Genomics and Transcriptomics

Class 07 - Sequence Mapping

INSTRUCTOR:
Aureliano Bombarely
Department of Bioscience
Universita degli Studi di Milano
aureliano.bombarely@unimi.it

Outline of Topics

1. Basics of read mapping.

2. Short read mapping.

3. Long read mapping.

4. Mapping of transcriptomic reads.

5. Visualization of mapped reads.

6. Uses and analysis.

Outline of Topics

1. Basics of read mapping.

2. Short read mapping.

3. Long read mapping.

4. Mapping of transcriptomic reads.

5. Visualization of mapped reads.

6. Uses and analysis.

1. Basics of read mapping.

<u>Read Mapping</u>:
It is the process of search the **location** of a read **comparing** the its sequence and the **sequence of a reference**.

```
1.........10.........20.........30.........40.........50
ATGGCGTGGCAGCGACCAGTGACCAGTGACGTGTGCAGACGTGATATGCAG     ref
          |||||||||||  ||
          GCAGCGACCAGCGA    read
```

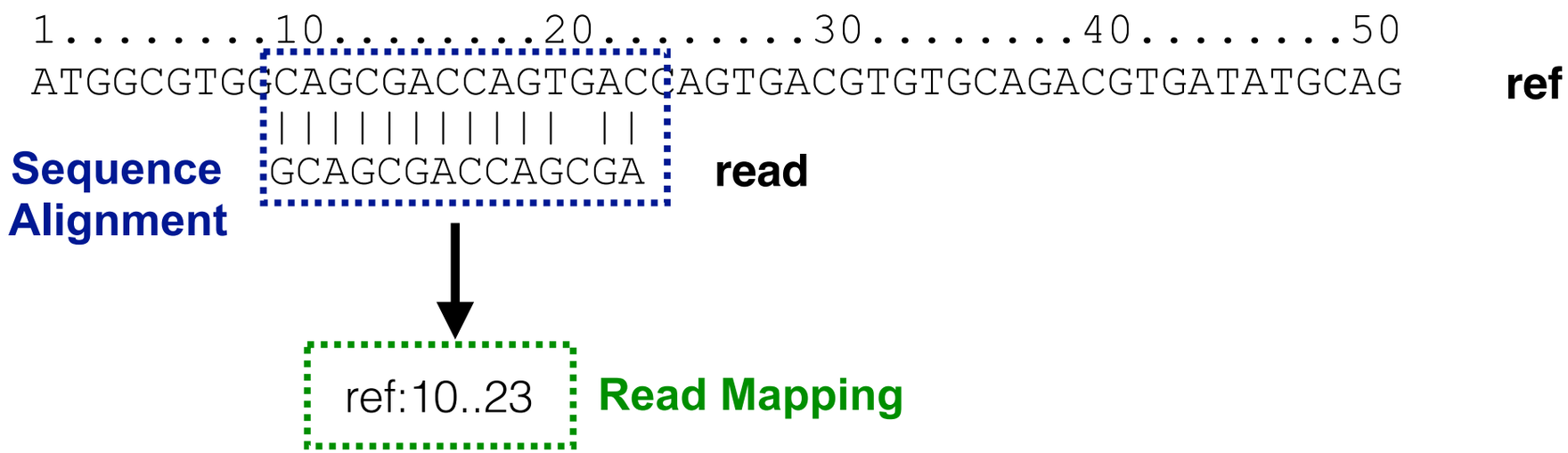
ref:10..23

Sequence Alignment:
In bioinformatics, a sequence alignment is a way of arranging the sequences of DNA, RNA, or protein to identify regions of similarity that may be a consequence of functional, structural, or evolutionary relationships between the sequences.
Aligned sequences of nucleotide or amino acid residues are typically represented as rows within a matrix.

http://en.wikipedia.org/wiki/Sequence_alignment

1. Basics of read mapping.

<u>Read Mapping</u>:
It is the process of search the **location** of a read **comparing** the its sequence and the **sequence of a reference**.



