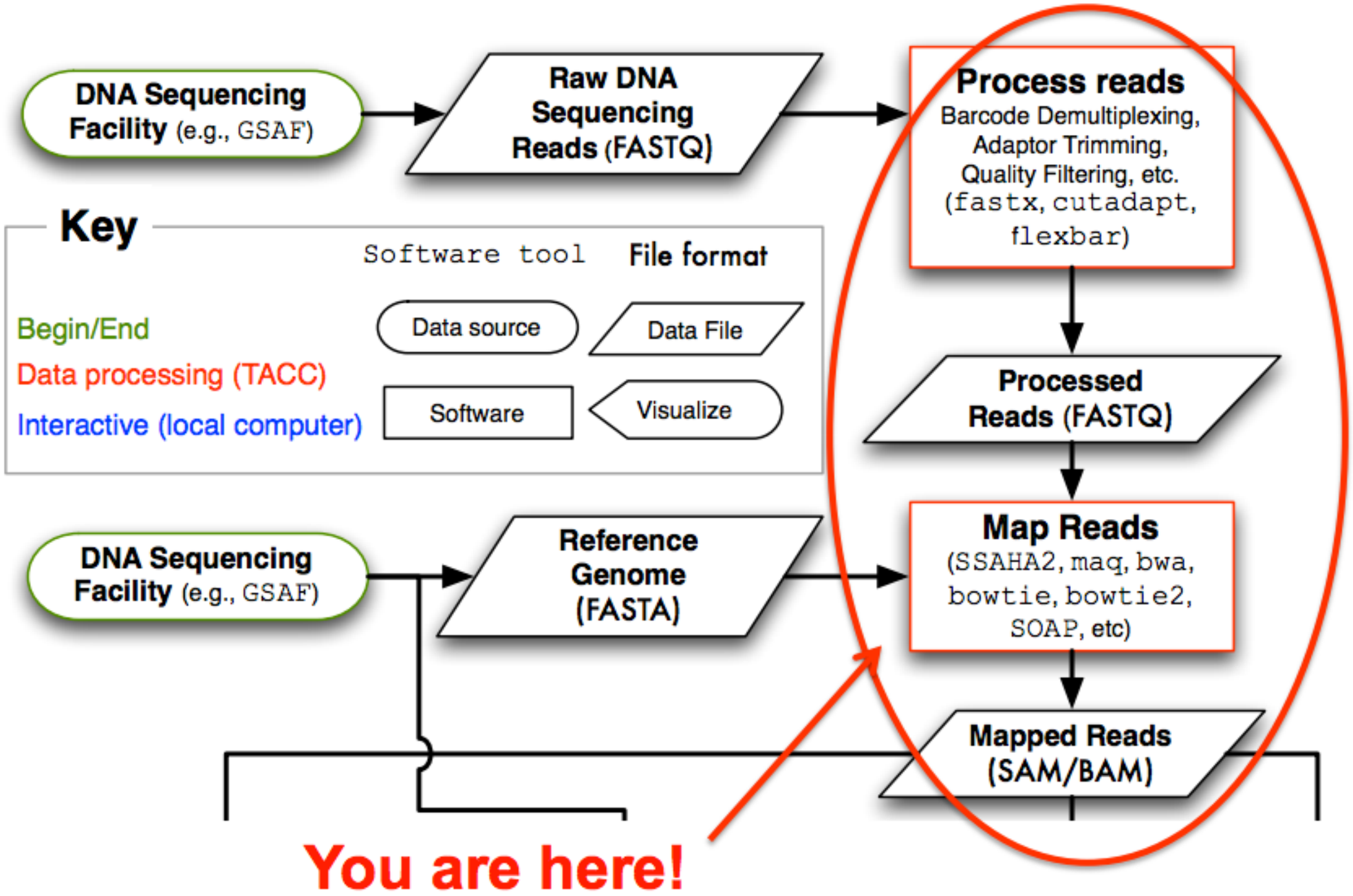
GENOMIKA

Alinhamento

Rodrigo Bertollo
rodrigo@genomika.com.br

Pipeline


DNA Sequencing Facility (e.g., GSAF)
Raw DNA Sequencing Reads (FASTQ)
Process reads
Barcode Demultiplexing, Adaptor Trimming, Quality Filtering, etc. (fastx, cutadapt, flexbar)
Key
Software tool
File format
Begin/End
Data processing (TACC)
Interactive (local computer)
Data source
Data File
Software
Visualize
Processed Reads (FASTQ)
DNA Sequencing Facility (e.g., GSAF)
Reference Genome (FASTA)
Map Reads
(SSAHA2, maq, bwa, bowtie, bowtie2, SOAP, etc)
Mapped Reads (SAM/BAM)
You are here!

