

Review

Alignment of Short Reads: A Crucial Step for Application of Next-Generation Sequencing Data in Precision Medicine

Hao Ye, Joe Meehan, Weida Tong and Huixiao Hong *

Division of Bioinformatics and Biostatistics, National Center for Toxicological Research, U.S. Food and Drug Administration, 3900 NCTR Road, Jefferson, AR 72079, USA;

E-Mails: hao.ye@fda.hhs.gov (H.Y.); joe.meehan@fda.hhs.gov (J.M.);

weida.tong@fda.hhs.gov (W.T.)

* Author to whom correspondence should be addressed; E-Mail: Huixiao.Hong@fda.hhs.gov; Tel.: +870-543-7296; Fax: +870-543-7854.

Academic Editor: Afzal R. Mohammed

Received: 21 October 2015 / Accepted: 17 November 2015 / Published: 23 November 2015

Abstract: Precision medicine or personalized medicine has been proposed as a modernized and promising medical strategy. Genetic variants of patients are the key information for implementation of precision medicine. Next-generation sequencing (NGS) is an emerging technology for deciphering genetic variants. Alignment of raw reads to a reference genome is one of the key steps in NGS data analysis. Many algorithms have been developed for alignment of short read sequences since 2008. Users have to make a decision on which alignment algorithm to use in their studies. Selection of the right alignment algorithm determines not only the alignment algorithm but also the set of suitable parameters to be used by the algorithm. Understanding these algorithms helps in selecting the appropriate alignment algorithm for different applications in precision medicine. Here, we review current available algorithms and their major strategies such as seed-and-extend and q-gram filter. We also discuss the challenges in current alignment algorithms, including alignment in multiple repeated regions, long reads alignment and alignment facilitated with known genetic variants.

Keywords: precision medicine; next-generation sequencing; genetic variants; alignment; short reads

1. Introduction

Under the “one-size fits all” therapy model in conventional medicine, certain medical interventions can be more effective or cause fewer side effects for some patients than for others. Therefore, it is important to identify the potential patients who are more or less likely to benefit from the intervention. Precision medicine, or personalized medicine, has been proposed as a modernized and promising medical strategy, which emphasizes prevention and treatment strategies that take individual variability into account [1]. Thus, suitable individuals can receive proper treatment based on their individual genetic makeups. Although many factors, including environment, lifestyle and medical history contribute to the differences in treatment of drugs among individuals, genomics provides the most comprehensive genetic characteristics of each individual and is often believed to be the leading driver of precision medicine [2].

Genetic biomarkers play key roles in implementation of precision medicine. There has been much effort to advance biomarker discovery and application in regulatory science [3–8]. Emerging technologies have been used for biomarker development [9]. Many genetic biomarkers that are used in clinical practice and drug development were identified through genome-wide association studies (GWAS) using genotyping microarray technology [10]. There are some sources of microarray genotyping errors [11], including batch effect [12–15] and variation in genotype calling algorithms [16–18], which are considered part of the reason for why GWAS have not fully satisfied the expectations of scientists to completely decipher the human genetic architecture. Recently, next-generation sequencing (NGS) technologies have emerged as the most popular tools for deciphering human genetic variations [19], profiling miRNA [20], and identifying genetic biomarkers for clinical diagnosis [21] and prognosis [22]. Quality control is important for better utilization of NGS data [23] and proteomics data [24].

Scientists have already launched several large human genetics projects in order to obtain a detailed catalogue of human genetic variation, such as the 1000 genomes project [25] and the Yan Huang project [26]. The illumina estimation indicates that, as of 2014, ~228,000 human genomes had been completely sequenced in the world [27]. The number of human genomes sequenced is expected to double about every 12 months, reaching ~1.6 million genomes by 2017 [28]. With the cost of human sequencing having dramatically dropped from \$3 billion for the Human Genome Project to \$1000 currently achieved by illumina X Ten system, the bottleneck of genomics has shifted from sequencing experiments to analyzing and interpreting the sequencing data [29,30].

Figure 1 gives a typical workflow in genetic studies using NGS (next-generation sequencing) technology, including DNA extraction, DNA library building, sequencing, alignment, genetic variants detection and downstream data analysis. Millions of raw reads with a length from 175 to 300 bp are usually generated from current NGS platforms (Table 1). However, the latest developed PacBio RII platform generates very long reads (up to 40k bp). With the raw reads, data analysis methods are used to detect the genetic variants in the samples. Alignment of these raw reads into a reference genome is the first and essential step in almost all applications, such as genetic variants detection, methylation patterns profiling (MeDIP-Seq), protein-DNA interactions mapping (ChIP-Seq), and differentially expressed gene identification (RNA-Seq). All these applications require aligning large quantities of short reads to the human genome in a reasonably short time. Generally, the alignment process should quickly determine the correct position of the reads in the genome in consideration of sequencing error

and heterozygous variation [31]. To keep pace with the high-speed development of sequencing technologies, many alignment algorithms and tools have been developed in the last few years. Table 2 summarizes some popular alignment algorithms and software tools, without the intension of giving a complete list.

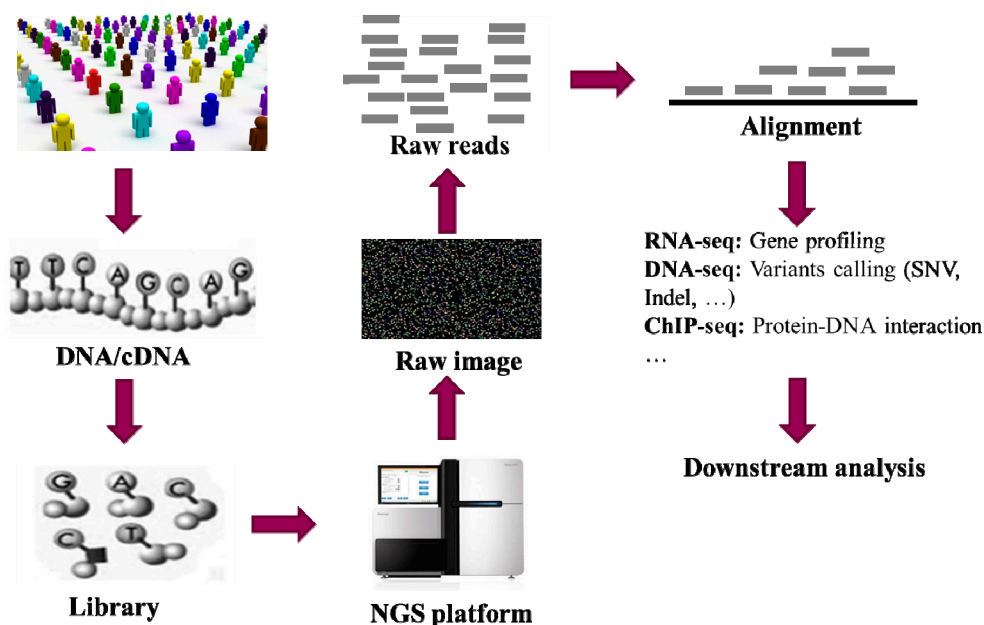






Figure 1. A brief flow chart of genetic studies using NGS. In the first step, DNA or cDNA samples are extracted from the cells of individuals. Then each of the samples is cleaved into small fragments and PCR is carried out to build the library by amplifying each of the small fragments. The library is then sequenced using the NGS platform. The original output from the NGS platform is a set of images. Thereafter, a base-calling algorithm is used to processing the images and output the so-called “raw reads”. An alignment algorithm is then used to align the millions of short reads onto the reference human genome followed up by genetic variants detection for downstream analysis in the genetics studies.