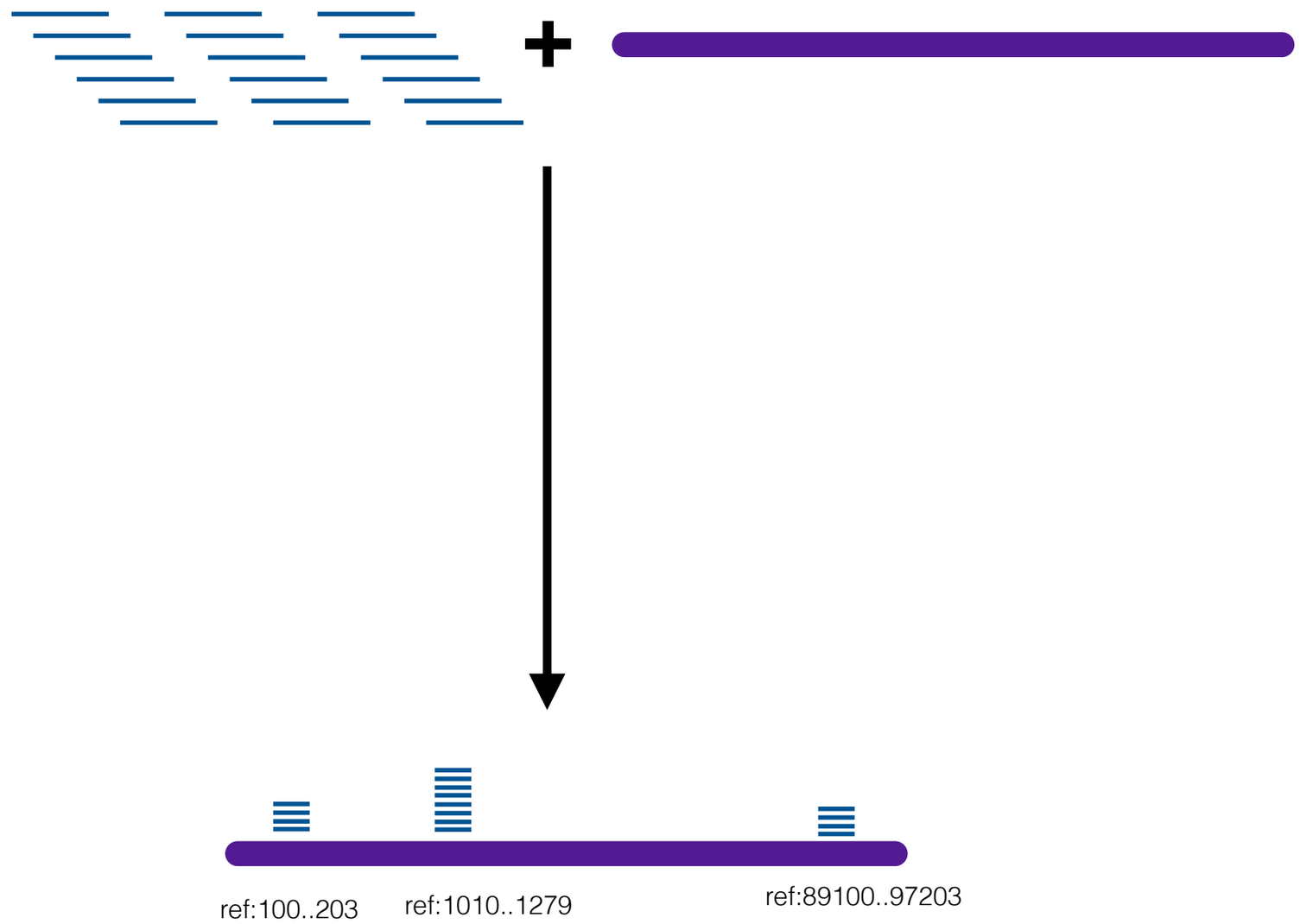
Sequence Alignment:
In bioinformatics, a sequence alignment is a way of arranging the sequences of DNA, RNA, or protein to identify regions of similarity that may be a consequence of functional, structural, or evolutionary relationships between the sequences. **Aligned sequences of nucleotide or amino acid residues are typically represented as rows within a matrix**.

http://en.wikipedia.org/wiki/Sequence_alignment

1. Basics of read mapping.

<u>Read Mapping Considerations</u>:

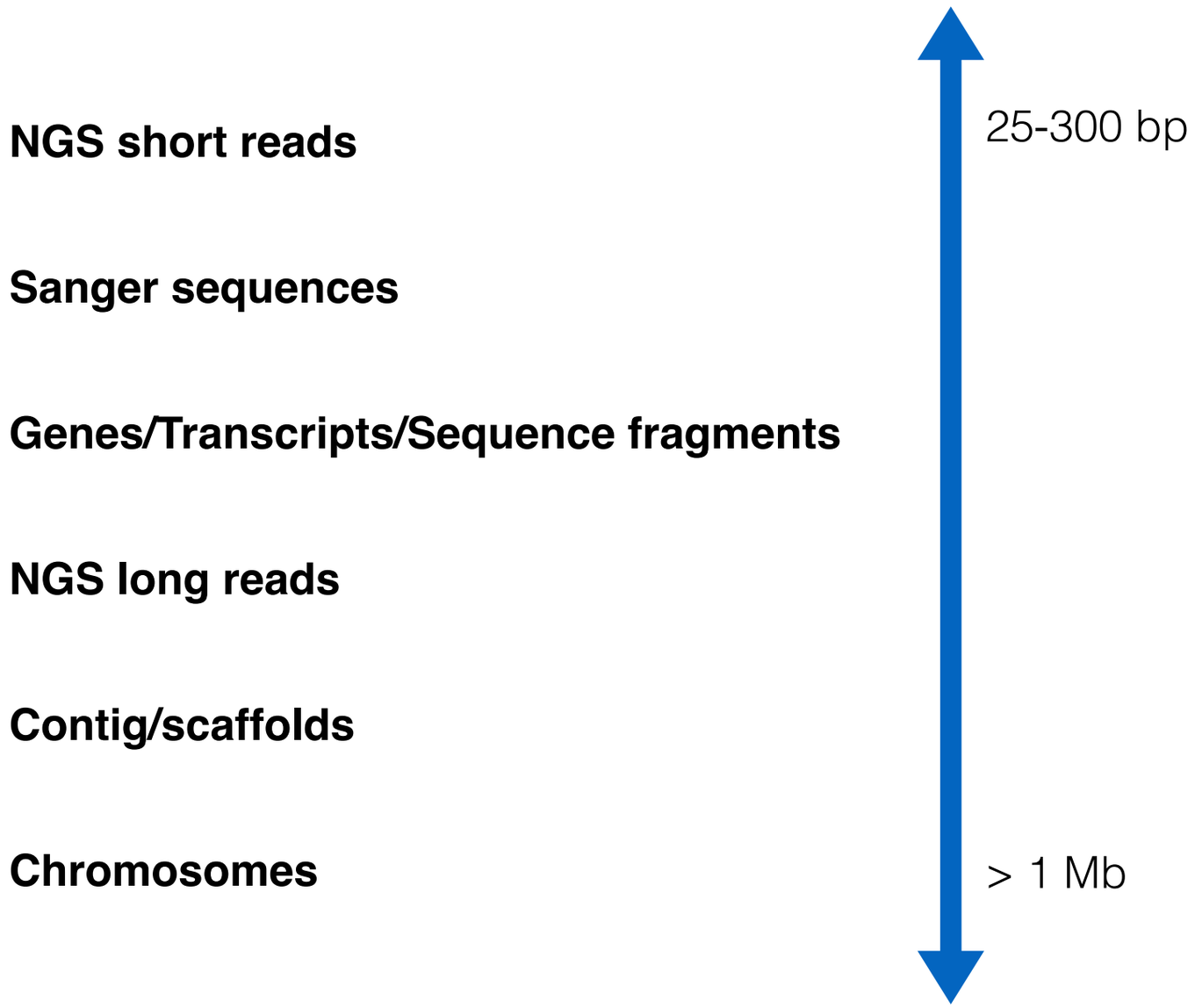
- Length of the read.
- Number of reads.
- Size of the reference.
- Differences with the reference



1. Basics of read mapping.

<u>Read Mapping Considerations</u>:

