Review of Recent NGS Short Reads Alignment Tools

BMI-231 final project, Chenxi Chen

Spring 2014

Deciphering the information contained in DNA sequences began decades ago since the time of Sanger sequencing. The development of next generation sequencing makes it possible to study organisms on the genome scale. NGS have been used in various types of studies including genome re-sequencing (DNA-seq), DNA-protein interactions (ChIP-seq), transcriptom reconstruction, quantitative analysis (RNA-seq) and so on. As the improved efficiency of sequencing technologies such as PacBio RS II, Ion Torrent proton and Illumina HiSeq X Ten, sequencing price has dramatically dropped to as low as $1,000 for sequencing a human genome, which facilitates the rapid raising of data generating. Correct data interpretation is urgently needed. Aligning short reads to reference genome with sufficient quality is a prerequisite for many comparative genomic studies. Variant types of software are available for short reads alignment. This review will focus on discussing the most commonly used aligners in recent 2-3 years.

Bowtie and Bowtie2

Indexing reference genome can efficiently speed up finding candidate alignment location(s) for each read. Bowtie[1] uses the Burrows-Wheeler transform (BWT) and the full-text minute index based scheme to index reference genome as shown in Figure1. Figure 1(a) and 1(b) shows the BWT using a dummy string and how to recover the original string by applying last first (LF) mapping repeatedly. 1(c) illustrates searching a string in FM index using a common method, EXACTMATCH, which identifies the range of rows first and then the exact query sequence.

Figure1. Burrow-Wheeler Transform. Cited from Genome Biol, 2009. **10**(3): p. R25. [1]

To make bowtie a suitable algorithm for mapping short reads, which have mismatches due to genetic variants and sequencing error, a greedy backtracking

search algorithm was induced, as shown in Figure2. Instead of aborting when encountering an empty range as EXACTMATCH, bowtie can backtrack to find a non-empty range one or an alignment with more occurrences to report. When a read with low sequencing quality and could not map or maps poorly to the reference sequence, bowtie will spend most its effort backtracking, which dramatically decrease the alignment efficiency. Double indexing, BWT of the genome (forward index) and reverse genome (mirror index), can help overcome excessive backtracking. Limited number of backtracking option (default: 125) is also adopted to force stop over backtracking when the reads have very poor quality and multiple mismatch alignment is allowed.

Figure2. EXACTMATCH (top) and backtrack (bottom) searching algorithm. Exact match will abort if there is no 'ggta' in the reference genome. However, inexact match will look for a nonempty one. Cited from Langmead, B., et al, Genome Biol, 2009. **10**(3): p. R25. [1]

Bowtie2 is an improved algorithm based on Bowtie to support gapped alignment and pair-end reads [2]. It contains two stages: un-gapped seed-finding stage and gapped extension stage. Figure3 shows the 4-steps alignment strategy for reads alignment: 1. Extract seed from each read and its reverse sequence. 2. Align each seed to reference genome using un-gapped FM-index way. 3. Prioritize seed alignments. 4. Extend seed to full reads alignment. The adoption of seeds-extension method makes Bowtie2 more efficient in terms of shorter running time and more aligned reads without losing sensitivity.

Gapped alignment method allows Bowtie2 to identify genetic variants such as insertion and deletion. It has been used as the core engine to align transcriptom reads onto a reference genome by Tophat 2 [3]. Reads spanning multiple exons are unable to be aligned directly to reference genome. These unmapped reads are spliced into shorter non-overlapped segments and re-aligned to genome. Left and right segments derived from same reads separated apart within defined maximum

intron size are considered locate at splicing sites. Tophat 2 can also consider pair-end reads as evidence in identifying splicing sites and fusion break points. Tophat is a good choice for transcriptom studies because of the alternative splicing events.

