applied sciences

MDPI

Article

Reference Mapping Considering Swaps of Adjacent Bases

Youngho Kim [1], Munseong Kang [2], Ju-Hui Jeong [1], Dae Woong Kang [1], Soo Jun Park [3] and Jeong Seop Sim [1,*]

[1] Department of Computer Engineering, Inha University, Incheon 22212, Korea; yhkim85@inha.ac.kr (Y.K.); jngjuhe@inha.edu (J.-H.J.); kdw8219@inha.edu (D.W.K.)
[2] Samsung Electronics, Suwon 16677, Korea; kmsung0102@gmail.com
[3] Electronics and Telecommunications Research Institute, Daejeon 34129, Korea; psj@etri.re.kr
* Correspondence: jssim@inha.ac.kr

Abstract: Since the time of the HGP, research into next-generation sequencing, which can reduce the cost and time of sequence analysis using computer algorithms, has been actively conducted. Mapping is a next-generation sequencing method that identifies sequences by aligning short reads with a reference genome for which sequence information is known. Mapping can be applied to tasks such as SNP calling, motif searches, and gene identification. Research on mapping that utilizes BWT and GPU has been undertaken in order to obtain faster mapping. In this paper, we propose a new mapping algorithm with additional consideration for base swaps. The experimental results demonstrate that when the penalty score for swaps was −1, −2, and −3 in paired-end alignment, for the human whole genome, SOAP3-swap aligned 4667, 2318, and 972 more read pairs, respectively, than SOAP3-dp, and for the drosophila genome, SOAP3-swap aligned 1253, 454, and 129 more read pairs, respectively, than SOAP3-dp. SOAP3-swap has the same functionality as that of SOAP3-dp and also improves the alignment ratio by taking biologically significant swaps into account for the first time.

Keywords: next-generation sequencing; mapping; approximate string matching; swap; GPU

Citation: Kim, Y.; Kang, M.; Jeong, J.-H.; Kang, D.W.; Park, S.J.; Sim, J.S. Reference Mapping Considering Swaps of Adjacent Bases. *Appl. Sci.* **2021**, *11*, 5038. https://doi.org/10.3390/app11115038

Academic Editor: Roger Narayan

Received: 9 May 2021
Accepted: 28 May 2021
Published: 29 May 2021

1. Introduction

Since the time of the Human Genome Project (HGP) [1,2], next-generation sequencing, which can reduce the cost and time of sequence analysis using computer algorithms, has been a very active area of research [3,4]. Mapping is a next-generation sequencing method that identifies sequences by aligning short reads with a reference genome for which sequence information is known. It can be applied to various types of analyses of genetic mutations and genetic polymorphisms, such as single-nucleotide polymorphism (SNP) calling, motif searches, and gene identification [5,6]. Most mapping algorithms consist of an indexing step, in which a data structure for a reference or reads is created, and an alignment step, in which fast mapping is performed using the data structures generated.

In order to enhance mapping speed, studies utilizing various data structures have been conducted. Burrows–Wheeler transform (BWT)-based mapping tools have been developed [7–10]. Among them, Bowtie [7] performs mapping based on the Ferragina and Manzini's algorithm [11] and takes mismatches into account. The Burrows–Wheeler aligner (BWA) [8] first generates a prefix trie for the reference sequence and then performs mapping, taking mismatches and gaps into account with the use of a top-down traversal method. SOAP2 [9] considers mismatches and gaps using a bi-directional BWT search (or 2way-BWT search) [12] and the Smith–Waterman algorithm [13], and it reduces space usage through a sampled suffix array. BWA-MEM [10] improved the performance of BWA using the seed alignments with maximal exact matches (MEM) and the seed extensions with the affine-gap Smith–Waterman algorithm. Meanwhile, tophat2 [14], considering very large deletions, inversions on the same chromosome and translocations involving different chromosomes were introduced.