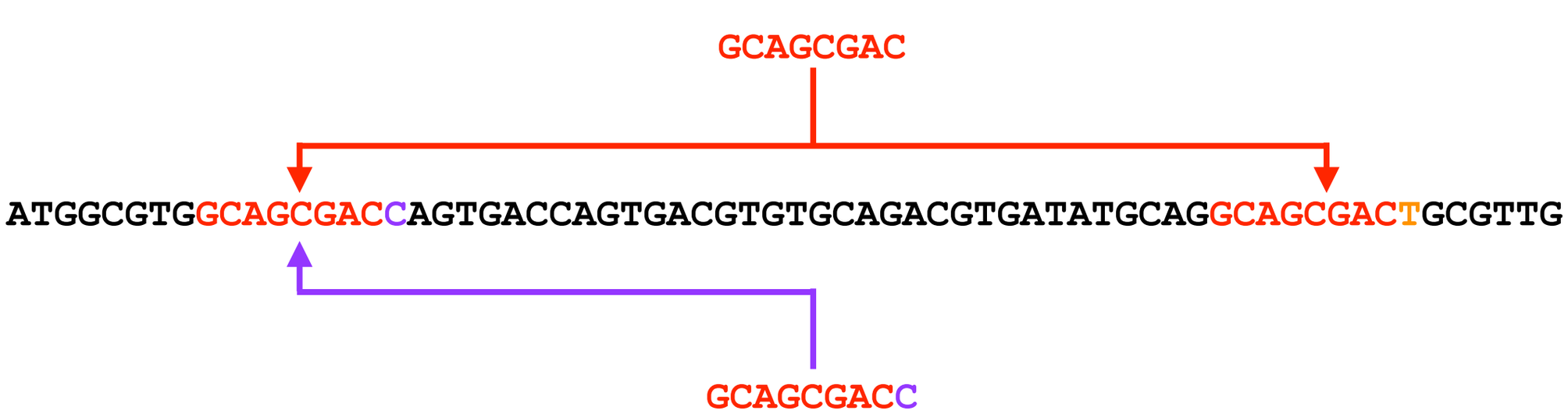
- Length of the read.
- Number of reads.
- Size of the reference.
- Differences with the reference



1. Basics of read mapping.

<u>Read Mapping Considerations</u>:

- Length of the read.
- Number of reads.
- Size of the reference.
- Differences with the reference