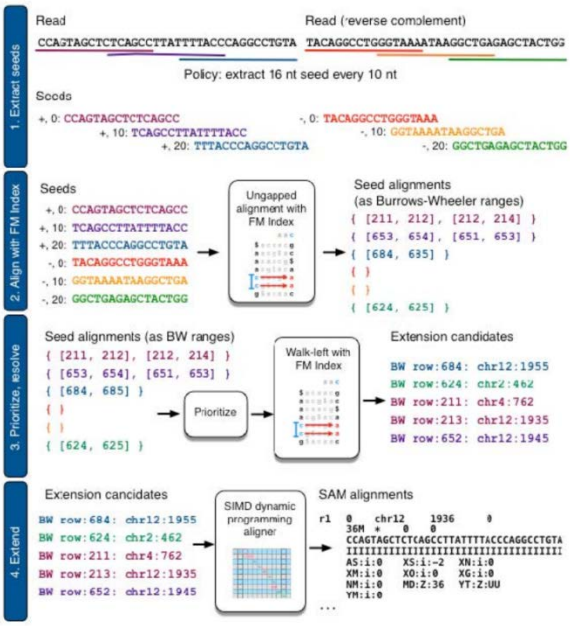

Figure3. Bowtie2 alignment workflow. Cited from Nat Methods, 2012. **9**(4): p. 357-9. [2]

CUSHAW2 and CUSHAW3

CUSHAW2 [4] published in 2012 was among the so called "second generation alignment tools" that adopt seed-and-extend heuristic to apply gap alignment. Interestingly, its predecessor CUSHAW [5] was GPU-based while CUSHAW2 is CPU-based, and author did not mention reason in the article. Considering the massive memory access requirement of Dynamic Programming and the natural memory limitation of GPU, the adoption of "seed-and-extend" strategy may be the main reason for platform transition.

The first step of seed-and-extend mapping is to generate seeds which are short matches indicating highly similar regions. There are multiple types of seeds been proposed: fixed-length seeds (k-mers), maximal exact matches (MEMs), maximal unique matches (MUMs) and adaptive seeds. CUSHAW2 uses MEMs which are the longest matches that with no mismatches.

After 2 years, CUSHAW3 was developed [6], aiming to improve the alignment sensitivity and accuracy, at cost of speed. It was observed by the author that all the "second generation aligners" still have difficulties in aligning all short reads correctly to large reference such as human genome, which was probably due to the trade off to get acceptable speed. However, with the development and deployment of high performance computational resources for high-throughput data, a new balance is

expected towards the mapping sensitivity and accuracy. To fulfill this purpose, in additional of one mechanism to extend seeds, CUSHAW3 serially uses 3 different seed types (See Figure4 for workflow): MEM seeds, exact-match k-mer seeds, and variable-length seeds. More specifically, CUSHAW3 firstly conducts a regular MEM seeds extend process just like CUSHAW2. Then if the alignment standard is not met, CUSHAW3 will attempt to rescue the read and use the optimal local alignment as a variable-length seed, and then perform semi-global alignment identification. Lastly, if still no qualified alignment is found, the MEM seeds will be abandoned and replaced by a new generated exact-match k-mer seeds. Notably, only non-overlapping seeds are used here to improve speed. After another round of Dynamic Programming extension using this new seed, the unmapped reads can be finally claimed.

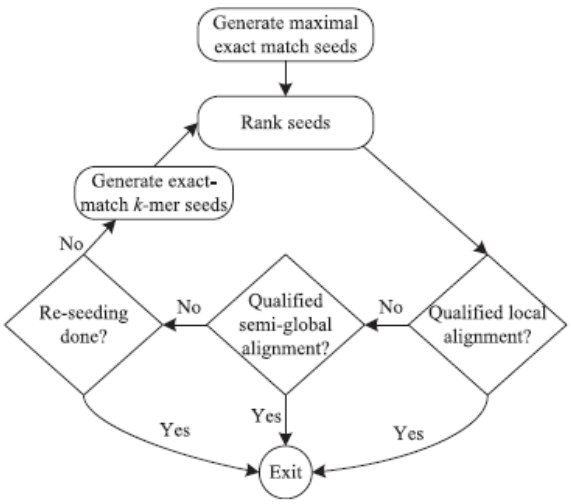

Figure4. workflow of sigle-end alignment using hybrid seeding. Cited from PLoS One, 2014. **9**(1): p. e86869. [6]

The GEM Mapper

The Genome Multitool (GEM) mapper was published by a Spain group in 2012 [7]. GEM mapper relies on Burrows-Wheeler transform (BWT), specifically FM indexing, just like some other mappers including bowtie. However, instead of adopting the popular “seed-and-extension” strategy, GEM focuses on the combination of two ideas: “filtering-based approximate string matching” and “region-based adaptive filtering” which is a method to determine filtering segments.

