

# Aligning new-sequencing reads by BWA

Heng Li

Broad Institute

4 February 2010

# Outline

- 1 Short-read alignment
  - Overview of read alignment
  - Short-read aligners
  
- 2 BWA: Burrows-Wheeler Aligner
  - Overview of BWA
  - Running BWA

# Outline

- 1 Short-read alignment
  - Overview of read alignment
  - Short-read aligners
- 2 BWA: Burrows-Wheeler Aligner
  - Overview of BWA
  - Running BWA

# Overview of read alignment

- Alignment and *de novo* assembly are the first steps once read sequences are obtained.
- The task: to align sequencing reads against a known reference sequence for variation discovery (SNPs, indels and CNVs), CHIP-seq or RNA-seq.
- Difficulties: efficiency and ambiguity caused by repeats and sequencing errors.
- Aligners for long reads (>200bp): BLAT, SSAHA2 and BWA-SW.

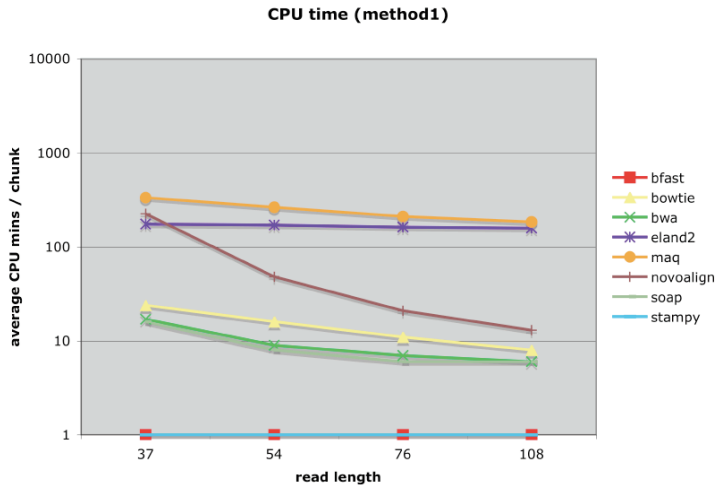
# There are many short-read aligners...

- Bfast
- BioScope
- Bowtie
- BWA
- CLC bio
- CloudBurst
- Eland/Eland2
- GenomeMapper
- GnuMap
- Karma
- MAQ
- MOM
- Mosaik
- MrFAST/MrsFAST
- NovoAlign
- PASS
- PerM
- RazerS
- RMAP
- SSAHA2
- Segemehl
- SeqMap
- SHRiMP
- Slider/SliderII
- SOAP/SOAP2
- Srprism
- Stampy
- vmatch
- ZOOM
- .....

# There are many short-read aligners...

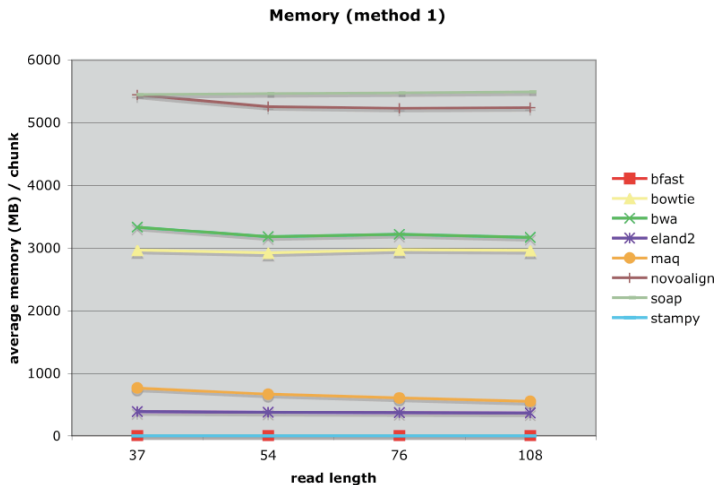
- Bfast
- BioScope
- Bowtie
- BWA
- CLC bio
- CloudBurst
- Eland/Eland2
- GenomeMapper
- GnuMap
- Karma
- MAQ
- MOM
- Mosaik
- MrFAST/MrsFAST
- NovoAlign
- PASS
- PerM
- RazerS
- RMAP
- SSAHA2
- Segemehl
- SeqMap
- SHRiMP
- Slider/SliderII
- SOAP/SOAP2
- Srprism
- Stampy
- vmatch
- ZOOM
- .....