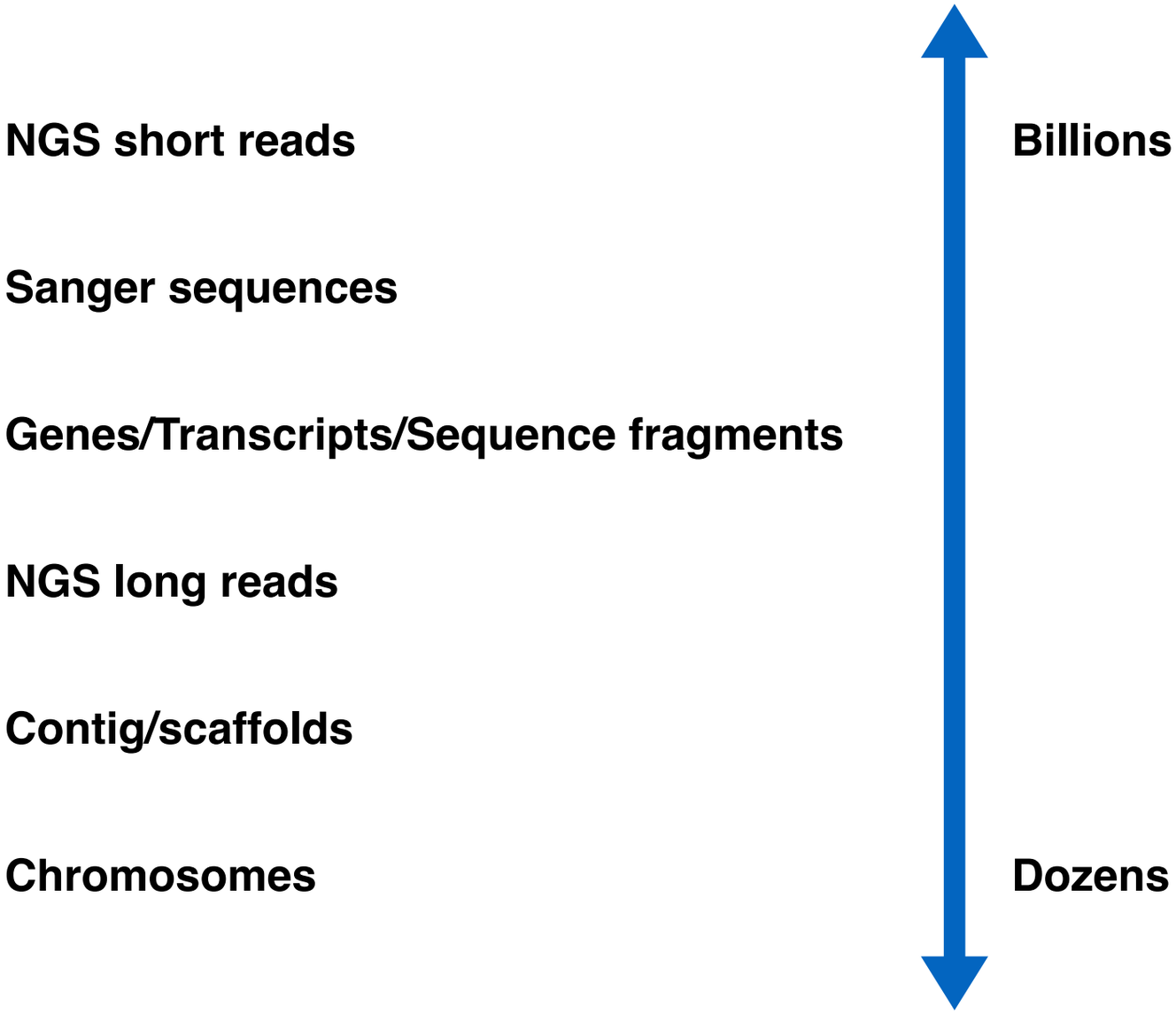


Shorter is the read, more possibilities of ambiguity

1. Basics of read mapping.

<u>Read Mapping Considerations</u>:

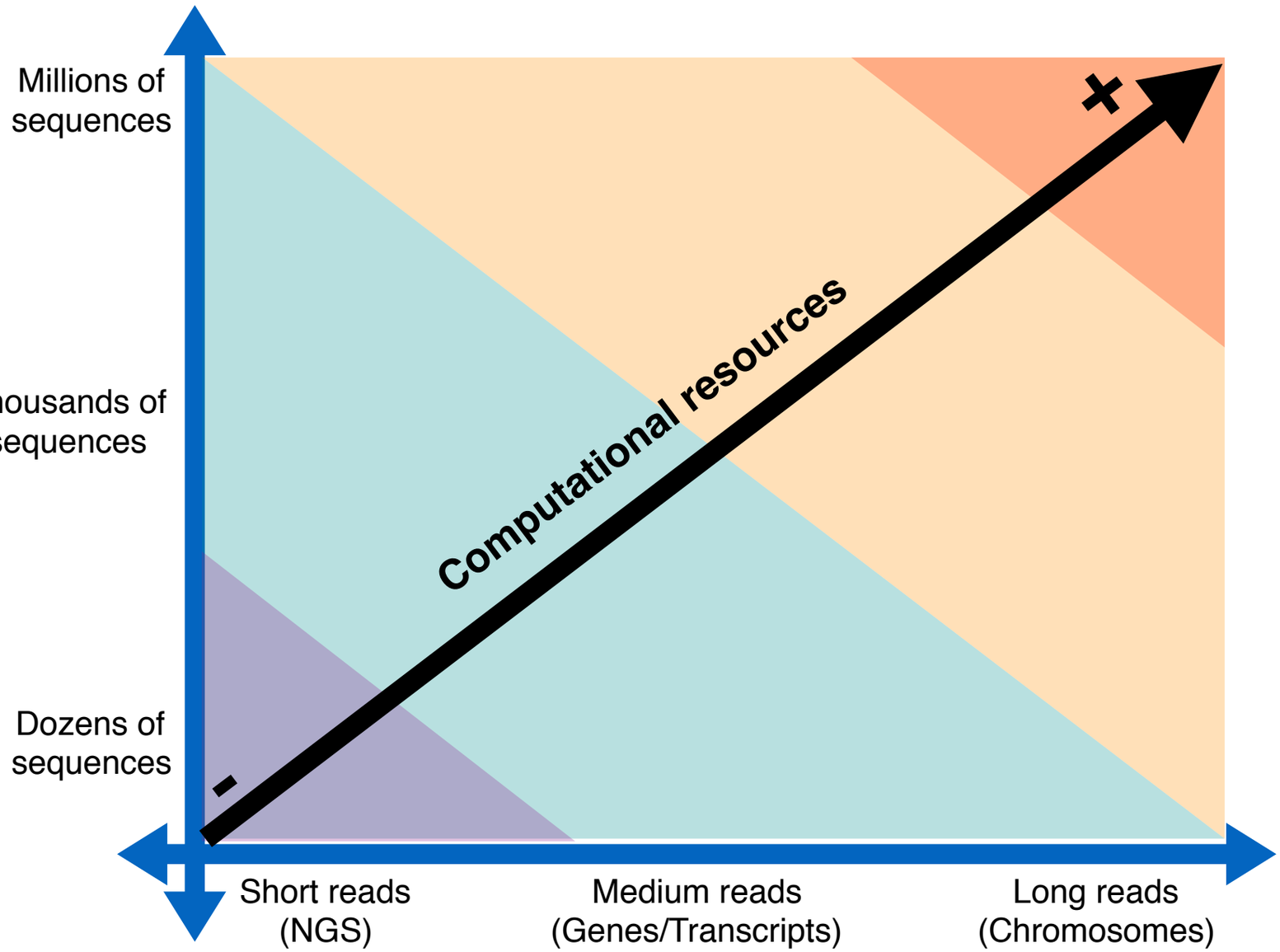
- Length of the read.
- Number of reads.
- Size of the reference.
- Differences with the reference



