DTU

DTU Health Technology
Bioinformatics

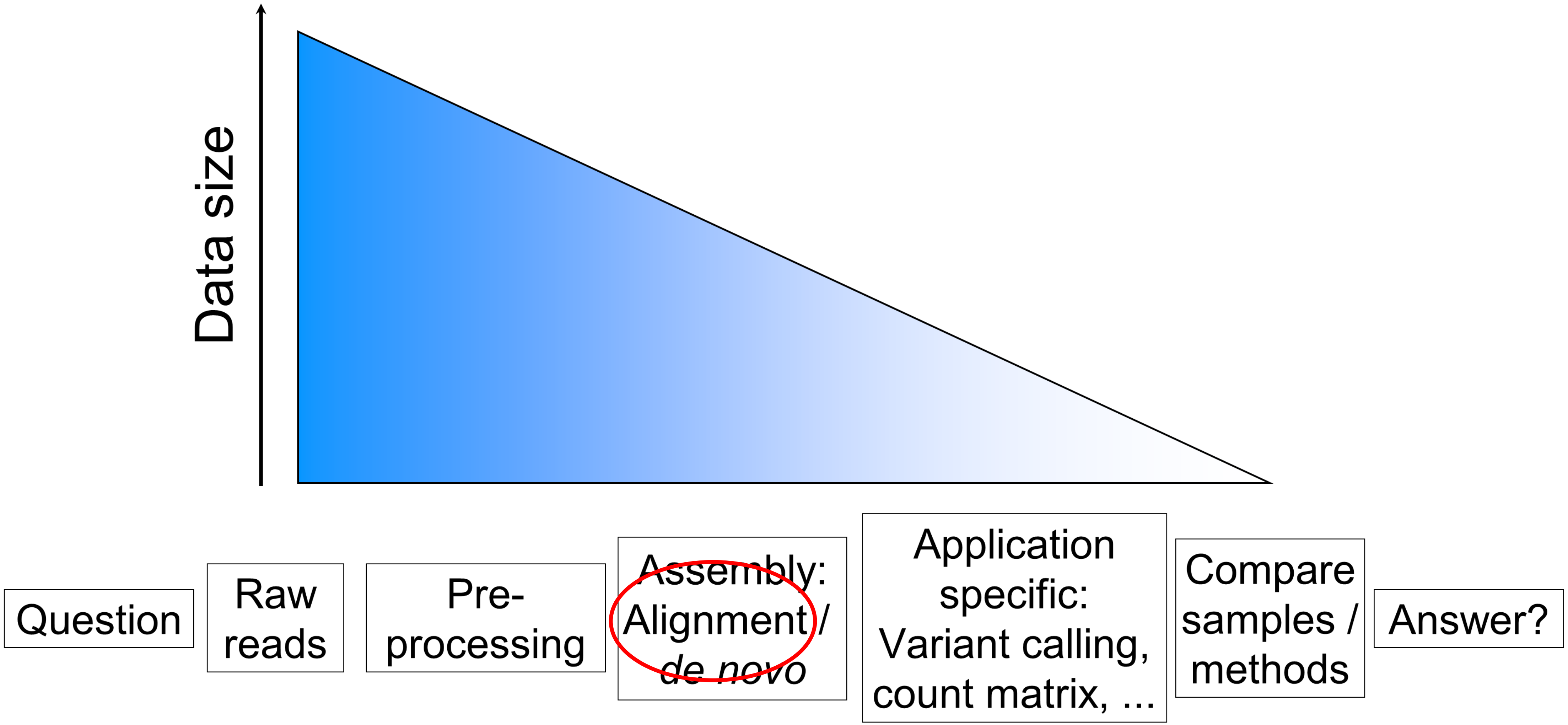
Alignment

Gisle Vestergaard
Associate Professor
Section of Bioinformatics
Technical University of Denmark
gisves@dtu.dk