As the performance of graphics processing units (GPUs) has improved, several mapping studies that utilize this technology have been conducted [15–19]. SOAP3 [15] was developed based on SOAP2, in which the memory access method of the auxiliary data structure was improved with the consideration of the characteristics of Compute Unified Device Architecture (CUDA). Hard patterns referring to the read that causes some of the GPU processors to idle as a result of too many branches in alignment are processed, and a coalesced memory access strategy is employed. Subsequently, SOAP3-dp [16], which is based on SOAP3, improved the alignment ratio using the Smith–Waterman algorithm. BarraCUDA [17] was developed based on BWA, utilizing the texture memory of CUDA and a depth-first search algorithm. CUSHAW [18] uses 64 threads per thread block and a shared memory to improve performance, and it uses a depth-first search algorithm that considers the number of mismatches and the quality score of the mismatched base positions. CUSHAW2-GPU [19] improved upon the CUSHAW2 [20] algorithm by utilizing CUDA, a collaborative calculation of CPU and GPU, seed-based alignment, and a tile-based Smith–Waterman algorithm.

Early mapping tools performed alignment only with exact sequence matches, but later alignment algorithms allowed a limited number of mismatches to search for SNPs. In recent years, mapping tools that consider not only mismatches but also gaps have been developed to facilitate the analysis of various genetic mutations. Approximate string matching algorithms that compare sequences in the presence of errors are used [5–9,12,15–20]. In approximate string matching algorithms, distance functions are used for measuring errors. Typical distance functions include the Hamming distance, edit distances, and weighted edit distances [21]. Studies have also been conducted on the extended edit distance, which takes into consideration swaps that change the positions of two adjacent bases [22–25] (Figure 1). Swaps occur during genetic mutation and replication and are known to be associated with spinal muscular atrophy [26,27].

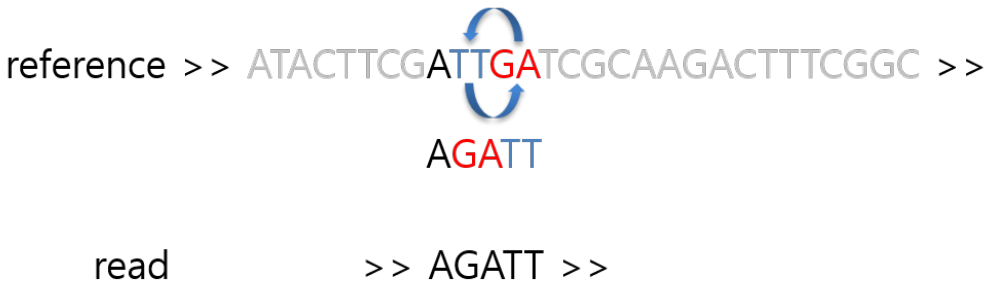

Figure 1. An example of swaps when the swapped block size (the length of the swapped bases) is 2.

In this paper, we propose the SOAP3-swap, which is based on SOAP3-dp and performs reference mapping with additional consideration for swaps. SOAP3-swap can produce alignments in the presence of mismatches, small gaps, and swaps.

Previous Work

SOAP3-dp is a mapping tool that is an extension of SOAP3 [15], the first GPU-based mapping tool. SOAP3 and SOAP3-dp use an improved memory access method and 2way-BWT and allocate reads to different groups according to the amount of calculation required when considering the characteristics of CUDA. SOAP3-dp has a similar execution time to that of SOAP3 but performs an additional alignment of the reads not aligned during the initial attempt using dynamic programming. The addition of this step leads to an enhanced alignment ratio compared to that of SOAP3. While SOAP3 performs alignments that only consider mismatches, SOAP3-dp performs alignments that consider mismatches and also small gaps using the Smith–Waterman algorithm [13], a well-known sequence comparison algorithm.

