Lecture 9

Short read aligners

Reading assignments

- Alignment of Next-Generation Sequencing Reads (https://www.ncbi.nlm.nih.gov/pubmed/25939052)

Overview

- Long reads
  - ~1,000bp (Sanger sequencing) or > 10kb (Long-read sequencing)
  - small number of reads
  - aligner designed to perform near-optimal alignments and to find most (if not all) alternative alignments.

- Short reads
  - 50-300bp
  - Extremely large number of reads by NGS
  - aligner designed to quickly select the best location of each read at the expense of not investigating all potential alternative alignments.

Overview

- Sequencing reads are treated as small, independent measurements to be subsequently matched either 1) against a known genome (resequencing) or 2) against one another (de novo assembly).

- This new use case and its requirements lead to the development of an entire subfield of alignment algorithms, often referred to as "short read aligners" or "short read mappers".

- Short read aligners are commonly used software tools in bioinformatics, designed to align a very large number of short reads (billions). Most short-read aligners work in the range of 50-300bp.

Limitations of short read aligners

- Most short read aligners will find only alignments that are reasonably similar to the target. This means that the algorithm gives up searching beyond a certain threshold.

- Most short read aligners typically cannot handle long reads or become inefficient when doing so.

Limitations of short read aligners

- The minimum length of the read is algorithm- and implementation-dependent. Many tools stop working properly when the read lengths drop under 30 bases.

- When studying small RNAs, for example, microRNAs, we have to look for tools that can align very short reads.

Mapping and alignment

- Alignment and mapping appear to mean the same thing, but there are subtle differences.

- The word “mapper” is often used to emphasize that the optimal alignment of a read is not guaranteed. The purpose of a mapper tool is locating a region in a genome, not producing an optimal alignment to that region.

Mapping and alignment

- Mapping:
  - A mapping is a region where a read sequence is placed.
  - A mapping is regarded to be correct if it overlaps the true region.

- Alignment:
  - An alignment is the detailed placement of each base in a read.
  - An alignment is regarded to be correct if each base is placed correctly.

Mapping and alignment

- Alignment oriented applications:
  - Studies examining SNPs and variations in a genome

- Mapping oriented applications:
  - Studies focusing on RNA-seq

Key features

- Alignment algorithm: global, local, or semi-global (global-local)?
- Is there a need to report non-linear arrangements?
- How will the aligner handle insertions and deletions?
- Can the aligner skip (or splice) over large regions?
- Can the aligner filter alignments to suit our needs?
- Will the aligner find chimeric alignments?

Applications: detecting germline variation

- The classic application is to align reads to a high-quality reference genome.

- The main goal in mapping genomic DNA back to the organism's genome reference sequence is to determine sequence variation.

- Variant detection in the germlines of individuals, families, or populations

- Frequent and infrequent variation (such as disease-causing mutations) within the gene pool.

Applications: detecting somatic variation in tumors

- DNA sequencing of tumor material has become feasible through the massive drop in sequencing cost.

- Standard solid-tumor histopathology practice typically results in DNA being extracted from formalin-fixed, paraffin-embedded (FFPE) tissue.

- DNA fragments derived from this archival format are limited in length and quality, making short to medium-length sequencing methods well suited for this purpose.

Applications: detecting somatic variation in tumors

- The surrounding normal tissue or a separate healthy specimen from the same individual is frequently used to differentiate somatic from germline variation.

- The variation frequency in a population of tumor cells down to a detection limit of a few percent can be measured if the depth of coverage is sufficient.

- Copy-number variation may be detected at very high resolution across both whole human genomes and whole-exome samples. The observed oversampling rate of a given genomic region is compared with the expected rate to infer copy-number changes.

Applications: RNA

- RNA reverse transcription and cDNA sequencing with alignment of the resulting reads are performed to measure expression levels of transcripts in a specific tissue.
- Alignments must be split along the exon-intron boundaries; very long introns and the small sizes of some exons make this a challenging problem to solve.
- Expression levels can also be derived from this form of analysis.
- In cancer, sequencing of transcripts can also identify and verify expressed somatic variations, such as fusion genes or single point mutations in tissue-specific isoforms.