The limitation of the “seed-and-extension methods”, although with fast speed, is it’s inflexible and incapable of returning all the possible existing matches. Missing part of possible matches may not be very important issue before because it was regarded as proper tradeoff for limited computation resources, but current biological problems such as researches on non-model-organism which does not have a complete reference for mapping, or comparisons inter-species that naturally requires the higher tolerance to errors and mismatches. In other cases, such as Bisulfit-seq or metagenomics, exhaustive searching of every possible match is necessary, while

seed-and-extend alignment is not sufficient. In other words, non-exhaustive algorithms analyze more complicated paths and prune off some and skip portions of the target searches, and thus they cannot make certain statements for the number of matches in a target distance.

In GEM, firstly each query sequence is divided into different segments, instead of simply equally sized segment, GEM dividing sequences into segments with similar match numbers in genome. Purpose of doing this is to prevent segment with too many matches in genome that requires too much computation resources. In the next step, all the matches of every segment up to some distance are retrieved. Then after verifying segment match against the entire query, the non-redundant matches are reported.

Although using exhaustive matching mode, GEM is faster than most of its competitors probably due to the smart segment dividing mechanism, especially in a more mismatch environment.

SOAP3 and SOAP3-dp

SOAP2 is a well-used short reads alignment tool especially for Illumina reads un-gapped alignment [8]. In 2012, its successor, SOAP3, was published, and successfully leveraged the computation power of Graphic Processing Unit (GPU) to achieve the significant improvement in speed [9].

Compare to CPU, GPU provides massive parallelism as low-cost hardware, but the down side is the limited memory and restricted usage. In the high-throughput short reads alignment application, a number of GPU-based tools was emerged in 2012, including CUSHW [5] and BarraCUDA [10] which implemented BWA to align reads in parallel using GPU. However, due to the different working environments of GPU and CPU, simply transferring working platform of aligners will not automatically lead to performance improvement. Taking BarraCUDA as example, it works sub-optimally on GPU and only showed 4-time boost to a single-thread BWA, because of the limited branch and bound trie algorithm. To solve this problem and optimized for the limited memory access on GPU, authors of SOAP3 redesigned the data structure to reduce requirement of memory. Another difference in GPU working environment is that processors in the same unit must execute the exact same instruction in the same time. While the nature of BWT includes many diverging branches and thus will always make some processors stay idle. Solution provided by SOAP3 is simple, as a useful parameter was introduced to predict whether too many branches will be generated in a path and the positive paths will be stopped before actually running. In general SOAP3 is an optimized GPU-based version of SOAP2.

Releasing the parallel computation capacity of GPU did greatly improved speed in mapping: Although SOAP2 and SOAP3 did not show many differences in mapping

sensitivity or data loading time. But actual alignment timing of SOAP3 showed an average 10 times improvement compared to SOAP2 using real testing data [SOAP3: ultra-fast GPU-based parallel alignment tool for short reads].

SOAP3 was still limited in gapped alignment, and SOAP3-dp [11] was developed to address that issue. Usually the "seed-and-extend" process and "dynamic programming" (DP) strategy are needed to conduct the detailed gap mapping of reads to the target region. GPU-based environment is especially preferred for this DP process, because seeds can often be mapped to multiple locations and required parallel computation to process them all at the same time. SOAP3-dp runs three steps in mapping (See Figure5 for workflow) : 1) Use 2 way-BWT to align paired-end reads, just like what SOAP3 did; 2) For reads with only one end mapped, use DP to align the unmapped ends in the target region calculated by sequencing insertion length; 3) For all other reads, use 2 way-BWT to locate seeds first then perform DP to mapping.

Another impressive fact of SOAP3-bp is the fact that it has been successfully deployed and computing-cloud environment such as Amazon EC2, NIH BioWulf and Tianhe-1A, which will be discussed in detail later.