In this article, we review the basic strategies frequently used in current alignment algorithms. We also discuss the challenges in alignment of short reads.

Table 1. The most frequently used NGS platforms *.

Platform Name	Illumina HiSeq 2500	Ion Torrent-Proton II	PacBio RS II	OxFord Nanopore Minion
Instrument				
Cost (USD) **	690 k	224 k	695 k	1 k ***
Reagent cost Per run/per GB	4126/45.84	1000/20.41	100/1111.11	900/1000
Reads per run	300 millions	280 millions	0.03 millions	0.1 millions
Average Read length	2 × 150 bp	175 bp	14,000 bp	9,000 bp
Run time	10 h	5 h	2 h	6 h
Major errors	substitution	indel	indel	deletion
Error rate (%)	0.1	1	1	4
Amplification	bridgePCR	emPCR	none, SMS	none, SMS
Advantage	low cost per GB; high output	low cost	long reads; no amplification bias	long reads; no amplification bias
Disadvantage	high cost	homopolymer errors	low throughput; high cost	high error rate

* Sources: <http://www.molecularecologist.com/next-gen-fieldguide-2014/> and websites of the companies;

** Sources: <http://www.molecularecologist.com/next-gen-table-3a-2014/>;

*** Accessing fee. Sources: <https://www.nanoporetech.com/products-services/minion-mki>.

Table 2. Alignment algorithms and software tools.

Name	Website	Reference	Remark
SOAP *	soap.genomics.org.cn	[32–35]	k-mer inexact match seed; support at most 3 mismatches; GPU calculation supported
CUSHAW \$	cushaw3.sourceforge.net/homepage.htm#downloads	[36–39]	k-mer inexact match, maximal exact match and hybrid seeds; GPU supported
Bowtie &	bowtie-bio.sourceforge.net	[40,41]	k-mer inexact match seed; high speed; double-index; up to 3 mismatches
BWA	bio-bwa.sourceforge.net	[42,43]	k-mer inexact match and maximal exact match seed
GASSST	www.irisa.fr/symbiose/projects/gasst/	[44]	k-mer exact match seed; it currently has been tested for reads up to 500 bp
GNUMAP	dna.cs.byu.edu/gnumap/	[45]	k-mer exact match seed; probabilistically mapping reads to repeat regions
MOSAİK	gkno.me/pipelines.html#mosaik	[46]	k-mer exact match seed
NextGenMap	cibiv.github.io/NextGenMap/	[47]	q-gramq-gram filter; GPU calculation supported
QPALMA	www.raetschlab.org/suppl/qpalma	[48]	k-mer inexact match; incorporate read quality score and splice site
RMAP	rulai.cshl.edu/rmap/	[49,50]	k-mer inexact match seed; 10 mismatches allowed; incorporate read quality score
Segemehl	www.bioinf.uni-leipzig.de/Software/segemehl/	[51]	k-mer inexact match seed; enhanced suffix arrays
SeqMap	www-personal.umich.edu/~jianghui/seqmap/	[52]	k-mer inexact match; support windows, linux, Mac OS

Table 2. Cont.

Name	Website	Reference	Remark
Stampy	www.well.ox.ac.uk/project-stampy	[53]	k-mer inexact match; support up to 30 bp indels in paired-end reads alignment
Cloudburst	sourceforge.net/projects/cloudburst-bio/	[54]	Highly sensitive read mapping with MapReduce.
drFAST	drfast.sourceforge.net/	[55]	k-mer inexact match; specially designed for better delineation of structural variants
BFAST	sourceforge.net/projects/bfast/	[56]	k-mer spaced seeds
MAQ	maq.sourceforge.net	[57]	k-mer spaced seeds; incorporate quality scores of reads in alignment
MOM	go.vcu.edu/mom	[58]	k-mer spaced seeds; unlimited mismatches in the 3' and 5' flanking regions.
PASS	pass.cribi.unipd.it	[59]	k-mer spaced seeds; implemented in C++ and supported on Linux and Windows
PerM	code.google.com/p/perm/	[60]	k-mer spaced seeds; 9 mismatches are allowed
SHRiMP2	compbio.cs.toronto.edu/shrimp/	[61,62]	combined k-mer spaced seeds and q-gram filter
ZOOM	www.bioinfor.com/zoom/general/overview.html	[63]	k-mer spaced seeds; tolerate 2 mismatches by default
BarraCUDA	seqbarracuda.sourceforge.net/	[64]	Incorporate GPU to speed up BWA
GEM	gemlibrary.sourceforge.net/	[65]	q-gram filter
MPSCAN	www.atgc-montpellier.fr/mpscan/	[66]	q-gram filter; support Windows, linux, Mac OS
ERNE	iga-rna.sourceforge.net/	[67]	long gap support; Works on Windows, Mac OS X, linux
SARUMAN	www.cebitec.uni-bielefeld.de/brf/saruman/saruman.html	[68]	k-mer inexact matched seed; support GPU calculation
LAST	last.cbrc.jp/	[69]	adaptive seed
Genalice	www.genalice.com/product/genalice-map/	NA	cloud calculation; High sensitivity for SNPs and long INDELS
Novoalign	www.novocraft.com/	NA	support up to 7 and 16 mismatches in single-end and pair-end reads.
PRIMEX	bioinformatics.cribi.unipd.it/primex	[70]	k-mer inexact match seed; written in C++; lookup table and server functionality
SOCS	solidsoftwaretools.com/gf/project/socs/	[71]	good at align CpG methylation-enriched reads
SToRM	bioinfo.lifl.fr/yass/iedera_solid/storm/	[72]	doesn't support pair-end reads
iSAAC	https://github.com/sequencing/isaac_aligner	[73]	k-mer inexact match seed; high speed
RazerS	www.seqan.de/projects/razers/	[74]	q-gram filter; support Windows, linux, Mac OS X
SSAHA2	www.sanger.ac.uk/resources/software/ssaha2/	[75]	k-mer inexact match seed; support various output formats
UGENE	ugene.unipro.ru/	[76]	works on Windows, linux and Mac OS X

* Include SOAP, SOAP2, SOAP3 and SOAP3-dp; [§] Include CUSHAW (k-mer inexact match seed), CUSHAW2 (maximal exact match seed) and CUSHAW3 (hybrid seeds); [&] Include Bowtie and Bowtie2. NA: commercial software, no reference available.

2. Strategies of Current Alignment Algorithms

In theory, a read can be successfully aligned onto a reference genome by applying a series of insertions, deletions, and substitutions. An alignment algorithm assigns a score to the alignment of a short read onto a reference to estimate how well they align. The score is used to identify the optimized location of the read in the reference genome. A good alignment algorithm is able to map reads onto a reference genome rapidly and accurately. Currently, most alignment algorithms utilize two major strategies: seed and extend and q-grams filter. In addition, the index methods that are used to memory-efficiently organize the reference genome and short reads are different among the alignment algorithms. Hash-table data structure was initially designed to scan and index sequence (raw reads or reference) in the first wave of alignment programs such as MAQ and SOAP [32]. However, Borrows–Wheeler Transformation (BWT) based FM-index was adopted by subsequently developed alignment algorithms such as BOWTIE [40], BWA [42] and SOAP2 [33]. Compared with the large memory usage in a hash-table based index, BWT based FM-index could index the human genome in less than 5.4 GB of memory [77]. It is interesting to note that the final index for the human genome used by BWA is approximately 2.3 GB in size. Although such index methods are important components in alignment algorithms, in this review, we will only focus on the two basic alignment strategies mentioned above. For each of the two strategies, only a few popular alignment algorithms will be selected for detailed illustration.