Alignment/Mapping

- Assemble your reads by aligning them to a closely related reference genome
- High sequence similarity between individuals makes this possible

GENOMIKA

Some pitfalls

- 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

Basic Steps

1. Read file quality control

2. Build reference sequence index

3. Map DNA sequencing reads
   – Exact tool/approach depends on sequencing technology and DNA fragment library type

4. Convert result to SAM/BAM database

5. Application specific analysis...
   – These steps are common to any referencebased (opposed to de novo) data analysis.
   – We will look at variant calling.

Reference considerations

A reference genome is a consensus sequence built up from high quality sequencing samples from different populations. It is the control reference sequence to compare our samples.

Genome Reference Consortium (GRC) created to deliver assemblies:

- http://www.ncbi.nlm.nih.gov/projects/genome/assembly/grc/

Current human assembly is GRCh38, released in the summer of 2014. Major projects are considering to use it.

When GRCh37 was released, the UCSC genome browser team performed the following adaptation to the sequences, and called the end result “hg19”.

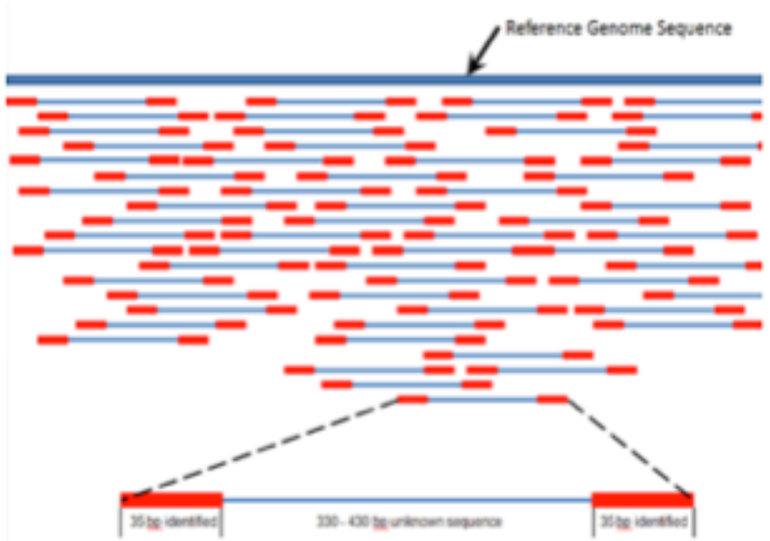
Reference genomes can be downloaded from:

– GRC: Human genome available at:
ftp://ftp.ncbi.nlm.nih.gov/genbank/genomes/Eukaryotes/vertebrates_mammals/Homo_sapiens/GRCh37/Primary_Assembly/assembled_chromosomes/FASTA/
– Ensembl: many available vertebrates genomes http://www.ensembl.org/info/data/ftp/index.html
– Ensembl Genomes: http://ensemblgenomes.org/

