reads against the whole human genome. Time unit is minutes and memory unit is GB. The Pre.Time column presents the pre-processing time (includes indexing) and R.Aligned column presents the number of reported aligned reads.

5 DISCUSSION

The development of numerous mappers for HTS data is motivated not only by novel developments of HTS technology but also by the growing number of biological applications. The variety of applications has led to the appearance of specific types of data (e.g., miRNA, RNA, ChIP, and bisulfite). Previously existing mappers have been adapted while others were developed from scratch to deal with these developments. The increasing number of resequencing projects has also motivated the development of mappers optimized to align reads to multiple reference genomes (e.g., DynMap and GenomeMapper). It is expected that further improvements in efficiency, multi-reference alignment, and support for longer read lengths will be topics of future research.

One commonly asked question is what is the best mapper for a given application. Although the "best mapper" criteria involves application specific requirements such as how well it works in conjunction with downstream analysis tools (e.g., variant callers), it often also includes speed and, in particular, accuracy. Despite some recent evaluation studies (Bock *et al.*, 2010; Li and Homer, 2010; Chatterjee *et al.*, 2012) determining the most accurate and fastest mappers for a particular application is still difficult. The primary challenge in assessing mappers is the lack of gold standard data sets for different applications and sequencing technologies. These data sets would not only include the reads but also their true locations and could be based on true data or data generated *in silico*, using novel or existing simulators such as ART (Huang *et al.*, 2011b), BEERS (Grant *et al.*, 2011), or FluxSimulator (Griebel *et al.*, 2012). The research community has started to address these issues in the context of different projects, such as the RGASP[1] and the Alignathon[2] projects, which aim, respectively, to assess the status of computational methods to map human RNAseq data and DNAseq data to whole genomes. However, no results are publicly available at this time. More generally, a common approach for comparing mappers has been to count the number of reads aligned. However, increasing the number of reads is not useful if the probability of the reads being correctly mapped decreases, i.e., if the increase in mapped reads is done at the expense of increasing the proportion of incorrectly mapped reads. One way to address this problem would be to compute the likelihood of a read being correctly mapped (e.g., as available in RMap or ZOOM) and allow the users to choose only the alignments above some threshold.

Users may want to consider several mappers in their HTS analysis and to incorporate them in pipelines, such as in ArrayExpressHTS (Goncalves *et al.*, 2011). This raises the issue of mapper interoperability. To achieve interoperability, input and output formats need to be standardized. Currently, the majority of the mappers accept input files in FASTQ or CFASTQ format and generate SAM/BAM files as output. Hence, the level of interoperability is high. However, there is still room to improve since FASTQ files include quality values encoded in different formats and BAM files can also come in different "flavours" (their standardization should be encouraged). Moreover, in the future, mappers may also include the option to output files in the CRAM format (Fritz *et al.*, 2011), which may prove useful for efficiently compressing DNA sequences. The input parameters of the mappers are far from being normalized, which makes it more difficult for a practitioner to switch between mappers. Hence, it would be useful if there was an effort to standardize the most commonly used parameters (e.g., for defining seed lengths, input/output files and formats).

Finally, the great flexibility and configurability of most mappers comes with a price: a considerable number of parameters that have to be set. Determining the best parameter values to achieve some predefined level of mapping specificity/sensitivity is far from being trivial. Mappers with the ability to automatically tune their parameters to achieve some user defined specificity/sensitivity may be a solution to this problem.

[1] RGASP: `http://www.gencodegenes.org/rgasp/`

[2] Alignathon: `http://compbio.soe.ucsc.edu/alignathon/`

ACKNOWLEDGEMENTS

We would like to thank the anonymous reviewers for their helpful and constructive comments, all developers that answered our survey, and our colleagues Jose Afonso, Mat Davis, Angela Goncalves, and Mar Gonzalez Porta for their helpful comments and discussion.

Funding: The research leading to these results has received funding from the European Community's FP7 HEALTH grants CAGEKID (grant agreement 241669), ENGAGE (grant agreement 201413), EurocanPlatform (grant agreement 260791), and GEUVADIS (grant agreement 261123).

REFERENCES

Alkan, C., Kidd, J., Marques-Bonet, T., Aksay, G., Antonacci, F., Hormozdiari, F., Kitzman, J., Baker, C., Malig, M., Mutlu, O., *et al.* (2009). Personalized copy number and segmental duplication maps using next-generation sequencing. *Nature Genetics*, **41**(10), 1061–1067.

Au, K., Jiang, H., Lin, L., Xing, Y., and Wong, W. (2010). Detection of splice junctions from paired-end RNA-seq data by SpliceMap. *Nucleic Acids Research*, **38**, 4570–4578.