2.1. Seed-and-Extend Strategy

Seed-and-extend strategy is based on the observation that a good alignment should contain exact or inexact short matches between two sequences. Figure 2 shows a general view of seed-and-extend strategy that contains four steps: seed generation, seed mapping, extending each matched seed, and alignment of the read to the reference sequence. Seeds are the shorter sequences extracted from a read and can be generated using different methods. For example, k -mer seeds are generated by sliding a window of length k over the read. The seeds that exactly match a reference sequence can be identified through a mapping process facilitated by an index method. Each of the exactly matched seeds is then extended on both right and left direction under certain constraints such as maximum mismatches and length of indels. Standard dynamic programs, based on Needleman–Wunsch (NW) [78] algorithm or Smith–Waterman (SW) algorithm [79], are implemented to do the final alignment. Seed extension is usually more time-consuming than seed generation and final mapping, especially when the majority of the exacted mapped seeds cannot be completely extended to accomplish the alignment on the reference sequence. Therefore, seed filtration strategies are frequently used before extension. In addition, the seed length used by an alignment program has a substantial impact on its performance. Shorter seeds increase sensitivity, whereas longer seeds increase speed.

2.1.1. k -mer Exact Match Seed

A k -mer exact match seed is a shorter sequence of k bases that exactly matches with a reference sequence. The strategy to use k -mer exact match seeds was first utilized in BLAST [80]. In brief, k -mer (default 11-mer) seeds that exactly match certain regions in reference genome are identified.

Those seeds are then extended to match the reference genome without gaps. The final alignment results are generated by the SW modification on the extending sequence. Several improvements have been made on the extending and dynamic programming used in this strategy. For example, GNUMAP [45] incorporates the base quality of a read into NW algorithm to improve the alignment accuracy and uses 9-mer seeds to initiate the mapping process.

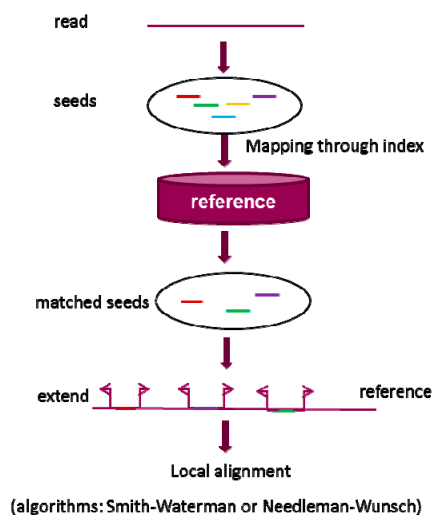


Figure 2. A brief workflow of seed-and-extend strategy in alignment. Generally, the strategy can be divided into three steps: (1) generate raw seed from a read; (2) identify the matched seed; and (3) extend the matched seed and do the local alignment through standard dynamic programming algorithms based on Needleman–Wunsch (NW) [78] algorithm or Smith–Waterman (SW) algorithm [79].

2.1.2. k-mer Inexact Match Seed

A k-mer inexact match seed is generated from a read based on pigeonhole principle. The strategy using k-mer inexact match seeds supports mismatches and indels in mapping. The rationale behind the strategy using k-mer inexact match seeds is that if m bases are allowed to mismatch between a read and a reference sequence, the read is has n bases, and the read can be chopped into non-overlapping k-mers ($k = n/(m + 1)$), at least one exact match k-mer seed exists. The k-mer inexact match has been utilized in many alignment algorithms.

SOAP [32] splits a read into fragments, based on the number of mismatches allowed (default five), to implement the strategy using inexact match seeds. Newer SOAP versions improve the alignment speed but use the same seeding strategy. SOAP2 [33] speeds up the mapping process using a reference sequence that is indexed by the combination of BWT and hash table. The graphics processing unit (GPU) is incorporated by SOAP3 to facilitate parallel calculation [34]. More recently, SOAP3-dp [35] utilized dynamic programming and bi-directional BWT to reduce unsuccessful extending of the seeds with multiple locations in the reference.

Bowtie [40,41] generates k-mer inexact match seeds with at most three (default two) mismatches in the high-quality end of a read (default: the first 28 bp in the read). The “double indexing” technology is the one of major contributions to the high speed of Bowtie. One is called “forward” index, which

contains the BWT of a reference sequence, and another is referred as “mirror” index, which is composed of the BWT of the reversed reference sequence. Using double indexes, exactly matched seeds can be quickly identified. Bowtie uses a cutoff, the maximum acceptable quality score (default 70), to determine whether extension of a read continues. If the alignment has a quality score larger than the cutoff, the extension on the matched seed is stopped. This quality score constraint helps remove a lot of matched seeds for continuous extension alignment as early as possible.

BWA [42,43] generates k-mer inexact match seeds with a default of two mismatches allowed in each seed. Seeds are efficiently mapped to a reference genome, facilitated by a special index structure called prefix directed acyclic word graph (DAWG) [81]. DAWG represents the set of all substrings that are extracted from a string. In BWA, alignment speed is improved by reducing unnecessary seed extension for highly repetitive sequences. In brief, BWA heuristically identifies and discards seed extensions using a criterion in which the length of the overlapped region is shorter than the length of any previous successfully aligned regions in the reference genome. BWA only reports the alignments that are largely non-overlapped with the query sequence instead of giving all the local alignments.

2.1.3. k-mer Spaced Seed

Generally speaking, a seed allowing internal mismatches is called a spaced seed [82]. For example, in a 5-mer spaced seed “10110”, “1” indicates the position in which the base of a seed has to match with a reference sequence and “0” means the position in which the base of a seed is allowed to mismatch with a reference sequence. k-mer spaced seed was first introduced and proved to improve sensitivity of k-mer exact match in DNA homology searching by Ma [83].

RMAP [49,50] integrated base-calling quality scores to improve sensitivity in both seed mapping and extension processes. The quality scores from base-calling were used to weigh the penalties for mismatches at different positions. A mismatched base at a position with a lower quality score than the base-calling in a read is penalized less. Utilization of base-calling scores in alignment displayed high sensitivity in mapping of the bases for which the base caller has difficulties to call and thus gives low scores to the called bases.

MAQ [57] uses k-mer exact match seeds for mapping in the first step. When seeds fail to be exactly matched onto a reference sequence, MAQ generates 6-mer spaced seeds in which two or fewer mismatches are allowed in the first 28 bp for a read. This strategy saves a lot of time in seeds generation and mapping. In seed extension process, MAQ assigns each individual alignment a phred-scaled quality score (capped at 99) that is used to measure the probability that the true alignment is not the one found by MAQ. The larger phred the scaled quality score the better the alignment. The phred-scaled quality score is calculated as the sum of qualities of mismatched bases over the whole length of a read. When a read can be aligned equally well to multiple positions, MAQ randomly pick one position and gives it a mapping quality score zero.