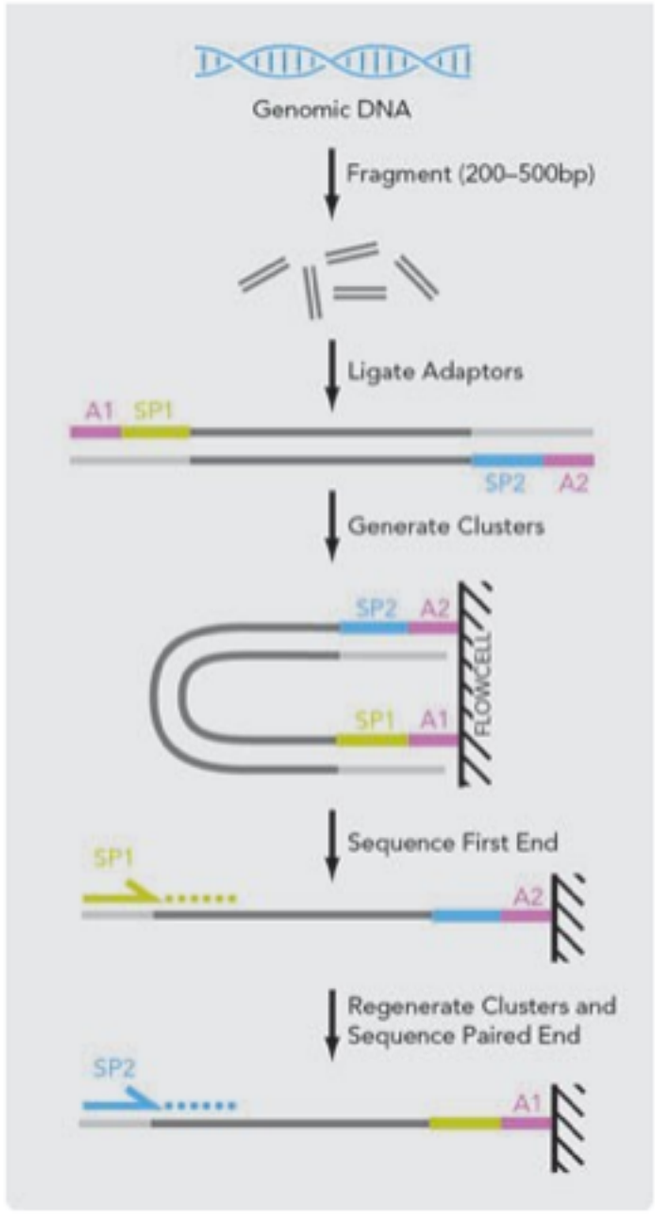
Mappers Aligners

Considerations:

- Which tool to use? What am I looking for? SNVs? INDELS? Long reads?
- Is it DNA or RNA?
- Single-end or paired-end? Paired-end when:
  - For very short reads, reduce the number of false positives alignments
  - Re-sequencing projects, Rna-seq?
  - Am I interested in Structural variation or gene fusions?
  - Reduce number of false positive variants
- Should I allow multiple hits?
- Should I remove low quality reads?



In general for *genomic variant analysis* we need high quality reads, paired-end datasets work better, and **no** multiple hits must be allowed



Taken from Illumina

Mappers Aligners

- Goals
  - **Sensitivity**, we are looking for genomic variants, reads with mismatches and INDELS must be properly aligned
  - **Specificity**, no wrong alignments should be provided
  - Being able to perform gapped alignments (RNA), exons must be correctly located
  - Good performance, efficiency matters
  - Easy to use
  - Open-source and maintained
  - Capable of align different data types: DNA, RNA-seq, BS-seq, …
- Unfortunately... most tools or algorithms only work well in a specific scenario

Algorithms

Smith-Waterman (SW)

SW finds the optimal local alignment between:

Sequence 1 = ACACACTA

Sequence 2 = AGCACACA

Given gap-scoring penalties:

w(match) = +2

w(a,-) = w(-,b) = w(mismatch) = -1

$$H = \begin{pmatrix} & - & A & C & A & C & A & C & T & A \\ - & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ A & 0 & 2 & 1 & 2 & 1 & 2 & 1 & 0 & 2 \\ G & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 1 \\ C & 0 & 0 & 3 & 2 & 3 & 2 & 3 & 2 & 1 \\ A & 0 & 2 & 2 & 5 & 4 & 5 & 4 & 3 & 4 \\ C & 0 & 1 & 4 & 4 & 7 & 6 & 7 & 6 & 5 \\ A & 0 & 2 & 3 & 6 & 6 & 9 & 8 & 7 & 8 \\ C & 0 & 1 & 4 & 5 & 8 & 8 & 11 & 10 & 9 \\ A & 0 & 2 & 3 & 6 & 7 & 10 & 10 & 10 & 12 \end{pmatrix}$$

Alignment result:

Sequence 1 = A - C A C A C T A

Sequence 2 = A G C A C A C - A

- Very popular algorithm developed in 198
- Provides a very **high sensitivity,** allowin alignments with any number of mismatch insertions and deletions
- Gives an *optimal alignment* between two sequences given a penalties, **it is not a mapper but an sequence aligner**
- No suitable for whole genome alignment: a 100bp read and the human genome 3C the matrix dimension: 100x3.10^9, using 4 Bytes for integers: **1.2TB of RAM !!**
- Although *dynamic programming* techniqu are applied to make SW more efficient, th CPU requirements are still too high, **SW too slow for NGS**