When a paired-end alignment is carried out using SOAP3-dp, the alignment is first performed through a 2way-BWT search considering only mismatches. For some read pairs that do not align during this process, a search is made for the position with the highest alignment score, which satisfies the interval condition between the pair, taking mismatches and gaps into account. At this point, the region of the reference sequence in which the corresponding read pair may exist is called the candidate region. When a read R ($|R| = n$) likely to be aligned to a candidate region T ($|T| = m$) is provided, the sub-region of T with the highest alignment score between the T's sub-region and R is searched. Let $T[i..j]$ represent the subsequence from i to j in T. Among the subsequences of $T[1..i]$, the alignment score of the subsequence with the highest alignment score for $R[1..j]$ is expressed as $M(i, j)$. $I(i, j)$ refers to the alignment score that inserts $R[j]$ next to $T[1..i]$, and $D(i, j)$ refers to the alignment score that deletes $T[i]$ from $T[1..i]$. Thus, the highest score at which R is aligned to T is $max_{1 \le i \le m} M(i, n)$. The starting position at which R is aligned to T can be calculated by tracing back the errors used to align from $max_{1 \le i \le m} M(i, n)$. The match score was expressed as S_{MA}, and the penalty scores for mismatches, gap openings, and gap extensions were expressed as S_{MI}, S_{GO}, and S_{GE}, respectively. The initial values of each alignment score are as follows: $I(i, 0) = -\infty$ $(1 \le i \le m)$, $D(0, j) = -\infty$ $(1 \le j \le n)$, $M(i, 0) = 0$ $(0 \le i \le m)$, and $M(0, j) = S_{GO} + (j - 1)S_{GE}$ $(1 \le j \le n)$. Algorithm 1 shows the Smith–Waterman algorithm for SOAP3-dp [16]. Here, $\delta(a, b)$ denotes the match or mismatch score between two sequence elements a and b. If $T[i] = R[j]$, then $\delta(T[i], R[j]) = S_{MA}$, and if $T[i] \neq R[j]$, then $\delta(T[i], R[j]) = S_{MI}$.

Algorithm 1 Compute *I*, *D*, and *M*.

1: Assign tables I, D, and M of size $(m + 1) \times (n + 1)$
2: Initialize(I, D, M)
3: **for** $i \leftarrow 1$ **to** m **do**
4: **for** $j \leftarrow 1$ **to** n **do**
5: $I(i, j) \leftarrow max\{M(i, j - 1) + S_{GO}, I(i, j - 1) + S_{GE}\}$
6: $D(i, j) \leftarrow max\{M(i - 1, j) + S_{GO}, D(i - 1, j) + S_{GE}\}$
7: $M(i, j) \leftarrow max\{M(i - 1, j - 1) + \delta(T[i], R[j]), I(i, j), D(i, j)\}$
8: **end for**
9: **end for**

2. Materials and Methods

Consider two sequences, T and R. If $T[(i - 2k + 1)..(i - k)] = R[(j - k + 1)..j]$ and $T[(i - k + 1)..i] = R[(j - 2k + 1)..(j - k)]$ $(i, j \ge 2k)$, we can determine that a swap of length k took place at $T[i]$ and $R[j]$. A swap of length k means that the order of two adjacent subsequences with a length of k is changed. Let S_{SW} be the penalty score for swaps. The alignment score of the swap is calculated by adding S_{SW} to the score before the swap occurred. In Algorithm 2, which computes the tables for SOAP3-swap, the alignment score $M(i, j)$ is defined as the largest score among the alignment scores that considers matches, mismatches, gaps, and swaps (line 9). In SOAP3-swap, swaps of length 1 to 3 can be considered, and the time complexity is the same as that of SOAP3-dp, as shown in Algorithm 2.

The Sequence Alignment/Map (SAM) format is a text format for storing read alignments against reference sequences, and it supports short and long reads (up to 128 Mbp) produced by different sequencing platforms [28,29]. The Binary Alignment/Map (BAM) format is a binary representation of the SAM and keeps exactly the same information as the SAM. To achieve fast random access in a zlib-compatible compressed file [30–33], the BAM can be compressed using the BGZF library, a generic library developed by Handsaker and modified by Li for remote file access and in-memory caching [28]. The SAM/BAM format includes the Compact Idiosyncratic Gapped Alignment Report (CIGAR) string format to describe how a read aligns to a reference. The CIGAR operations in SAM include the following: 'M' for match/mismatch, 'I' for insertion compared with the reference, 'D'

for deletion, 'N' for skipped bases on the reference, 'S' for soft clipping where the clipped subsequence is shown in the read sequence in the SAM format, 'H' for hard clipping where the clipped subsequence is not shown in the read sequence in the SAM format, 'P' for padding, '=' for match, and 'X' for mismatch [28,29].

Algorithm 2 Compute I, D, and M considering swaps.