2.1.4. Maximum Extend Match (MEM) Seed

MEM [84] is an exact match between two strings that cannot be extended in either direction without allowing a mismatch. Compared to fixed-length seeds (seed length is predefined) mentioned above, the variable seed length is the key feature of MEM seed. MEM reduces the number of mapping positions of each seed onto a reference genome. Alignment speed using MEM seeds is improved because invalid seed extensions are prevented. The efficiency of generating MEM seeds plays a key role in an alignment algorithm using MEM seeds. In general, indexing a sequence in a full-text suffix tree is the frequently used strategy in detecting MEMs [85,86]. The obvious drawback of using full-text suffix tree is the large memory usage. Recently developed methods improved index structure to reduce the memory usage. Enhanced suffix array (ESA) was first introduced as a space-sparse suffix array to replace full-text index in suffix tree [85] and can be used to find MEMs with much less memory usage [87]. Khan [84] developed an algorithm to generate a special sparse suffix array that stores every k -th position of the text. In contrast to a full-text index that stores every position of the text, a sparse suffix array uses much less memory. Fernandes developed slaMEM to detect MEMs [88]. The slaMEM algorithm uses a new index structure called longest common prefix (LCP) array and the backward search method of the FM-index and achieves a good tradeoff between mapping speed and memory usage. More recently, E-MEM was developed by Nilesh to decipher MEMs in large genome sequences. E-MEM uses much less memory and is highly amenable to parallelization. It has been reported that all MEMs of minimum length 100 between whole human and mouse genomes could be calculated within 10 min on a 12-core machine, using 2 GB of memory [89].

BWA-MEM is the latest developed algorithm in BWA software for sequence alignment. BWA-MEM utilizes a new index structure called FMD-index in which both forward and reverse strand DNA sequences are indexed. It can efficiently facilitate detection of all MEMs between a read and a reference sequence. The super-maximal exact matches (SMEMs) are the matches that are not contained in any other MEMs of the read and are chosen for seed mapping and extension. Using SMEMs saves a lot of alignment time by reducing most invalid extensions of all other MEMs in the read. If a read cannot be aligned to a reference sequence by extension of the SMEMs, BWA-MEM uses a re-seeding process to generate new seeds for mapping and extension. Specifically, when the length of a SMEM is larger than 28 bp (default), the longest MEM seed which covers the middle of the SMEM in this read is used to initialize re-seed. In seed extension, BWA-MEM stops an extension at a certain point if the difference between the best alignment score in the extension and the score at that point is larger than a predefined value that is further adjusted by number of gaps in the alignment. When an extension reaches the whole read, this algorithm accepts the alignment as a successful one mapping between the read and the reference sequence if the best improvement in alignment score in the extension is larger than a predefined value.

CUSHAW2 [38] is another software package that uses MEM seeds to initiate alignment. MEM seeds are detected from the FM-index of a read or a reference sequence. An important parameter, Q , indicating the minimum seed length is used to filter MEM seeds to avoid invalid extensions. Specifically, the default Q is set at 16, 22 and 35 for the read length of 100, 200 and 500, respectively. Users can set Q . Parallelization with GPU is implemented in CUSHAW2.

2.1.5. Adaptive Seed

Adaptive seed is the shortest sub-sequence in a read that exactly matches a reference sequence with a mapping frequency less than a predefined value. In contrast to fixed-length seeds, the length of an adaptive seed is not fixed, but determined through analyzing the mapping of possible seeds to the reference genome. To identify the adaptive seed for read, seeds with different lengths are generated at first. The frequency of mapped positions in the reference sequence for each of the generated seeds is then calculated. The seed that is the shortest among the seeds that have a mapping frequency less than a predefined value is selected as the adaptive seed for extension. Several algorithms have been developed for efficient identification of adaptive seeds.

LAST [69] was proposed by Kielbasa to reduce the redundancy of seeds in identification of adaptive seed. This algorithm selects the shortest seed among the seeds that exactly map to the reference sequence starting at the same position. LAST only reduces the redundancy by dropping off the longer overlapped seeds. However, the redundancy from the overlapped seeds that map to the reference sequence starting at different positions is not considered in this algorithm. Seeds redundancy could be eliminated by more sophisticated algorithms. However, eliminating seeds redundancy may be more time-consuming. LAST is a good tradeoff, removing part of the seed redundancy through a very simple approach.

AMAS [90] splits a read into several non-overlapping adaptive seeds. Specifically, the adaptive seeds are generated one by one through scanning the read base by base in the left-to-right direction. The first 10 bp in the read are used by default to initiate the scanning process. When the number of matches of a seed is less than a predefined frequency threshold, the seed is selected as an adaptive seed and a new seed scanning is initiated using the next 10 bp in the read. AMAS also makes an improvement on filtering the adaptive seeds, especially for the last seed in each read, which are usually shorter and map to much more locations than other seeds in the same read. In brief, AMAS only filters out last seeds of each read whose numbers of matches are higher than the predefined frequency cutoff and contribute to more than 95% of the candidate locations of their respective reads.

2.2. *q-gram Filter*

Similar to seeds, *q*-grams are small fragments extracted from a read. *q*-gram filter alignment [91] is based on the hypothesis that two sequences should contain a certain number of *q*-gram if the edit distance between them is within a certain threshold. Herein, the edit distance [92] between two sequences is defined as the minimum number of editing operations (such as insertion, deletion, and substitution) that are needed to transform one sequence into the other. Figure 3 gives an overall flowchart of alignment based on *q*-gram filter strategy. In the first step, *q*-grams from a query read are generated and mapped to a reference sequence. This is a multiple-seeds mapping process. In the second step, the highly mapped regions are selected as candidates by an inverted index of grams, leading to a relatively larger memory usage. Lastly, the candidates are aligned to the read and the false positives are identified and discarded. The difference between *q*-gram filter and seed-and-extend is that *q*-gram filter based algorithms align a read to reference sequence by multiple seeds mapping without extension, while algorithms using seed-and-extend generate an alignment between a read and a

reference sequence through mapping a single seed following by extension. Without extension process, q-grams filters based algorithms support more insertions and deletions in alignment than the algorithms using seed-and-extend strategy.

SHRiMP (Short read mapping package) [61,62] was developed by University of Toronto and is a popular alignment software utilizing q-gram filter. It incorporates the concept of spaced seed in q-grams generation and mapping to improve alignment sensitivity. Many predetermined positions such as SNP (single nucleotide polymorphism) sites in a reference genome are not required to exactly match with reads during q-grams mapping. Candidate q-grams are selected using a predefined minimum of hits on a reference sequence. Number of hits of a q-gram on a reference sequence is used to estimate its mapping goodness. The candidate q-grams of each read are sorted according to their numbers of hits on the reference sequence. SHRiMP identifies candidates for each read by top ranking the q-grams using their numbers of hits on the reference sequence. The number of top hits used to identify candidate q-grams can be defined by users.