Algorithms

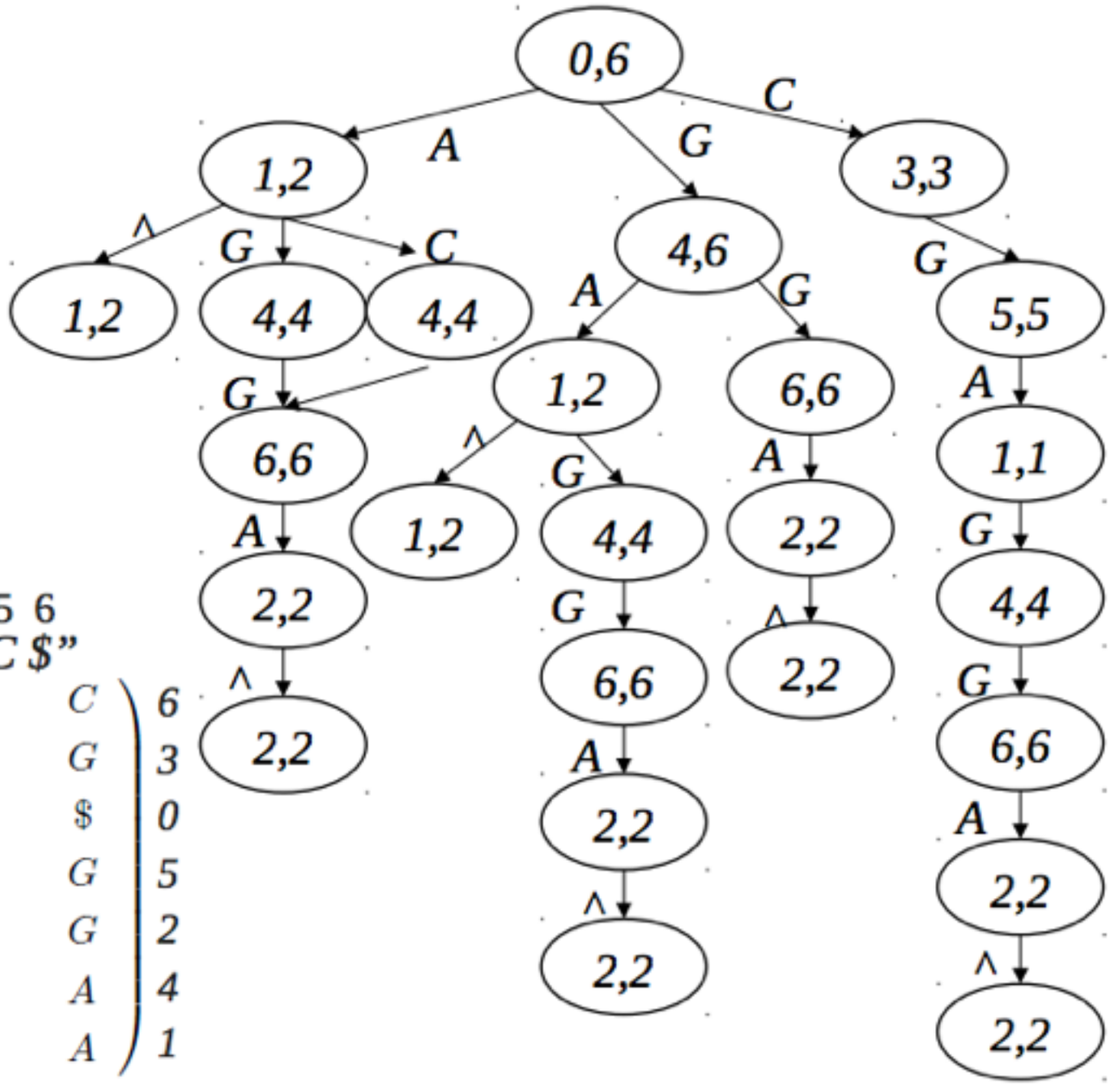
BLAST

- BLAST is one of the most widely used programs in Bioinformatics developed in 1990 at NIH. Allows comparing and searching amino-acid and DNA sequences in a database of sequences
- BLAST uses a heuristic algorithm to speed-up searches, it is **much faster** than calculating an optimal alignment with Smith-Waterman, **but it cannot guarantee the optimal alignment** of the query sequence in the database. It searches the most relevant *seeds* from query sequence in exact way and then SW is applied
- It presents a **high sensitivity,** allowing alignments with any number of mismatches, insertions and deletions, it can be used to align sequence between species
- However, it is **still too slow** for NGS mapping, blast can align few thousands sequences per hour