Assign tables I, D, and M of size $(m+1) \times (n+1)$
Initialize(I, D, M)
for $i \leftarrow 1$ **to** m **do**
    for $j \leftarrow 1$ **to** n **do**
        $I(i,j) \leftarrow max\{M(i,j-1) + S_{GO}, I(i,j-1) + S_{GE}\}$
        $D(i,j) \leftarrow max\{M(i-1,j) + S_{GO}, D(i-1,j) + S_{GE}\}$
        $M(i,j) \leftarrow max\{M(i-1,j-1) + \delta(T[i], R[j]), I(i,j), D(i,j)\}$
        if $(i \geq 2k)$ and $(j \geq 2k)$ and $T[(i-2k+1)..(i-k)] = R[(j-k+1)..j]$ and $T[(i-k+1)..i] = R[(j-2k+1)..(j-k)]$ **then**
            $M(i,j) \leftarrow max\{M(i,j), M(i-2k, j-2k) + S_{SW}\}$
        end if
    end for
end for

Since a mapping tool that considers swaps was not available, no alignment information regarding swaps was available in the CIGAR string. Therefore, the symbols representing the swap were added to the CIGAR string of SOAP3-swap. When outputting the alignment results in SAM format, the reads where a swap occurred are marked as 'T', 'W', and 'B' in the CIGAR string when the lengths of the two exchanged base sequences are 1, 2, and 3, respectively.

3. Results

The data used in these experiments were the human genome sequence and the drosophila genome sequence. The human genome sequence is GRCh38 (3.3 GB), and 100 bp-paired-end reads (SRR211279, 8.7 GB each) containing 25,467,888 reads in total. The reads were generated by Illumina GAIIx from the Washington University Genome Sequencing Center. The drosophila genome sequence is DhydRS2 (148.5 MB, Drosophila hydei), and 100 bp-paired-end reads (SRR6326389, 12.7 GB each) containing 38,873,031 reads in total. The reads were generated by Illumina HiSeq 2500 from Texas A&M University.

In order to examine the performance and effectiveness of SOAP3-swap, the algorithm was compared to SOAP3-dp, SOAP3, CUSHAW2-GPU, BarraCUDA, BWA, Bowtie2, and CUSHAW2. The experimental conditions were as follows:

- CPU and RAM: AMD Ryzen 9 3950X (3.5 GHz), 64 GB RAM (2666 MHz);
- OS: Fedora 27 (64 bit);
- GPU: GeForce RTX 2080 Ti (11 GB memory);
- Development tools and language: C++ (GCC 7.4.0), CUDA (SDK 9.1).

SOAP3-swap, SOAP3-dp, CUSHAW2-GPU, and BarraCUDA were tested using 32 CPU threads and one GPU device. SOAP3 was tested using six CPU threads, which were maximal and one GPU device. BWA, Bowtie2, and CUSHAW2 were tested using 32 CPU threads. In SOAP3-swap and SOAP3-dp, S_{MA} was set to 1, and S_{MI}, S_{GO}, and S_{GE} were set to -2, -3, and -1, respectively, as in [16]. In SOAP3-swap, S_{SW} was tested at -1, -2, and -3. Users can adjust the penalty scores to determine which operation is more likely to occur. The average running time of 20 experiments was taken to represent the running time of each tool. Tables 1 and 2 show the running times and alignment ratios of the paired-end alignments produced by the various tools for the human genome and for the drosophila genome, respectively.

Table 3 shows the number of read pairs including at least one swap according to the swap cost and the swapped block size for the human genome and the drosophila genome.

Table 1. Comparison of SOAP3-swap and other tools for the human genome.

Tools	Paired-End Reads			GPU-Based
	Runtime (s)	Number of Aligned Reads (Ratio)	Difference from SOAP3-dp	
SOAP3-swap (Full SA, $S_{SW} = -1$)	186.98	25,105,846 (98.58%)	+4667	Yes
SOAP3-swap (Full SA, $S_{SW} = -2$)	178.79	25,103,497 (98.57%)	+2318	Yes
SOAP3-swap (Full SA, $S_{SW} = -3$)	181.08	25,102,151 (98.56%)	+972	Yes
SOAP3-dp (Full SA)	178.52	25,101,179 (98.56%)	.	Yes
SOAP3	212.22	22,613,051 (88.79%)	−2,488,128	Yes
CUSHAW2-GPU	599.94	24,957,932 (98.00%)	−143,247	Yes
BarraCUDA	947.67	24,551,512 (96.40%)	−549,667	Yes
BWA	2731.73	24,598,492 (96.59%)	−502,687	No
Bowtie2 (Sensitive)	1414.67	24,798,944 (97.37%)	−302,235	No
CUSHAW2	2084.69	24,133,013 (94.76%)	−968,166	No

Table 2. Comparison of SOAP3-swap and other tools for the drosophila genome.