Menu

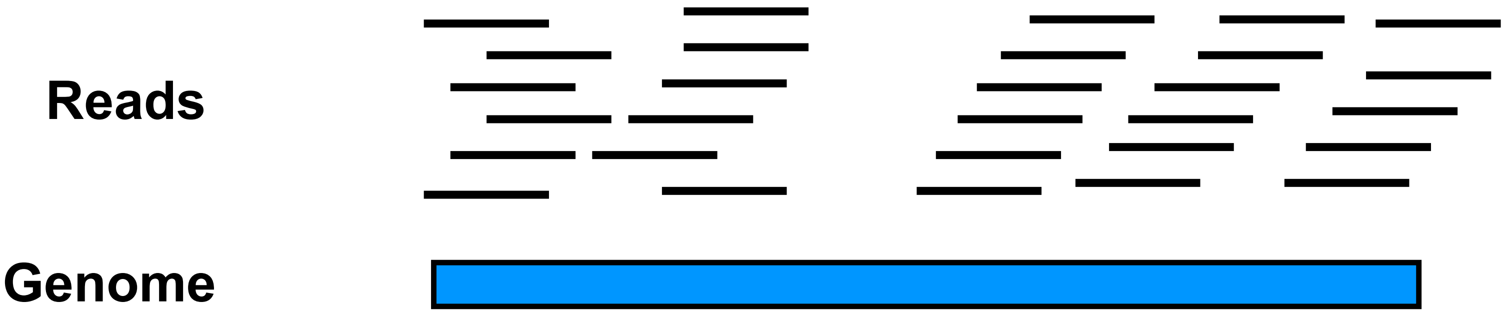
- Alignment approaches
- Burrows-Wheeler Transform
- Read-Depth
- SAM/BAM

Generalized NGS analysis



Alignment/Mapping

- Sometimes we have specific genomes of interest
- Sometimes we have specific genes of interest
- Assemble your reads by aligning them to a closely related reference genome



Sounds easy?

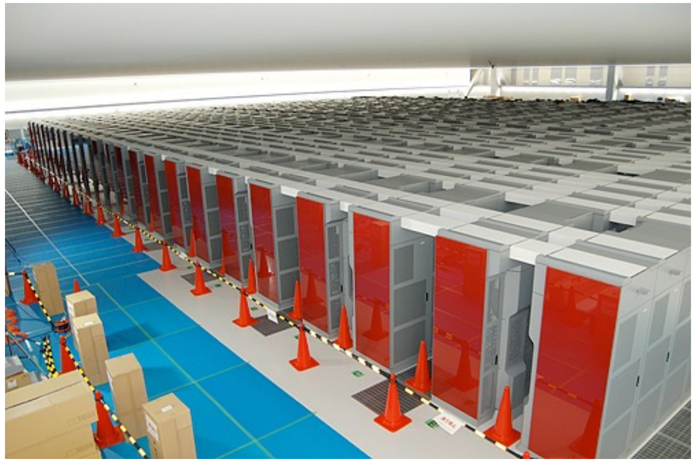
- Some pitfalls:
  - Divergence between sample and reference genome
  - Repeats in the genome
  - Recombination and re-arrangements
  - Poor reference genome quality
  - Read errors
  - Regions not in the ref. genome
  - Surprise sample

Simplest solution

- Exact string matches:

```
Reference: ACGTGCGGACGCTGAACGTGACG
               |||       | ||    |||
Read: GTG      GTG       G-TG    GTG
```

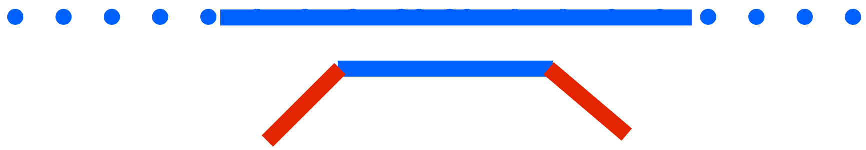
- We need to allow mismatches/indels (Smith-Waterman, Needleman-Wunsch)
- One of the worlds fastest computer (*K computer - RIKEN*)
- 20 mill reads 100 nt reads vs. human genome ~ 1 month
- We search each read vs. the <u>entire</u> reference



How about BLAST?