Algorithms

Burrows-Wheeler Transform (BWT)

- BWT is an algorithm used in data compression techniques such as *bzip2*
- It **efficiently** align short sequencing reads against a large reference sequence such as the human genome, a **prefix tree index** is created using reference genome
- In the transformation all permutations are sorted and all suffixes are grouped
- It is **much faster** than BLAST, it can align hundred of thousands sequences per second!
- However, it presents a **lower sensitivity**, it can allow a few mismatches, and in some implementation one INDEL



$$R = \text{“}A\,G\,G\,A\,G\,C\,\$\text{”}$$

0	0	1	2	3	4	5	6	
0	$	A	G	G	A	G	C	6
1	A	G	C	$	A	G	G	3
2	A	G	G	A	G	C	$	0
3	C	$	A	G	G	A	G	5
4	G	A	G	C	$	A	G	2
5	G	C	$	A	G	G	A	4
6	G	G	A	G	C	$	A	1

Algorithms

Suffix Arrays (SA)

- SA uses more memory than other related compressed data structures like BWT
- It **very efficiently** aligns sequencing reads against a large reference sequence such as the human genome, a **suffix array index** is created using reference genome
- Since it is not compressed the **performance** achieved is higher than with BWT, it can align million sequences per second in modern CPUs!
- Depending on the alignment strategy, a **high sensitivity** can be achieved, it can allow a relative high number of mismatches, and in some implementations several INDELs

1	attcatg$
2	ttcatg$
3	tcatg$
4	catg$
5	atg$
6	tg$
7	g$
8	$

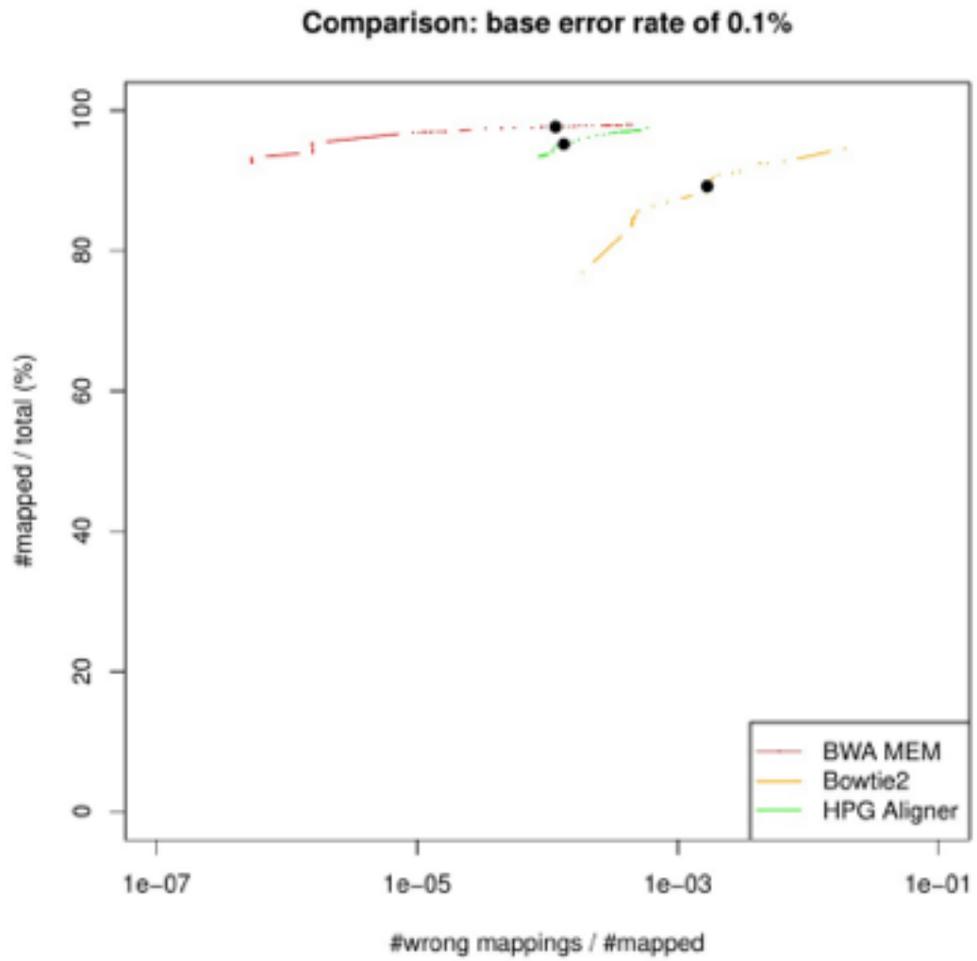
sort the suffixes alphabetically

the indices just "come along for the ride"