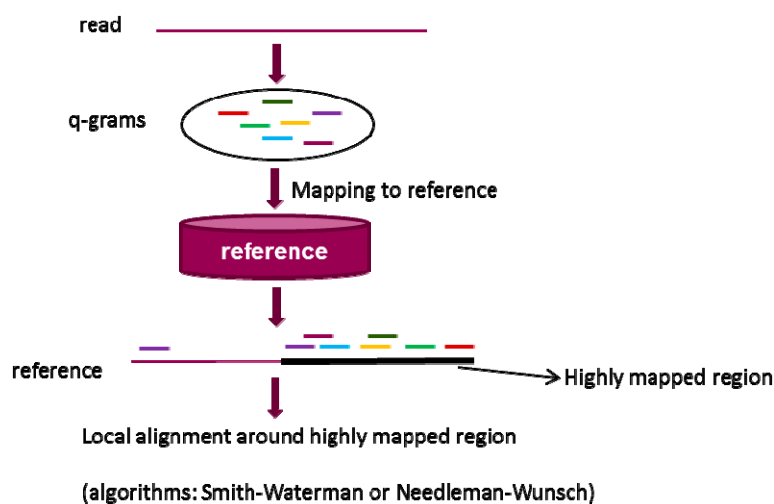


Figure 3. The overall workflow of q-gram filter in alignment. The strategy consists of three steps: (1) generation of q-grams from a read; (2) identification of highly mapped regions in a reference sequence through multiple q-grams mapping; and (3) local alignment of the read and highly mapped regions through standard dynamic programming algorithms based on Needleman–Wunsch (NW) [78] algorithm or Smith–Waterman (SW) algorithm [79].

Hobbes [93] is an optimized q-gram filter based method for efficient read alignment. It improves both q-gram generation and candidate filtering. A basic q-gram method usually generates numerous overlapped small q-grams for a read and thus needs intensive CPU calculation and large memory usage. Hobbes uses non-overlapping q-grams to optimize the number of q-grams based on a paradigm similar to the pigeonhole principle used in k-mer inexact match seed. The rationale behind this algorithm is that a read within an edit distance d to a reference sequence must contain at least n q-grams from $d + n$ non-overlapping q-grams of the reference sequence. In candidate selection, Hobbes uses two approaches to rigorously filter the highly mapped regions of non-overlapping q-grams in the reference sequence for the read. At first, it filters the candidate q-grams based on the edit distance between corresponding reference sequence and the neighboring sequence of the mapped

q-gram. The hypothesis is that if the neighbor of a matched q-gram has a large edit distance to the corresponding sequence in the read, the candidate q-gram is probably not a true match. The difference in frequency between the highly matched regions by overlapping q-grams and the read for the four bases (A, T, G, and C) is another filter to remove invalid candidate q-grams for further alignment evaluation.

3. Conclusions

The rapid development of next-generation sequencing technologies provides a promising opportunity to extend the capability of biomarker discovery in precision medicine. How to efficiently and correctly map millions of short reads to a reference genome is one of the major challenges in NGS data analysis. More specifically, speed and sensitivity are the two major concerns in current alignment algorithms, no matter which seed-and-extend or q-gram filter strategy is utilized. In this review, we have summarized the often-used alignment algorithms and discussed the approaches to achieve an optimized tradeoff between speed and sensitivity. Generally, higher sensitivity would be achieved by using shorter seeds or grams with more mismatches allowed, while alignment speed can be increased by optimizing seed generation or filtering seeds that most likely fail to extend. This review is expected to facilitate understanding of the alignment algorithms and their algorithmic parameters.

4. Further Perspectives

Many state-of-the-art software packages have already made great progresses in achieving an optimized tradeoff between speed and sensitivity. However, there are still some rooms for improvement.

One of the remaining challenges in reads alignment is how to align the reads that can be mapped to multiple repeated regions in a reference genome. According to the statistics on human reference genome version hg19, approximately 50% of the human genome has repeats [94]. Especially, some copies of the repeats are not the same but slight variants. This inevitably causes ambiguities in reads mapping. Usually, seeds or q-grams of reads are not specific and mapping may be very slow in the repeated regions. Currently, three simple methods have been used to improve alignment speed in repeated regions. The first method is to discard all seeds or q-grams that map to repeated regions. The second one is to randomly select one of the best alignments or report all of them. The last one is to select a number of top alignments. Obviously, ignoring all of the reads or part of reads mapped in the repeats may miss some important variants. In addition, the “best alignment” identified by an alignment software package in such regions may not always be correct, especially when a SNV or a small indel truly occurs in the repeat region. Nathan [45] first proposed a probabilistic model base on quality scores to align reads in repeated regions. Further efforts are still needed to improve alignment of reads in repeated regions of a reference genome.

Another challenge is to develop alignment algorithms for extremely long reads. Although most current NGS platforms produce short reads with length around 2×150 bp, there is no doubt that extremely long reads generated by so-called “third generation sequencing” platforms would be more and more promising and provide fundamentally more information than short reads. A finite coverage with short reads is not enough for deciphering a complex genome, especially for the regions where no or very few reads are mapped. A few long reads in the right spot may be able to identify the genetic variants. However, the error rate in long read platforms is a major concern in application of long reads.

Most current algorithms are designed for alignment of short reads. Many parameters in the algorithms for alignment of short reads would not be appropriate for alignment of long reads. For example, only the first 28 bp in each read was supported for at most three mismatches by default in Bowtie. Therefore, the extension process would be extremely long if a small k-mer seed was used to initiate the alignment of a long read. It is expected that new alignment algorithms will be exclusively designed for long reads, with consideration of their specific properties.

Integration of additional genetic information into sequence alignment will be a focus in the future. Currently, the majority of the alignment algorithms were designed to align reads to a single reference genome without consideration of the genetic variations in the reference genome. The assumption behind current algorithms is that the reference genome is highly similar to the genome sequenced and provides comprehensive enough genetic background. However, precision medicine needs genetic difference among individuals, thus such hypothesis is challenged. Human reference genome may not provide the most comprehensive genetic variations. A study on the novel assembly of an Asian and an African genome revealed ~19–40 Mb of novel sequences that were missed in the human reference genome [95]. In order to achieve more sensitivity and accuracy in alignment, genetic variation information, such as the 1000 genomes project [25] and the structural variation information in cancer from TCGA [96], should be integrated into alignment algorithm development in the future.

Acknowledgments

This research was supported in part by an appointment to the Research Participation Program at the National Center for Toxicological Research (Hao Ye) administered by the Oak Ridge Institute for Science and Education through an interagency agreement between the U.S. Department of Energy and the U.S. Food and Drug Administration. The content is solely the responsibility of the authors and does not necessarily represent the official views of the Food and Drug Administration.

Author Contributions

Hao Ye analyzed the algorithms and wrote the first draft of the manuscript. Joe Meehan and Weida Tong were involved in discussion of the algorithms and revised the manuscript. Huixiao Hong conceived the idea, analyzed the algorithms, and revised the manuscript. All authors read and approved the final version of the manuscript.

Conflicts of Interest

The authors declare no conflict of interest.

References

1. Collins, F.S.; Varmus, H. A new initiative on precision medicine. *N. Eng. J. Med.* **2015**, *372*, 793–795.
2. Khoury, M.J. The success of precision medicine requires a public health perspective. Available online: <http://blogs.cdc.gov/genomics/2015/01/29/precision-medicine/> (accessed on 17 November 2015).