- Everybody uses BLAST
- Everybody will believe your BLAST hits (pun intended)

What we can learn: Reducing the search space



- However BLAST
  - finds local alignments - not always what we want for short reads
  - and other stuff (alignment scores, output format, speed)
- Not practical for short reads!

Smart solution

1. Use algorithm to quickly find *possible* matches

Drastically reduced search space

2. Allow us to perform slow/precise alignment for possible matches (Smith-Waterman)

3.2Gb

X possible matches

1 best match

Hash based algorithms

Lookups in hashes are *fast!*

1. Index the reference using *k*-mers.
2. Search reads vs. hash *k*-mers
3. Perform alignment of entire read around seed
4. Report best alignment

Key → Value

Key	Value
.	.
.	.
.	.
ACTGCGTGTGA	Chr1_pos1234; Chr2_pos567
ACTGCGTGTGC	Chr7_posX
ACTGCGTGTGT	Chr7_posZ; ...
.	.
.	.
.	.

Also known as *Seed and extend*

Spaced seeds

- Key/*k*-mer is called a seed
- BLAST uses *k*=11 and all must be matches

`11111111111`

L = 11, 11 matches

- Smarter: Spaced seeds (only care about “1” in seed)
  – Higher sensitivity
  – One can use several seeds

`111010010100110111`

L = 18, 11 matches

Multiple seeds & drawbacks

- One could require multiple short seeds
  - Instead of extending around each seed, extend around positions with several seed matches

- Drawbacks of hash-based approaches:
  - Lots(!) of RAM to keep index in memory (hg ~48Gb!)

Burrows-Wheeler Transform

- Hash based aligners require lots of memory and are only reasonable fast
- Can we make it better/faster?
- Burrows Wheeler Transformation (BWT)
- BWT was originally created for compression

The concepts

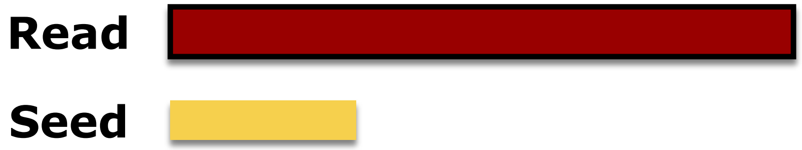
- Burrows-Wheeler Transform (BWT)
  - A reversible transformation of the genome
- Suffix Array is an array of integers giving the starting positions of suffixes of a string in lexicographical (alphabetical) order
- Full-text index in Minute space (FM) index
  - Allows us to recreate parts of the Suffix Array on the fly

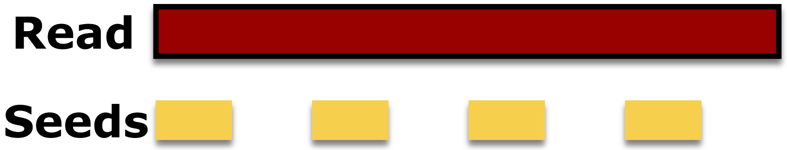
BWT for alignment

- Entire SA is 12Gb for human genome
  - Sorted means fast!
- FM-index
  - We only store certain parts of the array
  - We can calculate missing parts on the fly
  - Compressed means less memory!
- Human genome can be effectively indexed and searched using 3Gb RAM!