Approximate string matching problem

- For each read (single or paired-end), the goal is to find its true location with respect to the reference.

- The true location is not known and is found by solving an approximate matching problem. Searching for occurrences of the read sequence within the reference sequence but allowing for some mismatches and gaps between the two.

Approximate string matching problem

- If a genome had no repeats/variants and a sequencing experiment introduced no errors, then the need for approximate matching would disappear. That is, assuming a sufficient read length relative to the genome size, one could find the true locations using exact matching.

- Eukaryotic genomes are rich in repeats, and each sequencing technology introduces errors (e.g. error rate of $\geq 0.1\%$ for Illumina)

Approximate string matching problem

- If a repeat sequence is perfectly identical, then it is not possible to determine a read's true location if it lies completely within the repeat.

- We choose an error model such that the location where the read aligns with the minimal number of errors is likely to be the true location.

- The error model must be chosen according to the sequencing technology. For example, Illumina reads have more mismatches than indels at a relatively low error rate.

Main algorithmic ideas

- Dynamic programming:
  - For each read, $O(mn)$ in time and space, where m is the length of the read and n is the length of a genome.
  - Much too slow to align billions of reads to genomes of size 10^9 bp or more.

- Two main algorithmic ideas to address the problem of large input sizes (both in number of reads and size of the reference) for approximating string matching:
  - Filtering
  - Indexing

Filtering

- Quickly exclude large regions of the reference where no approximate match can be found.

- This can, for example, be done by identifying short regions in the reference (known as k-mers) that share a short piece of the read without errors (known as seeds).

- Regions that do not share such a short region are filtered out.

Indexing

- Involves preprocessing the reference sequence, the set of reads, or both in a more intricate way.

- A benefit of such preprocessing into string indices is that it typically does not require scanning the whole reference, and it can therefore conduct queries much faster at the expense of larger memory consumption.

- The string indices that are currently used are: suffix array, enhanced suffix array, FM-index, a data structure based on the Burrows-Wheeler transform, some auxiliary tables.

Verification

- Most algorithms require a final verification to ensure that the read has an approximate occurrence in the used error model.

- It examines the area around each candidate to determine whether a full high-scoring alignment exists in that vicinity.

- This is usually done by fast versions of dynamic programming based algorithms that can be speed up by computing only a certain region of the dynamic programming matrix.

Repeats

- Repetitive DNA elements are prevalent in genomes. Repeats make up approximately 50% of the human genome.

- If a read originates from a repeat family, how can a short read aligner identify the exact instance from which the read came?

- This simply is not possible, especially when copies of the repeat are exactly the same. Because this is a possibility, it is useful for the read aligner to report a degree of confidence in its alignment.

Repeats

- Short read aligners attempt to measure this confidence by considering all of the alignments discovered in the process of aligning a read.

- If a read aligns equally well to several instances of a repeat, then the aligner might report low confidence.

- If the read aligns perfectly to one locus and very poorly to a few other loci, then the aligner might report high confidence.

- We should be careful to take this confidence into account when weighing evidence derived from the alignment.

Repeats

- Information about all alignments found during the search can be used to estimate the probability that the highest-scoring alignment is correct.

- By correct, we mean that the read was placed in its true point of origin with respect to the reference.

- Mapping quality:

$$Q = -10\log_{10}(1 - p_{cor})$$

where p_{cor} is the probability that the alignment is correct.

Repeats

- Note that mapping quality and alignment score are distinct measures.

- A high alignment score implies a large degree of similarity (i.e. few mismatches and gaps) between the read and the reference, but does not imply high mapping quality.

- For instance, consider a read that aligns perfectly to the genome in two distinct loci. The alignment score is high, but intuitively, there is approximately a 50% chance of choosing the incorrect alignment ($Q \leq 3$).

Sequence differences from the reference

- A key point when interpreting read alignments is that the sequence of the subject genome is not identical to the sequence of the reference genome.

- The subject and reference genomes may be extremely similar (e.g. the genomes of two unrelated humans are about 99.8% similar by sequence), but they are not identical.