3. Hong, H.; Goodsaid, F.; Shi, L.; Tong, W. Molecular biomarkers: A US FDA effort. *Biomark. Med.* **2010**, *4*, 215–225.
4. Hong, H.; Slikker, W., Jr. Advancing translation of biomarkers into regulatory decision making. *Biomark. Med.* **2015**, *9*, 1043–1046.
5. Gong, G.; Hong, H.; Perkins, E.J. Ionotropic GABA Receptor Antagonism-Induced Adverse Outcome Pathways for Potential Neurotoxicity Biomarkers. *Biomark. Med.* **2015**, *9*, 1225–1239.
6. Zhang, C.; Hong, H.; Mendrick, D.L.; Tang, T.; Cheng, F. Biomarker-based Drug Safety Assessment in the Age of Systems Pharmacology: From Foundational to Regulatory Science. *Biomark. Med.* **2015**, *9*, 1241–1252.
7. Wang, Y.; Liu, Z.; Zou, W.; Hong, H.; Fang, H.; Tong, W. Molecular Regulation of miRNAs and Potential Biomarkers in the Progression of Hepatic Steatosis. *Biomark. Med.* **2015**, *9*, 1189–1200.
8. Koturbash, I.; Tolleson, W.H.; Guo, L.; Yu, D.; Chen, S.; Hong, H.; Mattes, W.; Ning, B. MicroRNAs as Pharmacogenomic Biomarkers for Drug Efficacy and Drug Safety Assessment. *Biomark. Med.* **2015**, *9*, 1153–1176.
9. Hong, H.; Tong, W. Emerging efforts for discovering new biomarkers of liver disease and hepatotoxicity. *Biomark. Med.* **2014**, *8*, 143.
10. Hong, H.; Xu, L.; Liu, J.; Jones, W.D.; Su, Z.; Ning, B.; Perkins, R.; Ge, W.; Miclaus, K.; Zhang, L.; *et al.* Technical reproducibility of genotyping snp arrays used in genome-wide association studies. *PLoS ONE* **2012**, *7*, e44483.
11. Hong, H.; Shi, L.; Su, Z.; Ge, W.; Jones, W.; Czika, W.; Miclaus, K.; Lambert, C.; Vega, S.; Zhang, J.; *et al.* Assessing sources of inconsistencies in genotypes and their effects on genome-wide association studies with hapmap samples. *Pharmacogenomics J.* **2010**, *10*, 364–374.
12. Hong, H.; Su, Z.; Ge, W.; Shi, L.; Perkins, R.; Fang, H.; Xu, J.; Chen, J.J.; Han, T.; Kaput, J.; *et al.* Assessing batch effects of genotype calling algorithm brlmm for the affymetrix genechip human mapping 500 k array set using 270 hapmap samples. *BMC Bioinforma.* **2008**, *9*, S17.
13. Miclaus, K.; Wolfinger, R.; Vega, S.; Chierici, M.; Furlanello, C.; Lambert, C.; Hong, H.; Zhang, L.; Yin, S.; Goodsaid, F. Batch effects in the brlmm genotype calling algorithm influence gwas results for the affymetrix 500k array. *Pharmacogenomics J.* **2010**, *10*, 336–346.
14. Luo, J.; Schumacher, M.; Scherer, A.; Sanoudou, D.; Megherbi, D.; Davison, T.; Shi, T.; Tong, W.; Shi, L.; Hong, H.; *et al.* A comparison of batch effect removal methods for enhancement of cross-batch prediction performance using MAQC-II microarray gene expression data. *Pharmacogenomics J.* **2010**, *10*, 278–291.
15. Hong, H.; Shi, L.; Fuscoe, J.C.; Goodsaid, F.; Mendrick, D.; Tong, W. Potential sources of spurious associations and batch effects in genome-wide association studies. In *Batch Effects and Noise in Microarray Experiments: Sources and Solutions*, Scherer, A., Ed.; John Wiley & Sons: West Sussex, UK, 2009; pp. 191–201.
16. Hong, H.; Su, Z.; Ge, W.; Shi, L.; Perkins, R.; Fang, H.; Mendrick, D.; Tong, W. Evaluating variations of genotype calling: A potential source of spurious associations in genome-wide association studies. *J. Genetics* **2010**, *89*, 55–64.
17. Zhang, L.; Yin, S.; Miclaus, K.; Chierici, M.; Vega, S.; Lambert, C.; Hong, H.; Wolfinger, R.; Furlanello, C.; Goodsaid, F. Assessment of Variability in GWAS with CRLMM Genotyping Algorithm on WTCCC Coronary Artery Disease. *Pharmacogenomics J.* **2010**, *10*, 347–354.

18. Miclaus, K.; Chierici, M.; Lambert, C.; Zhang, L.; Vega, S.; Hong, H.; Yin, S.; Furlanello, C.; Wolfinger, R.; Goodsaid, F. Variability in GWAS Analysis: the Impact of Genotype Calling Algorithm Inconsistencies. *Pharmacogenomics J.* **2010**, *10*, 324–335.
19. Zhang, W.; Meehan, J.; Su, Z.; Ng, H.W.; Shu, M.; Luo, H.; Ge, W.; Perkins, R.; Tong, W.; Hong, H. Whole genome sequencing of 35 individuals provides insights into the genetic architecture of korean population. *BMC Bioinforma.* **2014**, *15*, S6.
20. Liu, J.; Jennings, S.F.; Tong, W.; Hong, H. Next generation sequencing for profiling expression of miRNAs: Technical progress and applications in drug development. *J. Biomed. Sci. Eng.* **2011**, *4*, 666–676.
21. Su, Z.; Ning, B.; Fang, H.; Hong, H.; Perkins, R.; Tong, W.; Shi, L. Next-generation sequencing and its applications in molecular diagnostics. *Expert Rev. Mol. Diagn.* **2011**, *11*, 333–343.
22. Zhang, W.; Yu, Y.; Hertwig, F.; Thierry-Mieg, J.; Zhang, W.; Thierry-Mieg, D.; Wang, J.; Furlanello, C.; Devanarayan, V.; Cheng, J.; *et al.* Comparison of RNA-seq and microarray-based models for clinical endpoint prediction. *Genome Biol.* **2015**, *16*, 1–12.
23. Zhang, W.; Soika, V.; Meehan, J.; Su, Z.; Ge, W.; Ng, H.; Perkins, R.; Simonyan, V.; Tong, W.; Hong, H. Quality control metrics improve repeatability and reproducibility of single-nucleotide variants derived from whole-genome sequencing. *Pharmacogenomics J.* **2015**, *15*, 298–309.
24. Hong, H.; Dragan, Y.; Epstein, J.; Teitel, C.; Chen, B.; Xie, Q.; Fang, H.; Shi, L.; Perkins, R.; Tong, W. Quality control and quality assessment of data from surface-enhanced laser desorption/ionization (SELDI) time-of flight (TOF) mass spectrometry (MS). *BMC Bioinforma.* **2005**, *6*, S5.
25. 1000 Genomes Project Consortium. An integrated map of genetic variation from 1,092 human genomes. *Nature* **2012**, *491*, 56–65.
26. Qiu, J.; Hayden, E.C. Genomics sizes up. *Nature* **2008**, *451*, 234.
27. Regalado, A. Emtech: Illumina says 228,000 human genomes will be sequenced this year. Available online: <http://www.technologyreview.com/news/531091/emtech-illumina-says-228000-human-genomes-will-be-sequenced-this-year/> (accessed on 17 November 2015).
28. Bioethics news. Available online: <http://www.bioethics.net/news/emtech-illumina-says-228000-human-genomes-will-be-sequenced-this-year/> (accessed on 20 November 2015).
29. Hong, H.; Zhang, W.; Shen, J.; Su, Z.; Ning, B.; Han, T.; Perkins, R.; Shi, L.; Tong, W. Critical role of bioinformatics in translating huge amounts of next-generation sequencing data into personalized medicine. *Sci. China Life Sci.* **2013**, *56*, 110–118.
30. Ning, B.; Su, Z.; Mei, N.; Hong, H.; Deng, H.; Shi, L.; Fuscoe, J.C.; Tolleson, W.H. Toxicogenomics and cancer susceptibility: advances with next-generation sequencing. *J. Environ. Sci. Health Part C* **2014**, *32*, 121–158.
31. Trapnell, C.; Salzberg, S.L. How to map billions of short reads onto genomes. *Nature Biotechnol.* **2009**, *27*, 455–457.
32. Li, R.; Li, Y.; Kristiansen, K.; Wang, J. Soap: Short oligonucleotide alignment program. *Bioinformatics* **2008**, *24*, 713–714.
33. Li, R.; Yu, C.; Li, Y.; Lam, T.-W.; Yiu, S.-M.; Kristiansen, K.; Wang, J. Soap2: An improved ultrafast tool for short read alignment. *Bioinformatics* **2009**, *25*, 1966–1967.