Bock, C., Tomazou, E., Brinkman, A., Müller, F., Simmer, F., Gu, H., Jäger, N., Gnirke, A., Stunnenberg, H., and Meissner, A. (2010). Quantitative comparison of genome-wide DNA methylation mapping technologies. *Nature Biotechnology*, **28**(10), 1106–1114.

Bryant Jr, D., Shen, R., Priest, H., Wong, W., and Mockler, T. (2010). Supersplat-spliced RNA-seq alignment. *Bioinformatics*, **26**(12), 1500–1505.

Campagna, D., Albiero, A., Bilardi, A., Caniato, E., Forcato, C., Manavski, S., Vitulo, N., and Valle, G. (2009). PASS: a program to align short sequences. *Bioinformatics*, **25**(7), 967–968.

Chatterjee, A., Stockwell, P., Rodger, E., and Morison, I. (2012). Comparison of alignment software for genome-wide bisulphite sequence data. *Nucleic Acids Research*.

Chen, P., Cokus, S., and Pellegrini, M. (2010). BS Seeker: precise mapping for bisulfite sequencing. *BMC Bioinformatics*, **11**(1), 203.

Chen, Y., Souaiaia, T., and Chen, T. (2009). PerM: efficient mapping of short sequencing reads with periodic full sensitive spaced seeds. *Bioinformatics*, **25**, 2514–2521.

Clement, N., Snell, Q., Clement, M., Hollenhorst, P., Purwar, J., Graves, B., Cairns, B., and Johnson, W. (2010). The GNUMAP algorithm: unbiased probabilistic mapping of oligonucleotides from next-generation sequencing. *Bioinformatics*, **26**, 38–45.

Cloonan, N., Xu, Q., Faulkner, G., Taylor, D., Tang, D., Kolle, G., and Grimmond, S. (2009). RNA-MATE: a recursive mapping strategy for high-throughput RNA-sequencing data. *Bioinformatics*, **25**, 2615–2616.

Cokus, S., Feng, S., Zhang, X., Chen, Z., Merriman, B., Haudenschild, C., Pradhan, S., Nelson, S., Pellegrini, M., and Jacobsen, S. (2008). Shotgun bisulphite sequencing of the Arabidopsis genome reveals DNA methylation patterning. *Nature*, **452**(7184), 215–219.

David, M., Dzamba, M., Lister, D., Ilie, L., and Brudno, M. (2011). SHRiMP2: sensitive yet practical short read mapping. *Bioinformatics*, **27**(7), 1011.

De Bona, F., Ossowski, S., Schneeberger, K., and Ratsch, G. (2008). Optimal spliced alignments of short sequence reads. *Bioinformatics*, **24**, i174–i180.

Eaves, H. and Gao, Y. (2009). MOM: maximum oligonucleotide mapping. *Bioinformatics*, **25**(7), 969–970.

Emde, A., Grunert, M., Weese, D., Reinert, K., and Sperling, S. (2010). MicroRazerS: rapid alignment of small RNA reads. *Bioinformatics*, **26**(1), 123–124.

Flicek, P. and Birney, E. (2009). Sense from sequence reads: methods for alignment and assembly. *Nature Methods*, **6**(11s), S6–S12.

Flouri, T., Iliopoulos, C., and Pissis, S. (2011). DynMap: mapping short reads to multiple related genomes. In *Proceedings of the 2nd ACM Conference on Bioinformatics, Computational Biology and Biomedicine*, pages 330–334. ACM.

Fritz, M., Leinonen, R., Cochrane, G., and Birney, E. (2011). Efficient storage of high throughput dna sequencing data using reference-based compression. *Genome Research*, **21**(5), 734–740.

Frousios, K., Iliopoulos, C. S., Mouchard, L., Pissis, S. P., and Tischler, G. (2010). Real: an efficient read aligner for next generation sequencing reads. In *Proceedings of the First ACM International Conference on Bioinformatics and Computational Biology*, BCB '10, pages 154–159, New York, NY, USA. ACM.

Garber, M., Grabherr, M. G., Guttman, M., and Trapnell, C. (2011). Computational methods for transcriptome annotation and quantification using RNA-seq. *Nature Methods*, **8**(6), 469–477.

Goncalves, A., Tikhonov, A., Brazma, A., and Kapushesk y, M. (2011). A pipeline for RNA-seq data processing and quality assessment. *Bioinformatics*, **27**(6), 867–869.

Grant, G., Farkas, M., Pizarro, A., Lahens, N., Schug, J., Brunk, B., Stoeckert, C., Hogenesch, J., and Pierce, E. (2011). Comparative analysis of RNA-Seq alignment algorithms and the RNA-Seq unified mapper (RUM). *Bioinformatics*, **27**(18), 2518–2528.