- Differences that exist between genomes are often distributed unevenly along the genome's length.

Sequence differences from the reference

- For example, say that one has aligned a collection of sequencing reads and finds that, whereas most of the reference is covered nearly uniformly at a high average depth, some regions have little or no coverage.

- This could be evidence of a deletion in the subject genome, or could have arisen because the poorly covered regions contained so many sequence differences from the reference that the aligner failed to find the alignments that truly belonged in the region.

Substituting the reference genome

- A few approaches have been proposed to resolve this issue.
- One involve substituting the reference genome for another reference that more closely matches the subject genome.
- For example, if the subject is human and the ethnicity is known, a version of the reference genome from that ethnicity could be used. The tailored reference genome would differ from the reference genome such that common variants in the relevant population are used.

Substituting the reference genome

- A related idea is to use variant information inferred from read alignments to modify the reference genome to more closely resemble the subject genome.

- The alignment process can then be restarted using the new, tailored reference genome. This process can also be iterated until there are no more changes to be made to the reference.

SNP-aware alignment

- Another way to reduce the effect of differences between subject and reference genomes is to encode information about small-scale genetic variants in the reference sequence itself.

- For example, say a particular position in the genome is known to vary across individuals, and the alleles found there are either C or T. One could replace the character at the position with Y, the IUPAC code representing "either C or T".

- Read aligners that respect this convention are called SNP aware or SNP tolerant, and they avoid incurring a penalty when a C or T aligns a Y in the reference genome. This removes alignment score penalties associated with a nonreference allele, which in turn increases the fraction of reads for which the aligner finds an alignment.

- Such tools require that the user specifies an annotation file describing the SNP variants to be included in the reference genome.

Allelic alignment bias

- Say that one is analyzing sequencing reads from a human genome, which is diploid, and considering a position on the subject genome where there is a heterozygous SNP.

- Typically, one of the alleles of the SNP will match the allele in the reference genome (the reference allele), whereas the other will not.

- When aligning a read containing the reference allele, the read will match the reference genome at that position, but when aligning a read containing the nonreference allele, the read will mismatch at the position, reducing the alignment score.

Allelic alignment bias

- The aligner might fail to align reads containing many nonreference alleles, which in turn leads to an under-representation of reads from the haplotype with more nonreference alleles.
- For DNA sequencing data, this tends to drive the allelic balance away from the expected 1-to-1 ratio.
- Departure from the 1-to-1 ratio must therefore be expected by downstream tools, such as variant callers.

Allelic alignment bias

- Solutions to this problem:
  1) To mutate reference positions where heterozygous SNPs might occur in a third allele that is neither the reference allele nor the alternate allele. This alleviates the bias by ensuring that reference and alternate alleles are penalized equally.

  2) To use SNP-aware alignment. If the annotation provided to the SNP-aware aligner contains both alleles for a heterozygous position, then the aligner will not penalize alignments that overlap with the nonreference allele at that position.

  3) To build a custom, phased diploid reference genome for a human individual under study based on extensive genotype information for that individual and his or her ancestors.

The bwa aligner

- The bwa (Burrows-Wheelers Aligner, http://bio-bwa.sourceforge.net/) aligner is one of the most widely used short read alignment tool and is well suited for a broad number of applications.
- There are three algorithms:
  1. bwa-backtrack*
  2. bwa-sw**
  3. bwa-mem***
  4. bwa-mem2****
  5. minimap2*****

*Li H. and Durbin R. (2009) Fast and accurate short read alignment with Burrows-Wheeler Transform. Bioinformatics, 25:1754-60.

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

***Li H. (2013) Aligning sequence reads, clone sequences and assembly contigs with BWA-MEM. arXiv

****Md V., Misra S., Li H. and Aluru S. (2019) Efficient Architecture-Aware Acceleration of BWA-MEM for Multicore Systems. IEEE Parallel and Distributed Processing Symposium.

