Week 7, Lecture 14

István Albert

Bioinformatics Consulting Center
Penn State, 2015
Instrumentation drives the methods

- Sanger instrumentation → few long reads → finding similarities between genomes → local alignment → homology

- High throughput sequencing → very large number of very short reads → semi-global alignment → matching known information
Processing sequencing reads

Sequenced DNA fragments (DNA library)

Unknown Genome
De novo assembly. Assembler tools

Known Genome
Resequencing Alignment tools

Known genome (reference)

gene discovery (annotation)

chip-Seq, RNA-Seq, SNP calling etc.
Alignment concepts

**NOTE:** mismatches or indels can be longer than 1 base!

It gets complicated very quickly

Alignment scoring depends on mismatch scoring (different across bases!), gap open, gap extension penalties
Challenges: repetitive and low complexity regions – where to place the query?
Challenge: genomic reorganization can easily lead to mis-mapping.

Reference genome (aka the illusion):

```
GCAA GTATGGGGGCAGGC
```

Real genome (with small rearrangement):

```
GCAA GTATGGGGGCAG
```

This is the query:

```
GCAA
```

Mismapping against the reference:

```
AAGTATGGGGGCAGG
```
Short Read Mappers (aligners)

• Use **heuristics** to quickly identify locations (hits) where the reads match

• Heuristics → not all hits will be found!

• Tradeoff: **resource usage vs speed vs accuracy vs usability**

• Each domain of application may have more appropriate tools
A few popular aligners

- BWA, bowtie, SOAP2 and many others

There is no **single best tool**, the issues to consider:

1. **documentation** → can we figure out how it works
2. **input features** → what type of input can it handle
3. **reporting features** → will it produce the type of output that we can use
4. **performance** → is it feasible to run on my resources
Aligner features

• Some cannot handle **indels** (insertion/deletions) → these tools are typically extremely fast will run on a laptop for even large genomes (**bowtie 1**) → application Chip-Seq

• May or may not use the quality score during the alignment (usually not, that would alter the alignment score)

• Differences in the they report alignments:
  – unique alignments only (note that this is an ill defined concept!)
  – best alignments above a cutoff
  – all possible alignments (can slow down the process greatly)
CPU time to align 1 million reads

from: http://www.massgenomics.org/short-read-aligners
Tool comparisons

The vast majority of tools comparisons lead to “red-herring” type fallacies

1. Tool A is faster than Tool B and we tacitly assume that the results are the same

2. Tool A is more ‘accurate’ than Tool B and we assume the resource utilization is be the same
Robust results should be tool independent!

• We need tools that are correct and finish running within our resources

• Results should be hold across methods

• There may be exceptions to this – but those need to be explicitly documented
BWA (Burrows-Wheeler Aligner) by Heng Li

Burrows-Wheeler Aligner (BWA) is an efficient program that aligns relatively short nucleotide sequences against a long reference sequence such as the human genome. It implements two algorithms, bwa-short and BWA-SW. The former works for query sequences shorter than 200bp and the latter for longer sequences up to around 100kbp. Both algorithms do gapped alignment. They are usually more accurate and faster on queries with low error rates. Please see the BWA manual page for more information.

http://bio-bwa.sourceforge.net/

Download, unpack, compile with make and link to bin

Read also the bwa-mem controversy – the bwa-mem paper rejection
Self discoverable tools

Usage:

```
$ 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]`
- `-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]
Aligning reads with BWA-MEM

1. **Index the genome** – this only needs to be done once for each genome → `bwa index` ...

2. **Create the alignment** → `bwa mem` ...
Create a query sequence from the Ebola genome.

Align your query files with both **BWA** and **BLAST** and look at the output of both processes.

Both are sequence aligners, what seems to be main difference that you observe?

Which one seems to run faster?