Two implementations in BWA

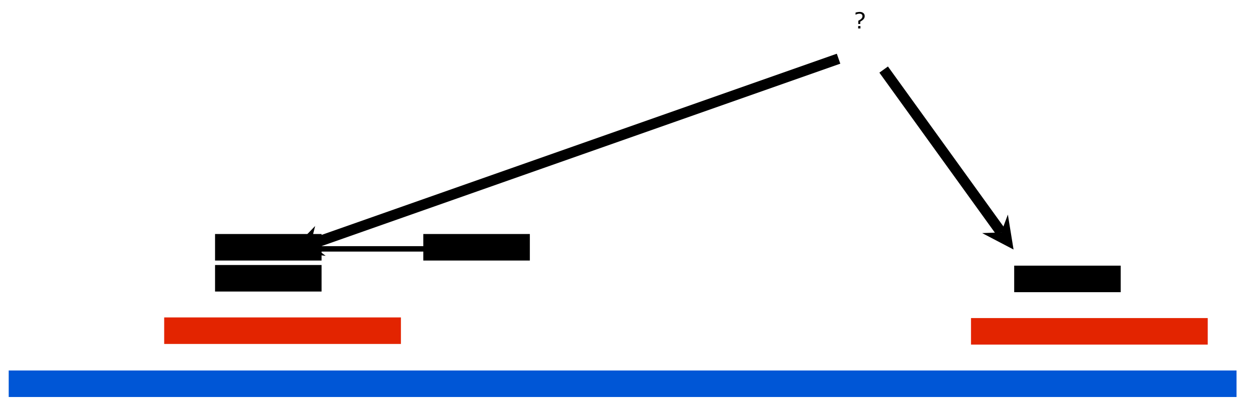
- Burrows Wheeler Aligner (BWA) can use:
  - *bwa aln*: First ~30nt of read as seed
    - Extend around positions with seed match
    - *For short reads*
  - *bwa mem*: Multiple short seeds across the read
    - Extend around positions with several seed matches
    - For longer reads





Single vs. Paired alignment

- Always get paired end reads (if possible)
- Can map across repeats
- Less mapping errors



Unmapped read can be “rescued” by a good aligning mate

Coverage

- Coverage/depth is how many times that your data covers the genome (on average)

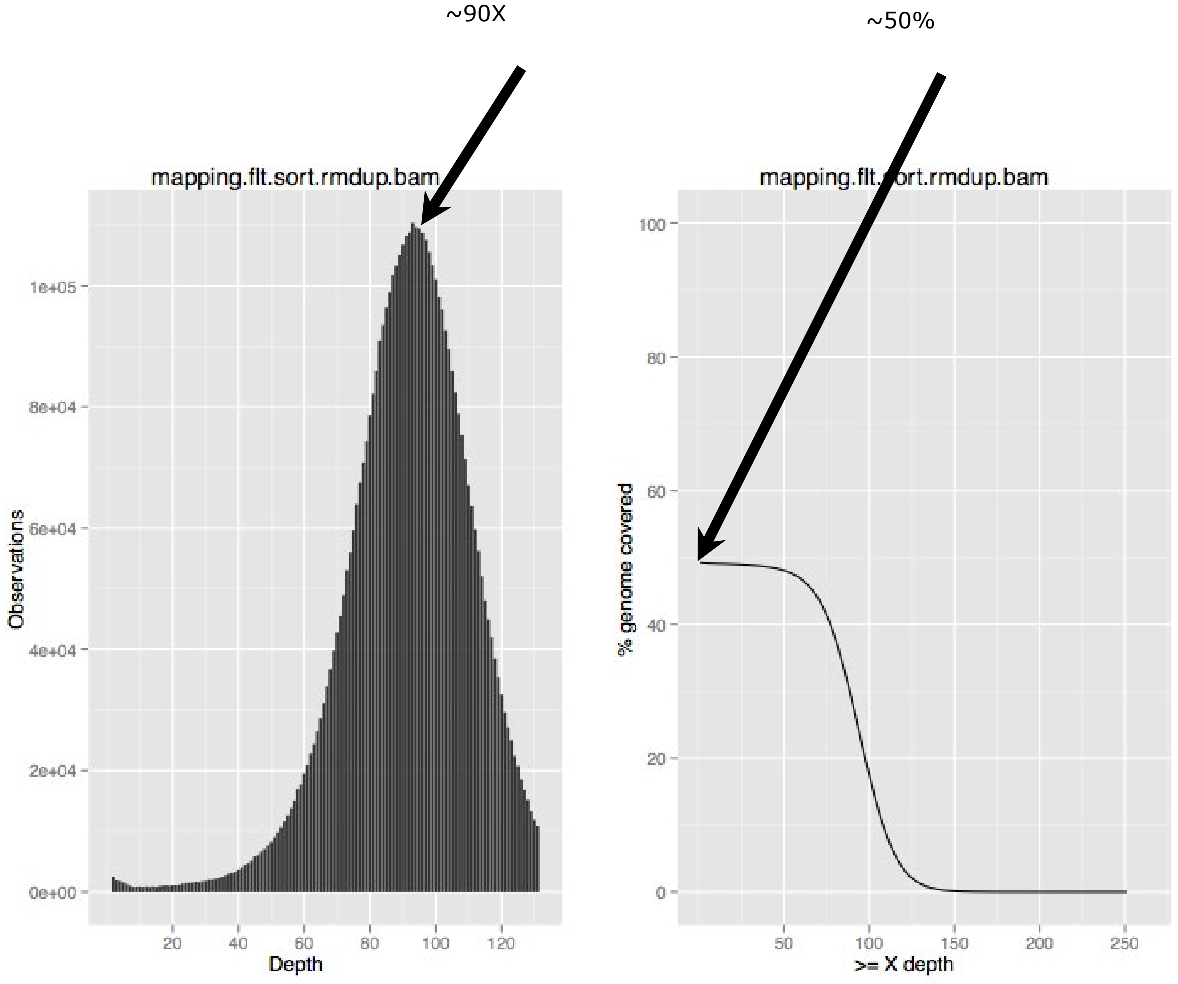
- Example:
  - N: Number of reads: 5 mill
  - L: Read length: 100
  - G: Genome size: 5 Mbases
  - C = 5*100/5 = 100X
  - On average there are 100 reads covering each position in the genome