******Li H. (2018) Minimap2: pairwise alignment for nucleotide sequences. Bioinformatics, 34:3094-3100.

The bwa aligner

- BWA-backtrack (bwa aln/samse/sampe): designed for Illumina sequence reads up to 100bp

- BWA-SW (bwa bwasw): designed for longer sequences ranged from 70bp to 1Mbp, long-read support and split alignment

- BWA-MEM (bwa mem): designed for longer sequences ranged from 70bp to 1Mbp, long-read support and split alignment, the latest, generally recommended for high-quality queries (faster and more accurate), better performance than BWA-backtrack for 70-100bp Illumina reads

The bwa aligner

- BWA-MEM2: about twice as fast as BWA-MEM and outputs near identical alignments

- minimap2: has replaced BWA-MEM for PacBio and Nanopore read alignment. It retains all major BWA-MEM features, but is ~50 times as fast, more versatile, more accurate and produces better base-level alignment

The bwa aligner

- bwa first need to construct the FM-index for the reference genome (`bwa index`). This only needs to be done once.

- The Burrows-Wheeler Transform (BWT) index is often called the FM-index.

- The reads in FASTA/FASTQ files are then aligned against this index.

Index building

- Index building consists of simply preparing and reformatting the reference genome so that the program can search it efficiently.

- Each program will build a different type of index. Sometimes it may produce multiple files with odd looking names or extensions. For this reason, it is best to place the reference genome in a separate folder and store the indices there as well.

- Depending on the program, the original FASTA file of the reference genome may need to be kept at the original location.

- The time and computational resources required for building an index will depend on the genome size.

Installing bwa (Installed)

```
conda install -c bioconda bwa
```

Ebola reference genome

- We will align ebola sequencing data against the 1976 Mayinga reference genome.

- We will hold the reference genome and all indices:

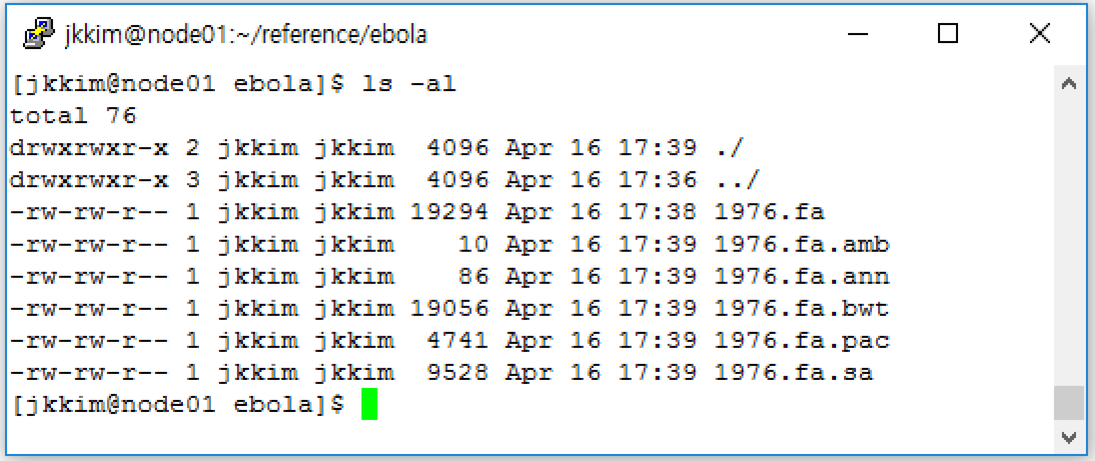
```
mkdir -p ~/reference/ebola
```

- Get the ebola genome in FASTA format:

```
efetch -db nuccore -format fasta -id AF086833 >
~/reference/ebola/1976.fa
```

Build an index with bwa

```
bwa index ~/reference/ebola/1976.fa
```



```
[jkkim@node01 ebola]$ ls -al
total 76
drwxrwxr-x 2 jkkim jkkim  4096 Apr 16 17:39 ./
drwxrwxr-x 3 jkkim jkkim  4096 Apr 16 17:36 ../
-rw-rw-r-- 1 jkkim jkkim 19294 Apr 16 17:38 1976.fa
-rw-rw-r-- 1 jkkim jkkim    10 Apr 16 17:39 1976.fa.amb
-rw-rw-r-- 1 jkkim jkkim    86 Apr 16 17:39 1976.fa.ann
-rw-rw-r-- 1 jkkim jkkim 19056 Apr 16 17:39 1976.fa.bwt
-rw-rw-r-- 1 jkkim jkkim  4741 Apr 16 17:39 1976.fa.pac
-rw-rw-r-- 1 jkkim jkkim  9528 Apr 16 17:39 1976.fa.sa
[jkkim@node01 ebola]$
```