# The speed varies...



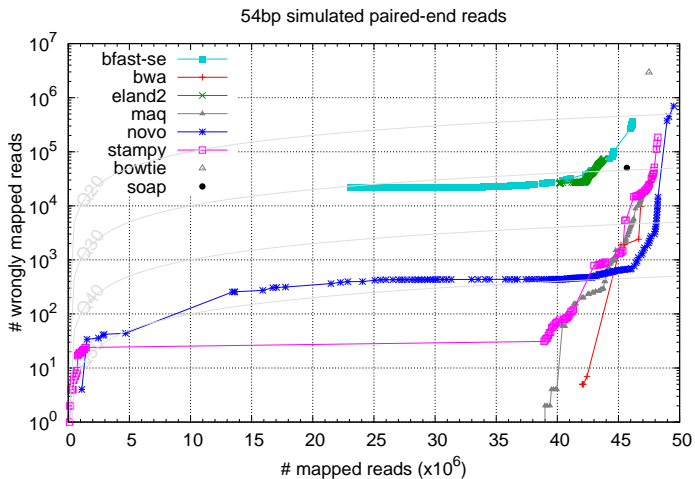
by Bala *et al.*

# The memory varies...



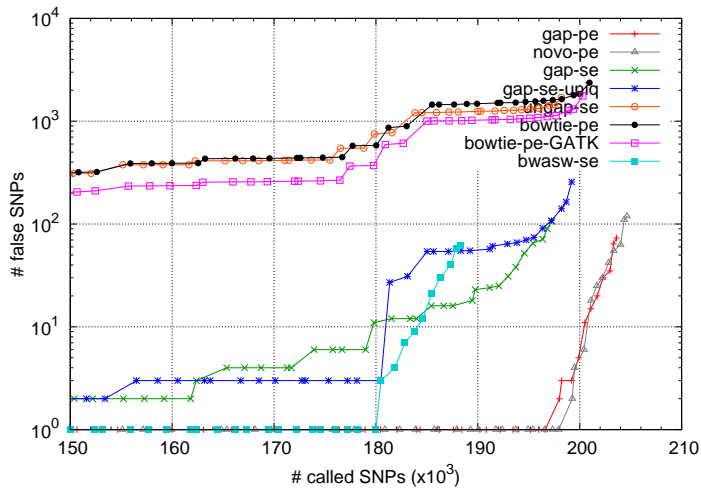
by Bala *et al.*

# The accuracy varies...



by Bala *et al.*

# Alignment strategy and SNP calling





# Choosing aligners

- There are many aligners and they vary a lot in performance.
- Aligners also vary in accuracy.
- Alignment accuracy is likely to affect the identification of structural variations (SVs), depending on algorithms though.
- In SNP calling, effective pair-end mapping and gapped alignment are essential to high SNP accuracy.

BWA is carefully designed to achieve a good balance between performance and accuracy.

# Choosing aligners

- There are many aligners and they vary a lot in performance.
- **Aligners also vary in accuracy.**
- Alignment accuracy is likely to affect the identification of structural variations (SVs), depending on algorithms though.
- In SNP calling, effective pair-end mapping and gapped alignment are essential to high SNP accuracy.

BWA is carefully designed to achieve a good balance between performance and accuracy.

# Choosing aligners

- There are many aligners and they vary a lot in performance.
- Aligners also vary in accuracy.
- Alignment accuracy is likely to affect the identification of structural variations (SVs), depending on algorithms though.
- In SNP calling, effective pair-end mapping and gapped alignment are essential to high SNP accuracy.

BWA is carefully designed to achieve a good balance between performance and accuracy.

# Choosing aligners

- There are many aligners and they vary a lot in performance.
- Aligners also vary in accuracy.
- Alignment accuracy is likely to affect the identification of structural variations (SVs), depending on algorithms though.
- In SNP calling, effective pair-end mapping and gapped alignment are essential to high SNP accuracy.

BWA is carefully designed to achieve a good balance between performance and accuracy.