8	$
5	atg$
1	attcatg$
4	catg$
7	g$
3	tcatg$
6	tg$
2	ttcatg$

Which aligner to use ?

- Many aligners available, more than 70!!
  - http://wwwdev.ebi.ac.uk/fg/hts_mappers/
- Can be difficult to select one, some criteria
  - Type of analysis: dna, rna, meth
  - Number of cites
  - ...
- **Selecting an aligner**: simulate datasets to choose the best:
  - Which one is more sensitive to INDELS?
  - Which produce less false positives alignments
  - Which RNA aligner works better with low coverage?
  - ...
- All of them work similarly
  - **Reference genome index**: this index can be a Burrows-Wheeler Transform (BWT), Suffix array (SA), ...
  - The reads are **aligned to that index or are split in seeds an then aligned**, seeds aligned are clustered together
  - In general poor performance when high number of mismatches or INDELS are present



BWA

- BWA stands from Burrows-Wheeler Aligner, developed by R. Durbin at Sanger Institute
  - http://bio-bwa.sourceforge.net/
- It was one of the first NGS mappers and is widely used, provides very good results in common scenarios
- It implements BWT and Suffix Arrays (SA) with support for few errors:
  - *BWA-SW and BWA-MEM both tolerate more errors given longer alignment. Simulation suggests that they may work well given 2% error for an 100bp alignment, 3% error for a 200bp, 5% for 500bp and 10% for 1000bp or longer alignment*
- Implementation is in C and it is multi-thread, but lacks some features such as support for RNA-seq or big INDELS
- Not designed to take advantage of new technologies and clusters, not specially fast

Bowtie

Bowtie allowed a few mismatches (<3) and no gaps, claimed to be the fastest, but it missed many reads

Bowtie2 improved sensitivity when compared to Bowtie:

- http://bowtie-bio.sourceforge.net/bowtie2/index.shtml

Widely used, however it is a little bit less sensitivity than BWA, fail to correctly map many mismatches and INDELS

Implementation is in C and it is multi-thread, but lacks some biological features such as support for RNA or big INDELS

Not designed to take advantage of new technologies and clusters

From mapped read to Alignment

- **Mapping** determines a "seed" position where the read shares a subsequence with the reference. But, is this the best match?
- **Alignment** starts with the seed and determines how the read is best aligned on a base-by-base basis around the seed.

Seed→Alignment score→Mapping quality

Alignment Score

- Dynamic programming algorithm (Smith-Waterman | Needleman-Wunsch)
- **Alignment score = Σ**
  - match reward
  - base mismatch penalty
  - gap open penalty
  - gap extension penalty
  - rewards and penalties may be adjusted for quality scores of bases involved

Reference sequence

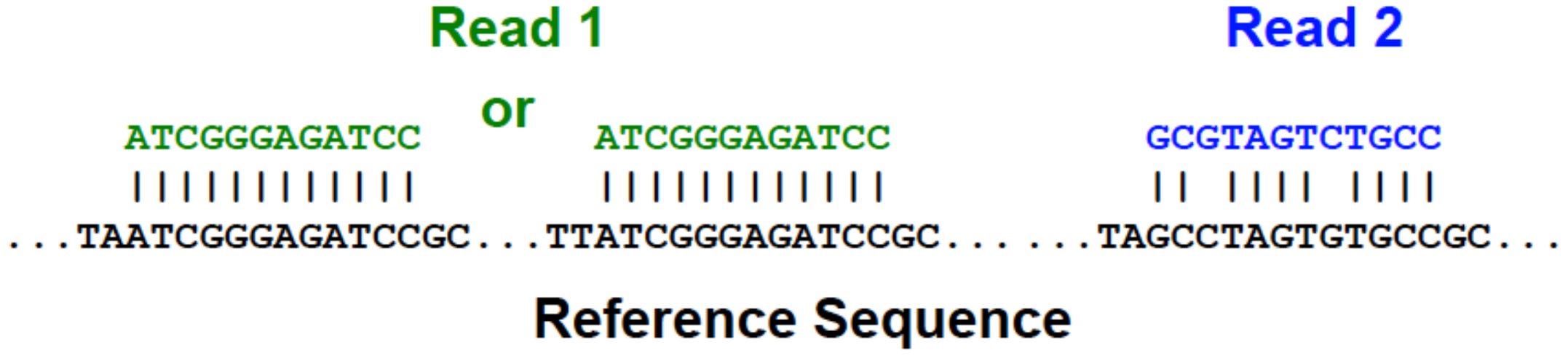
```
ATTTGCGATCGGATGAAGACGAA
|||||||||||||||
ATTTGCGATCGGATGTTGACTTT

ATTTGCGATCGGATGAAGACG..AA
|||||||||||||||XX|||XXX||
ATTTGCGATCGGATGTTGACTTTAA
```