Griebel, T., Zacher, B., Ribeca, P., Raineri, E., Lacroix, V., Guigó, R., and Sammeth, M. (2012). Modelling and simulating generic rna-seq experiments with the flux simulator. *Nucleic Acids Research*.

Hach, F., Hormozdiari, F., Alkan, C., Birol, I., Eichler, E., and Sahinalp, S. (2010). mrsFAST: a cache-oblivious algorithm for short-read mapping. *Nature Methods*, **7**, 576–577.

Homer, N., Merriman, B., and Nelson, S. (2009). BFAST: an alignment tool for large scale genome resequencing. *PLoS ONE*, **4**, e7767.

Huang, S., Zhang, J., Li, R., Zhang, W., He, Z., Lam, T.-W., Peng, Z., and Yiu, S.-M. (2011a). SOAPsplice: genome-wide ab initio detection of splice junctions from RNA-Seq data. *Frontiers in Genetics*, **2**(0).

Huang, W., Li, L., Myers, J., and Marth, G. (2011b). ART: a next-generation sequencing read simulator. *Bioinformatics*.

Jiang, H. and Wong, W. (2008). SeqMap: mapping massive amount of oligonucleotides to the genome. *Bioinformatics*, **24**(20), 2395.

Kent, W. (2002). BLAT - the BLAST-like alignment tool. *Genome Research*, **12**(4), 656–664.

Kim, Y., Teletia, N., Ruotti, V., Maher, C., Chinnaiyan, A., Stewart, R., Thomson, J., and Patel, J. (2009). ProbeMatch: rapid alignment of oligonucleotides to genome allowing both gaps and mismatches. *Bioinformatics*, **25**(11), 1424.

Krueger, F. and Andrews, S. (2011). Bismark: a flexible aligner and methylation caller for Bisulfite-Seq applications. *Bioinformatics*, **27**(11), 1571–1572.

Kurtz, S., Phillippy, A., Delcher, A., Smoot, M., Shumway, M., Antonescu, C., and Salzberg, S. (2004). Versatile and open software for comparing large genomes. *Genome Biology*, **5**(2), R12.

Lam, T., Sung, W., Tam, S., Wong, C., and Yiu, S. (2008). Compressed indexing and local alignment of DNA. *Bioinformatics*, **24**(6), 791.

Langmead, B. and Salzberg, S. (2012). Fast gapped-read alignment with Bowtie 2. *Nature Methods*.

Langmead, B., Trapnell, C., Pop, M., and Salzberg, S. (2009). Ultrafast and memory-efficient alignment of short DNA sequences to the human genome. *Genome Biololy*, **10**, R25.

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

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

Li, H. and Homer, N. (2010). A survey of sequence alignment algorithms for next-generation sequencing. *Briefings in Bioinformatics*, **11**(5), 473.

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

Li, H., Handsaker, B., Wysoker, A., Fennell, T., Ruan, J., Homer, N., Marth, G., Abecasis, G., Durbin, R., *et al.* (2009a). The sequence alignment/map format and SAMtools. *Bioinformatics*, **25**(16), 2078.

Li, J., Schmieder, R., Ward, R., Delenick, J., Olivares, E., and Mittelman, D. (2012). Seqanswers: an open access community for collaboratively decoding genomes. *Bioinformatics*, **28**(9), 1272–1273.

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

Li, R., Yu, C., Li, Y., Lam, T., Yiu, S., Kristiansen, K., and Wang, J. (2009b). SOAP2: an improved ultrafast tool for short read alignment. *Bioinformatics*, **25**, 1966–1967.

Li, Y., Terrell, A., and Patel, J. (2011). WHAM: a high-throughput sequence alignment method. In *Proceedings of the 2011 international conference on Management of data*, pages 445–456. ACM.

Lieberman-Aiden, E., Van Berkum, N., Williams, L., Imakaev, M., Ragoczy, T., Telling, A., Amit, I., Lajoie, B., Sabo, P., Dorschner, M., *et al.* (2009). Comprehensive mapping of long-range interactions reveals folding principles of the human genome. *Science*, **326**(5950), 289.

Lin, H., Zhang, Z., Zhang, M. Q., Ma, B., and Li, M. (2008). ZOOM! Zillions of oligos mapped. *Bioinformatics*, **24**(21), 2431–2437.

Lunter, G. and Goodson, M. (2011). Stampy: A statistical algorithm for sensitive and fast mapping of Illumina sequence reads. *Genome Research*, **21**(6), 936–939.

Malhis, N. and Jones, S. (2010). High quality snp calling using illumina data at shallow coverage. *Bioinformatics*, **26**(8), 1029–1035.

Malhis, N., Butterfield, Y., Ester, M., and Jones, S. (2009). Slider-maximum use of probability information for alignment of short sequence reads and SNP detection. *Bioinformatics*, **25**(1), 6–13.