Tools	Paired-End Reads			GPU-Based
	Runtime (s)	Number of Aligned Reads (Ratio)	Difference from SOAP3-dp	
SOAP3-swap (Full SA, $S_{SW} = -1$)	641.47	31,514,424 (81.07%)	+1253	Yes
SOAP3-swap (Full SA, $S_{SW} = -2$)	624.95	31,513,625 (81.07%)	+454	Yes
SOAP3-swap (Full SA, $S_{SW} = -3$)	623.67	31,513,300 (81.07%)	+129	Yes
SOAP3-dp (Full SA)	563.83	31,513,171 (81.07%)	.	Yes
SOAP3	698.08	24,409,682 (62.79%)	−7,103,489	Yes
BarraCUDA	1131.08	30,380,166 (78.15%)	−1,133,005	Yes
Bowtie2 (Sensitive)	1261.55	30,602,667 (78.72%)	−910,504	No
CUSHAW2	2697.97	30,099,066 (77.43%)	−1,414,105	No

Table 3. Number of read pairs including at least one swap according to the swap cost and the swapped block size among all the read pairs aligned by SOAP3-swap.

	Swap Cost	Swapped Block Size			
		1	**2**	**3**	**Total**
Human genome	$S_{SW} = -1$	265,727	7631	21	273,379
	$S_{SW} = -2$	202,850	117	31	202,998
	$S_{SW} = -3$	145,054	57	6	145,117
Drosophila genome	$S_{SW} = -1$	304,209	13,919	215	318,343
	$S_{SW} = -2$	212,278	627	273	213,178
	$S_{SW} = -3$	158,953	79	72	159,104

4. Discussion

The two main strengths of SOAP3-swap are its speed and its alignment ratio. As shown in Tables 1 and 2, except for SOAP3-dp, SOAP3-swap was significantly faster than the other tools in our experiments. With respect to alignment ratio, SOAP3-swap aligned more read pairs than any of the other reference mapping tools. For example, when S_{SW} was set to -1, -2, and -3, for the human genome, SOAP3-swap aligned 4667 more read pairs, 2318 more read pairs, and 972 more read pairs, respectively, than SOAP3-dp, and for the drosophila genome, SOAP3-swap aligned 1253 more read pairs, 454 more read pairs, and 129 more read pairs, respectively, than SOAP3-dp. Since the alignment results can be the basis for reconstructing genomes and identifying mutations of genomes, the higher alignment ratio can be more helpful to analyze genomes.

When S_{SW} was set to -1, -2, and -3, for the human genome, SOAP3-swap aligned $273,379$, $202,998$, and $145,117$ read pairs, respectively, and for the drosophila genome, SOAP3-swap aligned $318,343$, $213,178$, and $159,104$ read pairs, respectively, as shown in Table 3. Since we performed the paired-end alignment in our experiment, we considered it as a swap occurrence when a swap occurred in at least one read. This result shows that some of the read pairs aligned by SOAP3-dp may actually have been aligned by swaps rather than insertion/deletion. Therefore SOAP3-swap can provide a different alignment result compared to SOAP3-dp because it performs alignment by considering all of the mismatches, gaps, and swaps.

5. Conclusions

The SOAP3-swap proposed in this paper has the same functionality as that of SOAP3-dp and also improves the alignment ratio by taking biologically significant swaps into account for the first time. The reason why we considered the swaps with lengths of up to three is because swaps with lengths longer than three hardly occur in our experiment.

Further research is necessary to develop more efficient alignment methods that can be applied to situations including exchanges involving non-adjacent bases.

Author Contributions: Y.K. and M.K. are the main developers and analyzed SOAP3-dp and implemented SOAP3-swap. J.-H.J. and D.W.K. analyzed SOAP2 and SOAP3 and designed the parallel technique together. S.J.P. provided biological support and designed the experiments. J.S.S. provided algorithmic and biological support and is the manager in charge of the project. All authors read and approved the final manuscript.