34. Liu, C.-M.; Wong, T.; Wu, E.; Luo, R.; Yiu, S.-M.; Li, Y.; Wang, B.; Yu, C.; Chu, X.; Zhao, K. Soap3: Ultra-fast gpu-based parallel alignment tool for short reads. *Bioinformatics* **2012**, *28*, 878–879.
35. Luo, R.; Wong, T.; Zhu, J.; Liu, C.-M.; Zhu, X.; Wu, E.; Lee, L.-K.; Lin, H.; Zhu, W.; Cheung, D.W.; *et al.* Soap3-dp: Fast, accurate and sensitive gpu-based short read aligner. *PLoS ONE* **2013**, *8*, e65632.
36. Liu, Y.; Schmidt, B.; Maskell, D.L. Cushaw: A cuda compatible short read aligner to large genomes based on the burrows-wheeler transform. *Bioinformatics* **2012**, *28*, 1830–1837.
37. Liu, Y.; Schmidt, B. Long read alignment based on maximal exact match seeds. *Bioinformatics* **2012**, *28*, i318–i324.
38. Liu, Y.; Schmidt, B. Cushaw2-GPU: Empowering faster gapped short-read alignment using GPU computing. *Design Test IEEE* **2014**, *31*, 31–39.
39. Liu, Y.; Popp, B.; Schmidt, B. Cushaw3: Sensitive and accurate base-space and color-space short-read alignment with hybrid seeding. *PLoS ONE* **2014**, *9*, e86869.
40. Langmead, B.; Trapnell, C.; Pop, M.; Salzberg, S.L. Ultrafast and memory-efficient alignment of short DNA sequences to the human genome. *Genome Biol.* **2009**, *10*, R25.
41. Langmead, B.; Salzberg, S.L. Fast gapped-read alignment with bowtie 2. *Nature Methods* **2012**, *9*, 357–359.
42. Li, H.; Durbin, R. Fast and accurate short read alignment with burrows–wheeler transform. *Bioinformatics* **2009**, *25*, 1754–1760.
43. Hua, L.; Li, D.G.; Lin, H.; Li, L.; Li, X.; Liu, Z.C. The correlation of gene expression and co-regulated gene patterns in characteristic kegg pathways. *J. Theor. Biol.* **2010**, *266*, 242–249.
44. Rizk, G.; Lavenier, D. Gassst: Global alignment short sequence search tool. *Bioinformatics* **2010**, *26*, 2534–2540.
45. Clement, N.L.; Snell, Q.; Clement, M.J.; Hollenhorst, P.C.; Purwar, J.; Graves, B.J.; Cairns, B.R.; Johnson, W.E. The gnumap algorithm: Unbiased probabilistic mapping of oligonucleotides from next-generation sequencing. *Bioinformatics* **2010**, *26*, 38–45.
46. Lee, W.-P.; Stromberg, M.P.; Ward, A.; Stewart, C.; Garrison, E.P.; Marth, G.T. Mosaik: A hash-based algorithm for accurate next-generation sequencing short-read mapping. *PLoS ONE* **2014**, *9*, e90581.
47. Sedlazeck, F.J.; Rescheneder, P.; von Haeseler, A. NextGenMap: Fast and accurate read mapping in highly polymorphic genomes. *Bioinformatics* **2013**, *29*, 2790–2791.
48. De Bona, F.; Ossowski, S.; Schneeberger, K.; Rätsch, G. Optimal spliced alignments of short sequence reads. *Bioinformatics* **2008**, *24*, i174–i180.
49. Smith, A.D.; Chung, W.-Y.; Hodges, E.; Kendall, J.; Hannon, G.; Hicks, J.; Xuan, Z.; Zhang, M.Q. Updates to the rmap short-read mapping software. *Bioinformatics* **2009**, *25*, 2841–2842.
50. Smith, A.D.; Xuan, Z.; Zhang, M.Q. Using quality scores and longer reads improves accuracy of solexa read mapping. *BMC Bioinforma* **2008**, *9*, 128.
51. Hoffmann, S.; Otto, C.; Kurtz, S.; Sharma, C.M.; Khaitovich, P.; Vogel, J.; Stadler, P.F.; Hackermüller, J. Fast mapping of short sequences with mismatches, insertions and deletions using index structures. *PLoS Comput. Biol.* **2009**, *5*, e1000502.

52. Jiang, H.; Wong, W.H. Seqmap: Mapping massive amount of oligonucleotides to the genome. *Bioinformatics* **2008**, *24*, 2395–2396.
53. Lunter, G.; Goodson, M. Stampy: A statistical algorithm for sensitive and fast mapping of illumina sequence reads. *Genome Res.* **2011**, *21*, 936–939.
54. Schatz, M.C. Cloudburst: Highly sensitive read mapping with mapreduce. *Bioinformatics* **2009**, *25*, 1363–1369.
55. Hormozdiari, F.; Hach, F.; Sahinalp, S.C.; Eichler, E.E.; Alkan, C. Sensitive and fast mapping of di-base encoded reads. *Bioinformatics* **2011**, *27*, 1915–1921.
56. Homer, N.; Merriman, B.; Nelson, S.F. Bfast: An alignment tool for large scale genome resequencing. *PLoS ONE* **2009**, *4*, e7767.
57. Li, H.; Ruan, J.; Durbin, R. Mapping short DNA sequencing reads and calling variants using mapping quality scores. *Genome Res.* **2008**, *18*, 1851–1858.
58. Eaves, H.L.; Gao, Y. Mom: Maximum oligonucleotide mapping. *Bioinformatics* **2009**, *25*, 969–970.
59. Campagna, D.; Albiero, A.; Bilardi, A.; Caniato, E.; Forcato, C.; Manavski, S.; Vitulo, N.; Valle, G. Pass: A program to align short sequences. *Bioinformatics* **2009**, *25*, 967–968.
60. Chen, Y.; Souaiaia, T.; Chen, T. Perm: Efficient mapping of short sequencing reads with periodic full sensitive spaced seeds. *Bioinformatics* **2009**, *25*, 2514–2521.
61. Rumble, S.M.; Lacroute, P.; Dalca, A.V.; Fiume, M.; Sidow, A.; Brudno, M. SHRiMP: Accurate mapping of short color-space reads. *PLoS Comput. Biol.* **2009**, *5*, e1000386.
62. David, M.; Dzamba, M.; Lister, D.; Ilie, L.; Brudno, M. SHRiMP: Sensitive yet practical short read mapping. *Bioinformatics* **2011**, *27*, 1011–1012.
63. Lin, H.; Zhang, Z.; Zhang, M.; Ma, B.; Li, M. ZOOM! Zillions of oligos mapped. *Bioinformatics* **2008**, *24*, 2431–2437.
64. Klus, P.; Lam, S.; Lyberg, D.; Cheung, M.S.; Pullan, G.; McFarlane, I.; Yeo, G.S.; Lam, B.Y. Barracuda—a fast short read sequence aligner using graphics processing units. *BMC Res. Notes* **2012**, *5*, 27.
65. Marco-Sola, S.; Sammeth, M.; Guigó, R.; Ribeca, P. The gem mapper: Fast, accurate and versatile alignment by filtration. *Nature Methods* **2012**, *9*, 1185–1188.
66. Rivals, E.; Salmela, L.; Kiiskinen, P.; Kalsi, P.; Tarhio, J. Mpscan: Fast localisation of multiple reads in genomes. In *Algorithms in bioinformatics*; Springer: Philadelphia, PA, USA, 2009; pp. 246–260.
67. Prezza, N.; Del Fabbro, C.; Vezzi, F.; De Paoli, E.; Policriti, A. Erne-bs5: Aligning bs-treated sequences by multiple hits on a 5-letters alphabet; In Proceedings of the ACM Conference on Bioinformatics, Computational Biology and Biomedicine, New York, NY, USA, 8–10 October 2012; pp. 12–19.
68. Blom, J.; Jakobi, T.; Doppmeier, D.; Jaenicke, S.; Kalinowski, J.; Stoye, J.; Goesmann, A. Exact and complete short-read alignment to microbial genomes using graphics processing unit programming. *Bioinformatics* **2011**, *27*, 1351–1358.
69. Kielbasa, S.M.; Wan, R.; Sato, K.; Horton, P.; Frith, M.C. Adaptive seeds tame genomic sequence comparison. *Genome Res.* **2011**, *21*, 487–493.
70. Lexa, M.; Valle, G. Primex: Rapid identification of oligonucleotide matches in whole genomes. *Bioinformatics* **2003**, *19*, 2486–2488.