Mardis, E. (2008). Next-generation DNA sequencing methods. *Annu. Rev. Genomics Hum. Genet.*, **9**, 387–402.

Marioni, J., Mason, C., Mane, S., Stephens, M., and Gilad, Y. (2008). RNA-seq: an assessment of technical reproducibility and comparison with gene expression arrays. *Genome Research*, **18**, 1509–1517.

Meissner, A., Mikkelsen, T., Gu, H., Wernig, M., Hanna, J., Sivachenko, A., Zhang, X., Bernstein, B., Nusbaum, C., Jaffe, D., *et al.* (2008). Genome-scale DNA methylation maps of pluripotent and differentiated cells. *Nature*, **454**(7205), 766–770.

Metzker, M. (2009). Sequencing technologies: the next generation. *Nature Reviews Genetics*, **11**(1), 31–46.

Ning, Z., Cox, A., and Mullikin, J. (2001). SSAHA: a fast search method for large DNA databases. *Genome Research*, **11**(10), 1725–1729.

Ondov, B., Varadarajan, A., Passalacqua, K., and Bergman, N. (2008). Efficient mapping of Applied Biosystems SOLiD sequence data to a reference genome for functional genomic applications. *Bioinformatics*, **24**(23), 2776–2777.

Ozsolak, F. and Milos, P. (2010). RNA sequencing: advances, challenges and opportunities. *Nature Reviews Genetics*, **12**(2), 87–98.

Park, P. (2009). Chip–seq: advantages and challenges of a maturing technology. *Nature Reviews Genetics*, **10**(10), 669–680.

Prüfer, K., Stenzel, U., Dannemann, M., Green, R., Lachmann, M., and Kelso, J. (2008). PatMaN: rapid alignment of short sequences to large databases. *Bioinformatics*, **24**(13), 1530–1531.

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

Schatz, M. (2009). CloudBurst: highly sensitive read mapping with MapReduce. *Bioinformatics*, **25**(11), 1363–1369.

Schneeberger, K., Hagmann, J., Ossowski, S., Warthmann, N., Gesing, S., Kohlbacher, O., and Weigel, D. (2009). Simultaneous alignment of short reads against multiple genomes. *Genome Biology*, **10**(9).

Slater, G. and Birney, E. (2005). Automated generation of heuristics for biological sequence comparison. *BMC Bioinformatics*, **6**(1), 31.

Smith, A., Xuan, Z., and Zhang, M. (2008). Using quality scores and longer reads improves accuracy of Solexa read mapping. *BMC Bioinformatics*, **9**(1), 128.

Trapnell, C. and Salzberg, S. L. (2009). How to map billions of short reads onto genomes. *Nature Biotechnology*, **27**(5), 455–457.

Trapnell, C., Pachter, L., and Salzberg, S. (2009). TopHat: discovering splice junctions with RNA-Seq. *Bioinformatics*, **25**, 1105–1111.

Wang, K., Singh, D., Zeng, Z., Coleman, S., Huang, Y., Savich, G., He, X., Mieczkowski, P., Grimm, S., Perou, C., MacLeod, J., Chiang, D., Prins, J., and Liu, J. (2010). MapSplice: Accurate mapping of RNA-seq reads for splice junction discovery. *Nucleic Acids Research*, **38**, e178.

Wang, Z., Gerstein, M., and Snyder, M. (2009). RNA-Seq: a revolutionary tool for transcriptomics. *Nature Reviews Genetics*, **10**, 57–63.

Weese, D., Emde, A., Rausch, T., Doring, A., and Reinert, K. (2009). RazerS - fast read mapping with sensitivity control. *Genome Research*, **19**, 1646–1654.

Wood, D., Xu, Q., Pearson, J., Cloonan, N., and Grimmond, S. (2011). X-MATE: a flexible system for mapping short read data. *Bioinformatics*, **27**(4), 580.

Wu, T. and Nacu, S. (2010). Fast and SNP-tolerant detection of complex variants and splicing in short reads. *Bioinformatics*, **26**, 873–881.

Wu, T. and Watanabe, C. (2005). GMAP: a genomic mapping and alignment program for mRNA and EST sequences. *Bioinformatics*, **21**(9), 1859.

Xi, Y. and Li, W. (2009). BSMAP: whole genome bisulfite sequence MAPping program. *BMC Bioinformatics*, **10**(1), 232.

Zhang, Y., Lameijer, E., AC't Hoen, P., Ning, Z., Slagboom, P., and Ye, K. (2012). PASSion: a pattern growth algorithm-based pipeline for splice junction detection in paired-end RNA-Seq data. *Bioinformatics*, **28**(4), 479–486.