Funding: This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korean Government (MSIP) (No. 2021R1H1A2011633); by the Genome Program for Fostering New Post-Genome Industry of the National Research Foundation (NRF) funded by the Korean Government (MSIP) (NRF-2014M3C9A3064706); by the Institute of Information & Communications Technology Planning & Evaluation (IITP) grant funded by the Korean Government (MSIT)

(2020-0-01389, Artificial Intelligence Convergence Research Center (Inha University)); and by INHA UNIVERSITY Research Grant.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The software SOAP3-swap is available for download at https://github.com/yhkim8505/SOAP3-swap (accessed on 28 May 2021).

Acknowledgments: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

Sample Availability: For the human genome, the reference sequence, GRCh38, is available at: http://ftp.ncbi.nlm.nih.gov/genomes/all/GCA/000/001/405/GCA_000001405.15_GRCh38/seqs_for_alignment_pipelines.ucsc_ids/GCA_000001405.15_GRCh38_full_analysis_set.fna.gz (accessed on 28th May 2021). The read sequences and source codes used in the manuscript are available from the corresponding author on reasonable request. For the drosophila genome, the reference sequence, DhydRS2, is available at: https://www.ncbi.nlm.nih.gov/assembly/GCF_003285905.1/ (accessed on 28th May 2021). The read sequences are available at: https://trace.ncbi.nlm.nih.gov/Traces/sra/?run=SRR6326389 (accessed on 28th May 2021).

Abbreviations

The following abbreviations are used in this manuscript:

HGP	Human Genome Project
SNP	Single-nucleotide polymorphism
BWT	Burrows–Wheeler transform
BWA	Burrows–Wheeler aligner
GPU	Graphics processing unit
CUDA	Compute unified device architecture
CIGAR	Compact idiosyncratic gapped alignment report

References

1. Tilford, C.A.; Kuroda-Kawaguchi, T.; Skaletsky, H.; Rozen, S.; Brown, L.G.; Rosenberg, M.; McPherson, J.D.; Wylie, K.; Sekhon, M.; Kucaba, T.A.; et al. A physical map of the human Y chromosome. *Nature* **2001**, *409*, 943–945. [CrossRef]
2. International Human Genome Sequencing Consortium. Initial sequencing and analysis of the human genome. *Nature* **2001**, *409*, 860–921.
3. Metzker, M.L. Sequencing technologies—the next generation. *Nat. Rev. Genet.* **2010**, *11*, 31–46. [CrossRef]
4. Bao, S.; Jiang, R.; Kwan, W.; Wang, B.; Ma, X.; Song, Y.Q. Evaluation of next-generation sequencing software in mapping and assembly. *J. Hum. Genet.* **2011**, *56*, 406–414. [CrossRef]
5. 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. [CrossRef]
6. Li, H.; Ruan, J.; Durbin, R. Mapping short DNA sequencing reads and calling variants using mapping quality scores. *Genome Res.* **2008**, *18*, 1851–1858. [CrossRef]
7. 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. [CrossRef] [PubMed]
8. Li, H.; Durbin, R. Fast and accurate short read alignment with Burrows–Wheeler transform. *Bioinformatics* **2009**, *25*, 1754–1760. [CrossRef]
9. Li, R.; Yu, C.; Li, Y.; Lam, T.W.; Yiu, S.; Kristiansen, K.; Wang, J. SOAP2: An improved ultrafast tool for short read alignment. *Bioinformatics* **2009**, *25*, 1966–1967. [CrossRef] [PubMed]
10. Li, H. Aligning sequence reads, clone sequences and assembly contigs with BWA-MEM. *arXiv* **2013**, arXiv:1303.3997.
11. Ferragina, P.; Manzini, G. Opportunistic Data Structures with Applications. In Proceedings of the 41st Annual Symposium on Foundations of Computer Science, FOCS 2000, Redondo Beach, CA, USA, 12–14 November 2000; pp. 390–398. [CrossRef]
12. Lam, T.W.; Li, R.; Tam, A.; Wong, S.C.K.; Wu, E.; Yiu, S. High Throughput Short Read Alignment via Bi-directional BWT. In Proceedings of the 2009 IEEE International Conference on Bioinformatics and Biomedicine, BIBM 2009, Washington, DC, USA, 1–4 November 2009; pp. 31–36. [CrossRef]
13. Smith, T.F.; Waterman, M.S. Identification of Common Molecular Subsequences. *J. Mol. Biol.* **1981**, *147*, 195–197. [CrossRef]
14. Kim, D.; Pertea, G.; Trapnell, C.; Pimentel, H.; Kelley, R.; Salzberg, S.L. TopHat2: Accurate alignment of transcriptomes in the presence of insertions, deletions and gene fusions. *Genome Biol.* **2013**, *14*, 1–13. [CrossRef]