- **Important: Local** versus **global** alignment

Mapping Quality

Mapping quality– what is the probability that the read is correctly mapped to this location in the reference genome?



High **alignment** score ≠ high **mapping** quality.

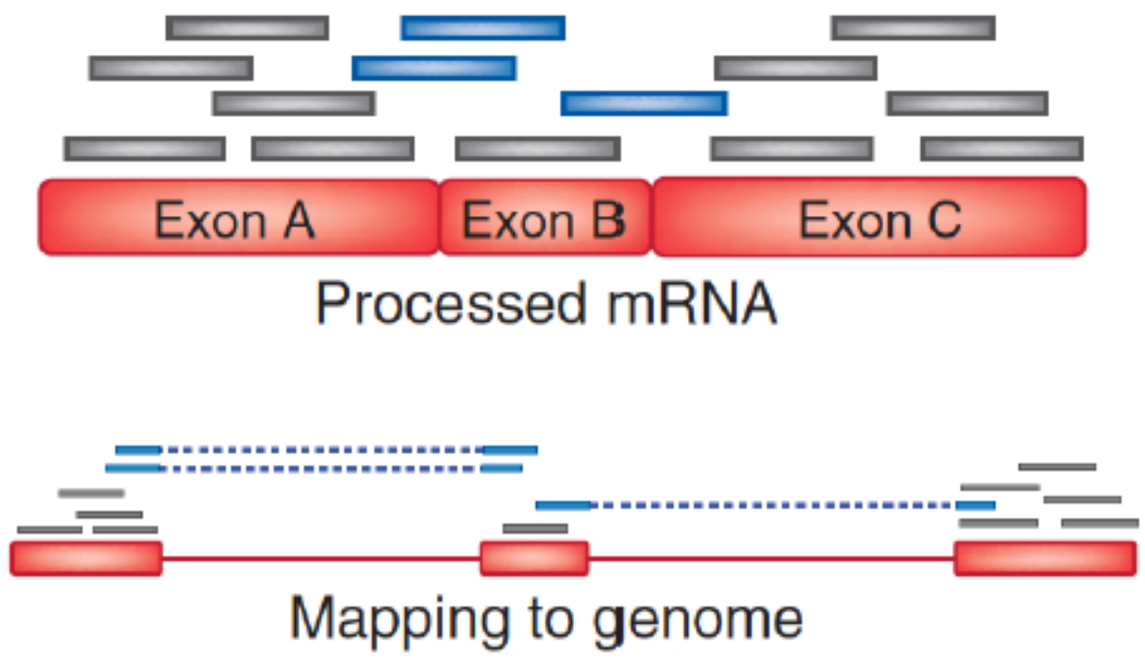
Phred score: $P(\text{mismapped}) = 10^{-MQ/10}$

Paired-End Mapping

- There is an expected insert size distribution based on the DNA fragment library.
- Mapping one read anchors the paired read to a specific location, even if the second read alone maps multiple places equally.
- Only one read in a pair might be mappable. (**singleton/orphan**)
- Both reads can map with an unexpected insert size or orientation (**discordant pair**)

Split-Read Alignment

- Useful for predicting splice variants or structural variants.
- Not many mappers do this directly, usually happens in a post-processing step.



Trapnell, C. & Salzberg, S. L. How to map billions of short reads onto genomes. *Nature Biotech.* **27**, 455–457 (2009).