Align a paired-end dataset

```
$fastq-dump -X 10000 --split-files SRR1972739
```

```
$bwa mem -t 10 -R
“@RG\tID:SRR1972739\tSM:ebola\tPL:Illumina”
~/reference/ebola/1976.fa SRR1972739_1.fastq
SRR1972739_2.fastq > SRR1972739.sam
```

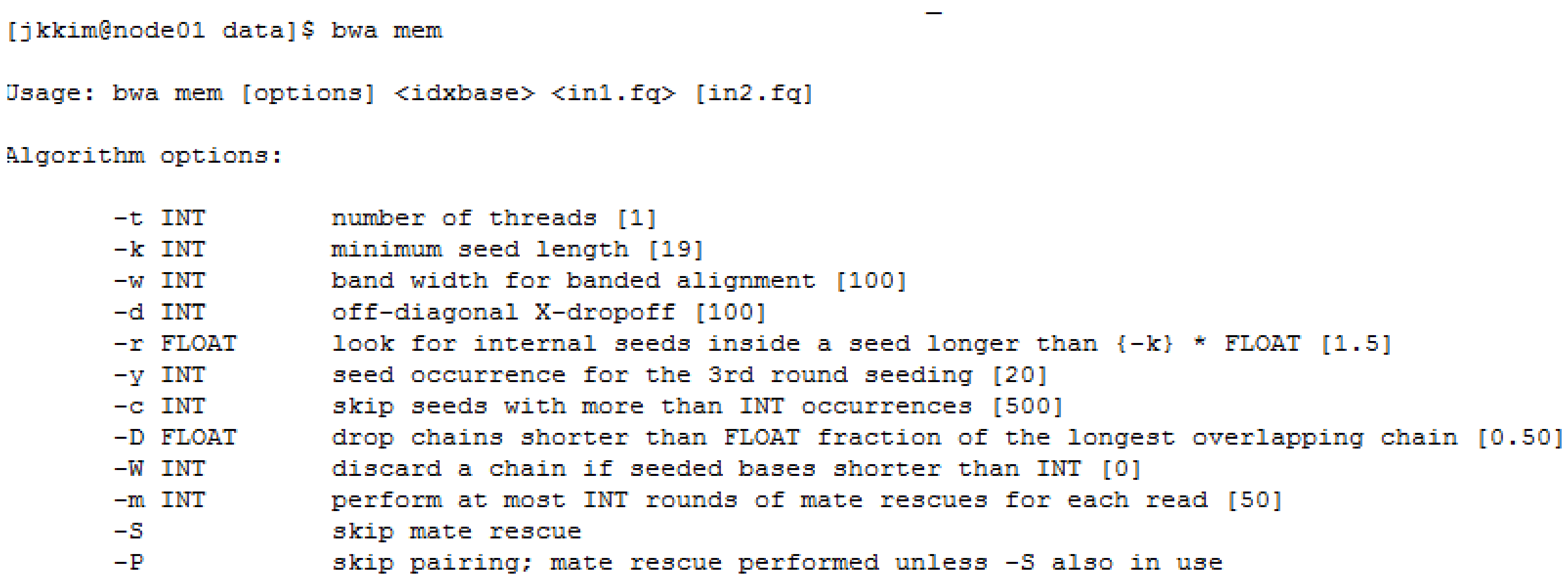
SAM

- The resulting file is in a so-called SAM format.

- It is one of the most recent bioinformatics data formats, one that by today has become the standard method to store and represent all high-throughput sequencing results

- A SAM file encompasses all known information about the sample and its alignment.