$$C = N \times \frac{L}{G}$$

Actual depth

- We aligned reads to the genome - how much do we actually cover?

- Avg. depth ~ 90X
- Range from 0-250X
- Only 50% of the genome was covered with reads



SAM/BAM format

- Sequence Alignment / Map format
- BAM = Binary SAM and zipped - always convert to BAM
- Two sections
  - – Header: All lines start with “@”
  - – Alignments: All other lines

SAM - Example

```
@HD VN:1.5 SO:coordinate
@SQ SN:ref LN:45
r001   99 ref  7 30 8M2I4M1D3M = 37  39 TTAGATAAAGGATACTG *
r002    0 ref  9 30 3S6M1P1I4M * 0    0 AAAAGATAAGGATA    *
r003    0 ref  9 30 5S6M       * 0    0 GCCTAAGCTAA       * SA:Z:ref,29,-,6H5M,17,0;
r004    0 ref 16 30 6M14N5M    * 0    0 ATAGCTTCAGC       *
r003 2064 ref 29 17 6H5M       * 0    0 TAGGC             * SA:Z:ref,9,+,5S6M,30,1;
r001  147 ref 37 30 9M         = 7  -39 CAGCGGCAT         * NM:i:1
```

Header section

Alignment section

Optional fields in the format of TAG:TYPE:VALUE

QUAL: read quality; * meaning such information is not available

SEQ: read sequence

TLEN: the number of bases covered by the reads from the same fragment. Plus/minus means the current read is the leftmost/rightmost read. E.g. compare first and last lines.

PNEXT: Position of the primary alignment of the NEXT read in the template. Set as 0 when the information is unavailable. It corresponds to POS column.

RNEXT: reference sequence name of the primary alignment of the NEXT read. For paired-end sequencing, NEXT read is the paired read, corresponding to the RNAME column.

CIGAR: summary of alignment, e.g. insertion, deletion

MAPQ: mapping quality

POS: 1-based position

RNAME: reference sequence name, e.g. chromosome/transcript id

FLAG: indicates alignment information about the read, e.g. paired, aligned, etc.

QNAME: query template name, aka. read ID

Exercise time!

http://teaching.healthtech.dtu.dk/22126/index.php/Alignment_exercise