15. Liu, C.; Wong, T.K.F.; Wu, E.; Luo, R.; Yiu, S.; Li, Y.; Wang, B.; Yu, C.; Chu, X.; Zhao, K.; et al. SOAP3: Ultra-fast GPU-based parallel alignment tool for short reads. *Bioinformatics* **2012**, *28*, 878–879. [CrossRef]
16. 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. Correction: SOAP3-dp: Fast, accurate and sensitive GPU-based short read aligner. *PLoS ONE* **2013**, *8*. [CrossRef]
17. 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. [CrossRef]
18. 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. [CrossRef] [PubMed]
19. Liu, Y.; Schmidt, B. CUSHAW2-GPU: Empowering Faster Gapped Short-Read Alignment Using GPU Computing. *IEEE Des. Test* **2014**, *31*, 31–39. [CrossRef]
20. Liu, Y.; Schmidt, B. Long read alignment based on maximal exact match seeds. *Bioinformatics* **2012**, *28*, 318–324. [CrossRef] [PubMed]
21. Gusfield, D. *Algorithms on Strings, Trees, and Sequences—Computer Science and Computational Biology*; Cambridge University Press: New York, NY, USA , 1997.
22. Lowrance, R.; Wagner, R.A. An Extension of the String-to-String Correction Problem. *J. ACM* **1975**, *22*, 177–183. [CrossRef]
23. Wagner, R.A. On the Complexity of the Extended String-to-String Correction Problem. In Proceedings of the 7th Annual ACM Symposium on Theory of Computing, Albuquerque, NM, USA, 5–7 May 1975; pp. 218–223. [CrossRef]
24. Kim, D.K.; Lee, J.; Park, K.; Cho, Y. Efficient Algorithms for Approximate String Matching with Swaps. *J. Complex.* **1999**, *15*, 128–147. [CrossRef]
25. Kang, D.W.; Kim, Y.; Sim, J.S. Parallel Computation for Extended Edit Distances Including Swap Operations. *J. KIISE Comput. Syst. Theory* **2014**, *41*, 175–181.
26. Lewin, B. Genes for SMA: Multum in parvo. *Cell* **1995**, *80*, 1–5. [CrossRef]
27. Amir, A.; Aumann, Y.; Landau, G.M.; Lewenstein, M.; Lewenstein, N. Pattern Matching with Swaps. *J. Algorithms* **2000**, *37*, 247–266. [CrossRef]
28. Li, H.; Handsaker, B.; Wysoker, A.; Fennell, T.; Ruan, J.; Homer, N.; Marth, G.T.; Abecasis, G.R.; Durbin, R. The Sequence Alignment/Map format and SAMtools. *Bioinformatics* **2009**, *25*, 2078–2079. [CrossRef] [PubMed]
29. The SAM/BAM Format Specification Working Group. Sequence Alignment/Map Format Specification . Available online: https://samtools.github.io/hts-specs/SAMv1.pdf (accessed on 28 May 2021).
30. Roelofs, G.; Gailly, J.L.; Adler, M. zlib. Available online: https://zlib.net/ (accessed on 28 May 2021).
31. Deutsch, L.P.; Gailly, J.L. ZLIB Compressed Data Format Specification Version 3.3. Available online: https://datatracker.ietf.org/doc/html/rfc1950 (accessed on 28 May 2021).
32. Deutsch, L.P. DEFLATE Compressed Data Format Specification Version 1.3. Available online: https://datatracker.ietf.org/doc/html/rfc1951 (accessed on 28 May 2021).
33. Deutsch, L.P. GZIP file Format Specification Version 4.3. Available online: https://datatracker.ietf.org/doc/html/rfc1952 (accessed on 28 May 2021).