# Choosing aligners

- There are many aligners and they vary a lot in performance.
- Aligners also vary in accuracy.
- Alignment accuracy is likely to affect the identification of structural variations (SVs), depending on algorithms though.
- In SNP calling, effective pair-end mapping and gapped alignment are essential to high SNP accuracy.

BWA is carefully designed to achieve a good balance between performance and accuracy.

# Outline

- 1 Short-read alignment
  - Overview of read alignment
  - Short-read aligners
- 2 BWA: Burrows-Wheeler Aligner
  - Overview of BWA
  - Running BWA

# Overview of the BWA algorithm

- Based on FM-index (Burrows-Wheeler Transform plus auxiliary data structures) which enables fast exact matching.
- Short-read algorithm: alter the read sequence such that it matches the reference exactly.
- Long-read algorithm (BWA-SW): sample reference subsequences and perform Smith-Waterman alignment between the subsequences and the read.
- Work for Illumina and SOLiD single-end (SE) and paired-end (PE) reads; new component BWA-SW for 454/Sanger SE reads.

# Key features

- Fast and moderate memory footprint (<4GB)
- SAM output by default
- Gapped alignment for both SE and PE reads
- Effective pairing to achieve high alignment accuracy; suboptimal hits considered in pairing.
- Non-unique read is placed randomly with a *mapping quality* 0; all hits can be outputted in a concise format.
- Guarantee to find *k*-difference in the seed (first 32bp by default).
- The default configuration works for most typical input.
  - Automatically adjust parameters based on read lengths and error rates.
  - Estimate the insert size distribution on the fly.

# Running BWA

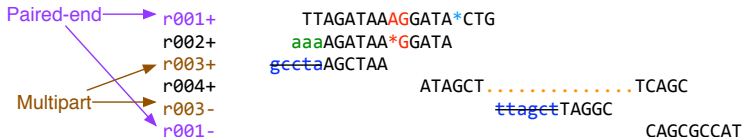
- Input: `ref.fa`, `read1.fq.gz`, `read2.fq.gz` and `long-read.fq.gz`
- Step 1: Index the genome ( $\sim 3$  CPU hours for the human genome):  
`bwa index -a bwtsv ref.fa`
- Step 2a: Generate alignments in the suffix array coordinate:  
`bwa aln ref.fa read1.fq.gz > read1.sai`  
`bwa aln ref.fa read2.fq.gz > read2.sai`  
Apply option `-q15` if the quality is poor at the 3'-end of reads.
- Step 3a: Generate alignments in the SAM format:  
`bwa sampe ref.fa read?.sai read?.fq.gz > aln.sam`
- Step 4a: Get multiple hits:  
`bwa samse -n 100 ref.fa read1.sai read1.fq.gz`
- Step 2b: Use BWA-SW for long reads:  
`bwa bwsw ref.fa long-read.fq.gz > aln-long.sam`

# The Sequence Alignment/Map format

```

coord 12345678901234 5678901234567890123456789012345
ref    AGCATGTTAGATAA**GATAGCTGTGCTAGTAGGCAGTCAGCGCCAT

```



Ins & padding  
 Soft clipping  
 Splicing  
 Hard clipping

```
@SQ SN:ref LN:45
```

```

r001 163 ref 7 30 8M2I4M1D3M = 37 39 TTAGATAAAGGATACTA *
r002 0 ref 9 30 3S6M1P1I4M * 0 0 AAAAGATAAAGGATA *
r003 0 ref 9 30 5H6M * 0 0 AGCTAA * NM:i:1
r004 0 ref 16 30 6M14N5M * 0 0 ATAGCTTCAGC *
r003 16 ref 29 30 6H5M * 0 0 TAGGC * NM:i:0
r001 83 ref 37 30 9M = 7 -39 CAGCGCCAT *

```

```

ref 7 T 1 . | ref 12 T 3 ... | ref 17 T 3 ...
ref 8 T 1 . | ref 13 A 3 ... | ref 18 A 3 ..-1G..
ref 9 A 3 ... | ref 14 A 2 .+2AG.+1G. | ref 19 G 2 *.
ref 10 G 3 ... | ref 15 G 2 .. | ref 20 C 2 ..
ref 11 A 3 ..C | ref 16 A 3 ... | ...

```

# Acknowledgement

- Richard Durbin and the Durbin research group
- All BWA users