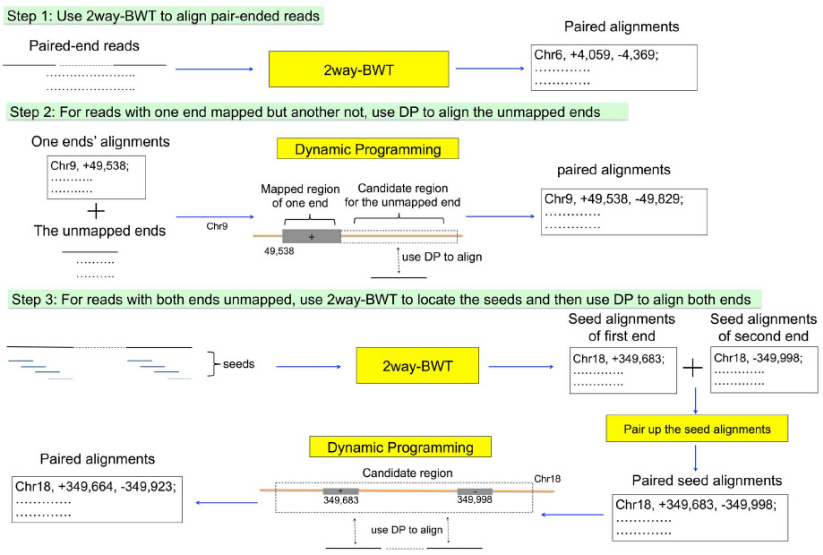

Figure5. SOAP3 alignment workflow. Cited from PLoS One, 2013. **8**(5): p. e65632. [11]

Performance Comparison and discussion

Most performance comparisons between different aligners focus on alignment time and sensitivity. It should be noted the memory consumption in short reads mapping application is far less than *de novo* assembly and is not issue for most of high performance computational machines. Alignment time is an important limitation for users to process large batch of samples or try different parameters. So in this comparison we will focus on the alignment timing on real data and the sensitivity on

simulated data where the true alignment is already known. Performance data was mainly extracted from two the publication of SOAP3-dp and CUSHAW3 and calibrated based on the one of the common participants: Bowtie2.

Both SOAP3-dp and CUSHAW3 paper generated multiple simulated data for testing, including different read length, single or paired end read, and different error rate. To make two datasets comparable, we chose sensitivity data from 100bp paired-end simulated data. If Bowtie2 is used as standard reference, we can see from the table below that 1) SOAP3 did the worst in sensitivity but SOAP3-dp improved greatly and made itself among the top of all; 2) CUSHAW2 and GEM perform better than Bowtie2; 3) CUSHAW3 did slightly improved from CUSHAW2, but not reach SOAP3-dp's sensitivity.

Sensitivity test on simulated data

Data from SOAP3-dp paper					
100PE	SOAP3-dp	SOAP3	Bowtie2	GEM	CUSHAW2
Sensitivity	99.66%	97.77%	98.82%	99.53%	99.17%
Data from CUSHAW3 paper					
100PE 2% error	CUSHAW3	CUSHAW2	Bowtie2	GEM	
Sensitivity	99.54%	99.43%	98.53%	99.20%	

Table modified from PLoS One, 2014. **9**(1): p. e86869[6] and PLoS One, 2013. **8**(5): p. e65632.[11]

For the camparison of alignment time of real data, 3 datasets from each publication was chosen and calibrate using Bowtie2 as 100% standard. All three datasets were paired-end reads at length of 100bp or 150bp, reflecting the latest Illumina sequencer output. From the 6 groups of comparison we can see: 1) CUSHAW2 is slightly faster than Bowtie2; 2) GEM is about 2 times faster than Bowtie2; 3) The two GPU-based aligners SOAP3 and SOAP3-dp runs about 10 times faster than Bowtie2; 4) CUSHAW3 is 3-5 times slower than Bowtie2.