Help on bwa mem



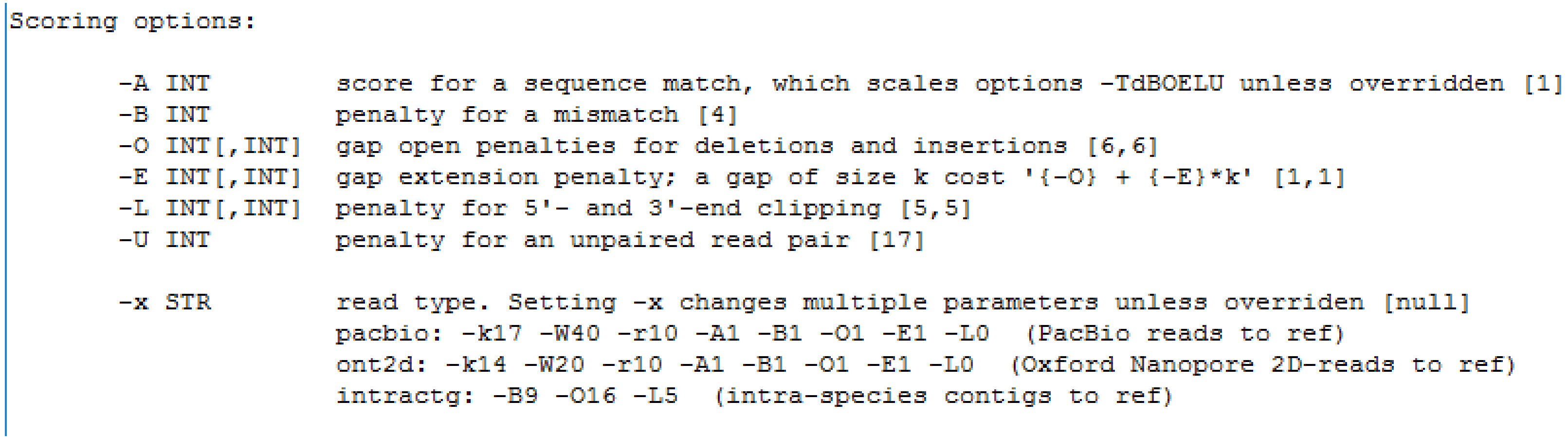
```
[jkkim@node01 data]$ bwa mem

Usage: bwa mem [options] <idxbase> <in1.fq> [in2.fq]

Algorithm options:

       -t INT        number of threads [1]
       -k INT        minimum seed length [19]
       -w INT        band width for banded alignment [100]
       -d INT        off-diagonal X-dropoff [100]
       -r FLOAT      look for internal seeds inside a seed longer than {-k} * FLOAT [1.5]
       -y INT        seed occurrence for the 3rd round seeding [20]
       -c INT        skip seeds with more than INT occurrences [500]
       -D FLOAT      drop chains shorter than FLOAT fraction of the longest overlapping chain [0.50]
       -W INT        discard a chain if seeded bases shorter than INT [0]
       -m INT        perform at most INT rounds of mate rescues for each read [50]
       -S            skip mate rescue
       -P            skip pairing; mate rescue performed unless -S also in use
```

Help on bwa mem



```
Scoring options:

       -A INT        score for a sequence match, which scales options -TdBOELU unless overridden [1]
       -B INT        penalty for a mismatch [4]
       -O INT[,INT]  gap open penalties for deletions and insertions [6,6]
       -E INT[,INT]  gap extension penalty; a gap of size k cost '{-O} + {-E}*k' [1,1]
       -L INT[,INT]  penalty for 5'- and 3'-end clipping [5,5]
       -U INT        penalty for an unpaired read pair [17]

       -x STR        read type. Setting -x changes multiple parameters unless overriden [null]
                     pacbio: -k17 -W40 -r10 -A1 -B1 -O1 -E1 -L0  (PacBio reads to ref)
                     ont2d: -k14 -W20 -r10 -A1 -B1 -O1 -E1 -L0  (Oxford Nanopore 2D-reads to ref)
                     intractg: -B9 -O16 -L5  (intra-species contigs to ref)
```