SAM File Format

Community flat file/database format that describes how reads align to a reference (and can also include unmapped reads).

Can tag reads as being from different instrument runs / technologies / samples.

Going forward you need the reference file and the SAM, no longer need the FASTQ.

Tab delimited with fixed columns followed by arbitrary user-extendable key:data values.

SAM File Format

Two example SAM lines:

```
SRR030257.264529      99   NC_012967   1521      29   34M2S   =    1564      79
     CTGGCCATTATCTCGGTGGTAGGACATGGCATGCCC
     AAAAAA;AA;AAAAAA??A%.;?&'3735',()0*,
     XT:A:M  NM:i:3  SM:i:29  AM:i:29  XM:i:3  XO:i:0  XG:i:0  MD:Z:23T0G4T4

SRR030257.2669090     147  NC_012967   1521      60   36M     =    1458      -99
     CTGGCCATTATCTCGGTGGTAGGTGATGGTATGCGC
     <<9:<<AAAAAAAAAAAAAAAAAAAAAAAAAAAA
     XT:A:U  NM:i:0  SM:i:37  AM:i:37  X0:i:1  X1:i:0  XM:i:0  XO:i:0  XG:i:0  MD:Z:36
```

SAM File Format

SAM fixed fields:

http://samtools.sourceforge.net/

Col	Field	Type	Regexp/Range	Brief description
1	QNAME	String	`[!-?A-~]{1,255}`	Query template NAME
2	FLAG	Int	`[0,2^16-1]`	bitwise FLAG
3	RNAME	String	`\*|[!-()+-<>-~][!-~]*`	Reference sequence NAME
4	POS	Int	`[0,2^29-1]`	1-based leftmost mapping POSition
5	MAPQ	Int	`[0,2^8-1]`	MAPping Quality
6	CIGAR	String	`\*|([0-9]+[MIDNSHPX=])+`	CIGAR string
7	RNEXT	String	`\*|=|[!-()+-<>-~][!-~]*`	Ref. name of the mate/next segment
8	PNEXT	Int	`[0,2^29-1]`	Position of the mate/next segment
9	TLEN	Int	`[-2^29+1,2^29-1]`	observed Template LENgth
10	SEQ	String	`\*|[A-Za-z=.]+`	segment SEQuence
11	QUAL	String	`[!-~]+`	ASCII of Phred-scaled base QUALity+33

```
SRR030257.264529    99   NC_012967   1521   29   34M2S   =   1564
    79   CTGGCCATTATCTCGGTGGTAGGACATGGCATGCCC
    AAAAAA;AA;AAAAAA??A%.;?&'3735',()0*,
    XT:A:M  NM:i:3  SM:i:29  AM:i:29  XM:i:3  XO:i:0  XG:i:0  MD:Z:23T0G4T4
```

CIGAR

```
Ref   CTGGCCATTATCTC--GGTGGTAGGACATGGCATGCCC
Read        aaATGTCGCGGTG.TAGGAggatcc
```

2S5M2I4M1D4M6S

	Op	BAM	Description
	M	0	alignment match (can be a sequence match or mismatch)
	I	1	insertion to the reference
	D	2	deletion from the reference
*	N	3	skipped region from the reference
	S	4	soft clipping (clipped sequences present in SEQ)
*	H	5	hard clipping (clipped sequences NOT present in SEQ)
*	P	6	padding (silent deletion from padded reference)
*	=	7	sequence match
*	X	8	sequence mismatch

*Rarer / newer

CIGAR = "Concise Idiosyncratic Gapped Alignment Report"

FLAGs

Bit	Description
0x1	template having multiple segments in sequencing
0x2	each segment properly aligned according to the aligner
0x4	segment unmapped
0x8	next segment in the template unmapped
0x10	SEQ being reverse complemented
0x20	SEQ of the next segment in the template being reversed
0x40	the first segment in the template
0x80	the last segment in the template
0x100	secondary alignment
0x200	not passing quality controls
0x400	PCR or optical duplicate

Flags

BAM Format

- "Human readable" text (SAM) and GZIP compressed binary (BAM) versions.
- BAM files can be **sorted** and **indexed**, so that all reads mapped to a given window of the reference genome can be retrieved rapidly (for display or processing).

GENOMIKA

Alinhamento

Marcel Caraciolo, CTO
marcel@genomika.com.br