Alignment timing on real data

Data from SOAP3-dp paper					
	SOAP3-dp	SOAP3	Bowtie2	GEM	CUSHAW2
PE100(5.07GB)	8.3%	8.8%	100.0%	34.3%	100.3%
PE100(12.24GB)	8.3%	21.8%	100.0%	29.4%	100.4%
PE150(56.23GB)	12.8%	8.2%	100.0%	48.1%	102.8%
Data from CUSHAW3 paper					
	CUSHAW3	CUSHAW2	Bowtie2	GEM	
PE100 (7.24G)	523.1%	68.9%	100.0%	61.5%	
PE100 (10.18G)	457.4%	70.5%	100.0%	64.3%	
PE100 (10.73G)	363.3%	69.5%	100.0%	70.2%	

Table modified from PLoS One, 2014. **9**(1): p. e86869[6] and PLoS One, 2013. **8**(5): p. e65632.[11]

It's not fair to compare directly the running time between CPU-based aligners and

GPU-based aligners because they were using different hardware platforms. In the comparison, the top performance desktop CPU I7-3930k and top performance desktop graphic card GTX 680 was used. If we consider the price for the CPU-based aligners, we will need around $300 for a current version of top desktop CPU, and for the GPU-based aligners we will need $500 for a top graphic card and additional $100 for a low-end CPU such as Intel I3. In general, without considering other costs we can spend about double price on hardware to achieve 10 times speed. Therefore, it is expected to see more aligners transfer to GPU computing platforms.

Another trend indicated by SOAP3-dp is the cloud platform, where data and conduct computation process can be stored. The concept of transferring the computing and data storage to cloud platform attracts most users because it's a simple and low-cost solution for most biological scientists who do not have the skills or finical capacity to build and maintain a private Linux cluster. Public cloud faces several challenges in other fields including data security and data uploading/downloading, but not in short reads mapping. Data security can be assured by keep annotation or phenotype data off line, since sequences itself contains very limited meaningful information without proper interpretation. And the latest model of Illumina sequencers are connected with internet and can be set as sequencing while uploading, so data will be ready on cloud platform right after sequencing is finished.

Conclusion

Great progress has been made on improving short reads aligner tools to be able to process reads with increased sequencing depth and length using much lower computational cost. Given the fact that genetic variants and alternative splicing events exist in eukaryotes genomes, algorithms allowing mismatches and gaps is a must for accurate alignment for downstream analysis. Challenges still remain for developing new mapping algorithms to satisfy the rapid changes of NGS technologies. GPU-based computing and cloud computing are new trends.

References:

1. Langmead, B., et al., *Ultrafast and memory-efficient alignment of short DNA sequences to the human genome.* Genome Biol, 2009. **10**(3): p. R25.
2. Langmead, B. and S.L. Salzberg, *Fast gapped-read alignment with Bowtie 2.* Nat Methods, 2012. **9**(4): p. 357-9.
3. Kim, D., et al., *TopHat2: accurate alignment of transcriptomes in the presence of insertions, deletions and gene fusions.* Genome Biol, 2013. **14**(4): p. R36.
4. Liu, Y. and B. Schmidt, *Long read alignment based on maximal exact match seeds.* Bioinformatics, 2012. **28**(18): p. i318-i324.
5. Liu, Y., B. Schmidt, and D.L. Maskell, *CUSHAW: a CUDA compatible short read aligner to large genomes based on the Burrows-Wheeler transform.* Bioinformatics, 2012. **28**(14): p. 1830-7.
6. Liu, Y., B. Popp, and B. Schmidt, *CUSHAW3: sensitive and accurate base-space and color-space short-read alignment with hybrid seeding.* PLoS One, 2014. **9**(1): p. e86869.
7. Marco-Sola, S., et al., *The GEM mapper: fast, accurate and versatile alignment by filtration.* Nat Methods, 2012. **9**(12): p. 1185-8.
8. Li, R., et al., *SOAP2: an improved ultrafast tool for short read alignment.* Bioinformatics, 2009. **25**(15): p. 1966-7.
9. Liu, C.M., et al., *SOAP3: ultra-fast GPU-based parallel alignment tool for short reads.* Bioinformatics, 2012. **28**(6): p. 878-9.
10. Klus, P., et al., *BarraCUDA - a fast short read sequence aligner using graphics processing units.* BMC Res Notes, 2012. **5**: p. 27.
11. Luo, R., et al., *SOAP3-dp: fast, accurate and sensitive GPU-based short read aligner.* PLoS One, 2013. **8**(5): p. e65632.