71. Ondov, B.D.; Cochran, C.; Landers, M.; Meredith, G.D.; Dudas, M.; Bergman, N.H. An alignment algorithm for bisulfite sequencing using the applied biosystems solid system. *Bioinformatics* **2010**, *26*, 1901–1902.
72. Noé, L.; Gîrdea, M.; Kucherov, G. Designing efficient spaced seeds for solid read mapping. *Adv. Bioinforma.* **2010**, *2010*, 708501.
73. Raczy, C.; Petrovski, R.; Saunders, C.T.; Chorny, I.; Kruglyak, S.; Margulies, E.H.; Chuang, H.-Y.; Källberg, M.; Kumar, S.A.; Liao, A. Isaac: Ultra-fast whole-genome secondary analysis on illumina sequencing platforms. *Bioinformatics* **2013**, *29*, 2041–2043.
74. Weese, D.; Holtgrewe, M.; Reinert, K. Razers 3: Faster, fully sensitive read mapping. *Bioinformatics* **2012**, *28*, 2592–2599.
75. Ning, Z.; Cox, A.J.; Mullikin, J.C. Ssaha: A fast search method for large DNA databases. *Genome Res.* **2001**, *11*, 1725–1729.
76. Okonechnikov, K.; Golosova, O.; Fursov, M. Unipro ugene: A unified bioinformatics toolkit. *Bioinformatics* **2012**, *28*, 1166–1167.
77. Flicek, P.; Birney, E. Sense from sequence reads: Methods for alignment and assembly. *Nature Methods* **2009**, *6*, S6–S12.
78. Needleman, S.B.; Wunsch, C.D. A general method applicable to the search for similarities in the amino acid sequence of two proteins. *J. Mol. Biol.* **1970**, *48*, 443–453.
79. Smith, T.F.; Waterman, M.S. Identification of common molecular subsequences. *J. Mol. Biol.* **1981**, *147*, 195–197.
80. Altschul, S.F.; Gish, W.; Miller, W.; Myers, E.W.; Lipman, D.J. Basic local alignment search tool. *J. Mol. Biol.* **1990**, *215*, 403–410.
81. Blumer, A.; Blumer, J.; Haussler, D.; Ehrenfeucht, A.; Chen, M.-T.; Seiferas, J. The smallest automation recognizing the subwords of a text. *Theor. Computer Sci.* **1985**, *40*, 31–55.
82. Li, H.; Homer, N. A survey of sequence alignment algorithms for next-generation sequencing. *Brief. Bioinforma.* **2010**, *11*, 473–483.
83. Ma, B.; Tromp, J.; Li, M. Patternhunter: Faster and more sensitive homology search. *Bioinformatics* **2002**, *18*, 440–445.
84. Khan, Z.; Bloom, J.S.; Kruglyak, L.; Singh, M. A practical algorithm for finding maximal exact matches in large sequence datasets using sparse suffix arrays. *Bioinformatics* **2009**, *25*, 1609–1616.
85. Abouelhoda, M.I.; Kurtz, S.; Ohlebusch, E. Replacing suffix trees with enhanced suffix arrays. *J. Discret. Algorithms* **2004**, *2*, 53–86.
86. Kurtz, S.; Phillippy, A.; Delcher, A.L.; Smoot, M.; Shumway, M.; Antonescu, C.; Salzberg, S.L. Versatile and open software for comparing large genomes. *Genome Biol.* **2004**, *5*, R12.
87. Höhl, M.; Kurtz, S.; Ohlebusch, E. Efficient multiple genome alignment. *Bioinformatics* **2002**, *18*, S312–S320.
88. Fernandes, F.; Freitas, A.T. Slamem: Efficient retrieval of maximal exact matches using a sampled lcp array. *Bioinformatics* **2014**, *30*, 464–471.
89. Khiste, N.; Ilie, L. E-mem: Efficient computation of maximal exact matches for very large genomes. *Bioinformatics* **2015**, *31*, 509–514.
90. Marke, L. AMAS: A fast tool for alignment manipulation and computing of summary statistics. *PeerJ PrePrints* **2015**, *3*, e1672.

91. Cao, X.; Li, S.C.; Tung, A.K. Indexing DNA Sequences Using q-grams. In *Database Systems for Advanced Applications*; Springer: Beijing, China, 2005; pp. 4–16.
92. Sankoff, D. Edit distance for genome comparison based on non-local operations. In *Combinatorial Pattern Matching*; Springer: Berlin, Germany, 1992; pp. 121–135.
93. Ahmadi, A.; Behm, A.; Honnalli, N.; Li, C.; Weng, L.; Xie, X. Hobbes: Optimized gram-based methods for efficient read alignment. *Nucleic Acids Res.* **2012**, *40*, e41.
94. Treangen, T.J.; Salzberg, S.L. Repetitive DNA and next-generation sequencing: Computational challenges and solutions. *Nature Rev. Genetics* **2012**, *13*, 36–46.
95. Li, R.; Li, Y.; Zheng, H.; Luo, R.; Zhu, H.; Li, Q.; Qian, W.; Ren, Y.; Tian, G.; Li, J. Building the sequence map of the human pan-genome. *Nature Biotechnol.* **2010**, *28*, 57–63.
96. International Cancer Genome Consortium. International network of cancer genome projects. *Nature* **2010**, *464*, 993–998.

© 2015 by the authors; licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution license (<http://creativecommons.org/licenses/by/4.0/>).