1. Basics of read mapping.

<u>Read Mapping Considerations</u>:

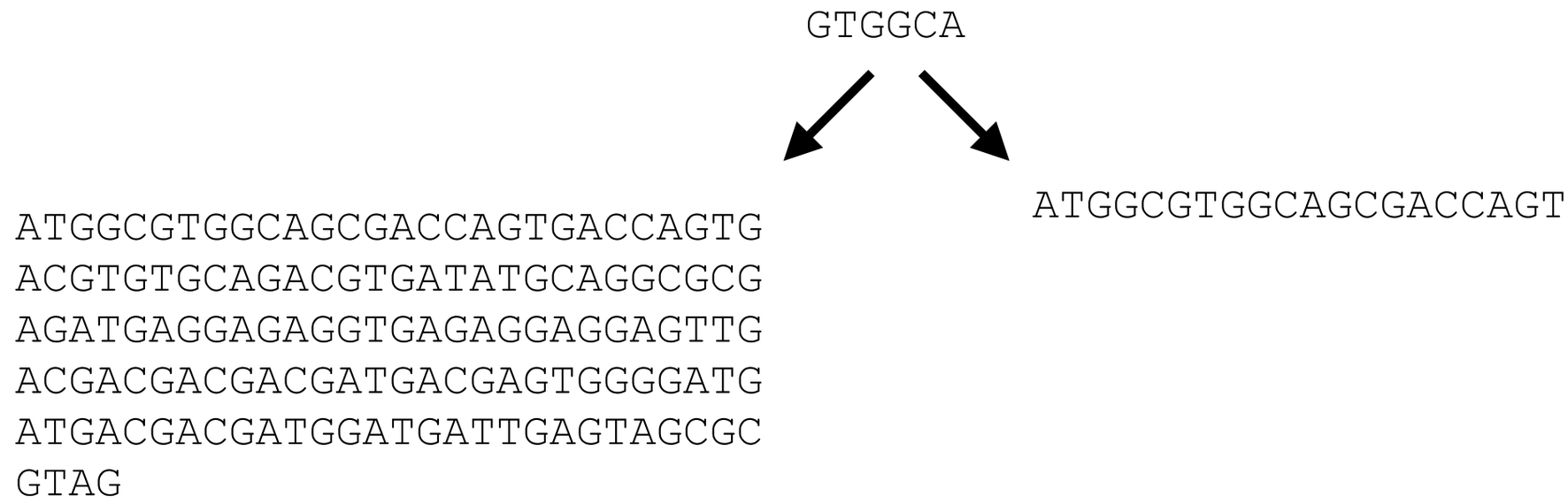
- Length of the read.
- Number of reads.
- Size of the reference.
- Differences with the reference



1. Basics of read mapping.

<u>Read Mapping Considerations</u>:

- Length of the read.
- Number of reads.
- Size of the reference.
- Differences with the reference



Smaller it is the reference, faster will be the search

1. Basics of read mapping.

<u>Read Mapping Considerations</u>:

- Length of the read.
- Number of reads.
- Size of the reference.
- Differences with the reference

Missmatches

```
GCAGCGACCAGTGACCA
||||||||||||  ||
GCAGCGACCAGCGA
```

Gaps

Reference

```
GCAGCGACCAG-GACCA
||||||||||||  ||
GCAGCGACCAGCGA
```

Read

```
GCAGCGACCAGTGACCA
||||||||||||  ||
GCAGCGACCAG-GA
```

1. Basics of read mapping.

<u>Read Mapping Considerations</u>:

- Length of the read.
- Number of reads.
- Size of the reference.
- Differences with the reference

```
GCAGCGACCAGTGACCA
||||||||||||  ||
GCAGCGACCAGCGA
```

```
GCAGTGACCAGTGACCA
||||  |||||||  ||
GCAGCGACCAGCGA
```

```
GGAGTGATCAGTGACCA
|  ||  ||  |||  ||
GCAGCGACCAGCGA
```

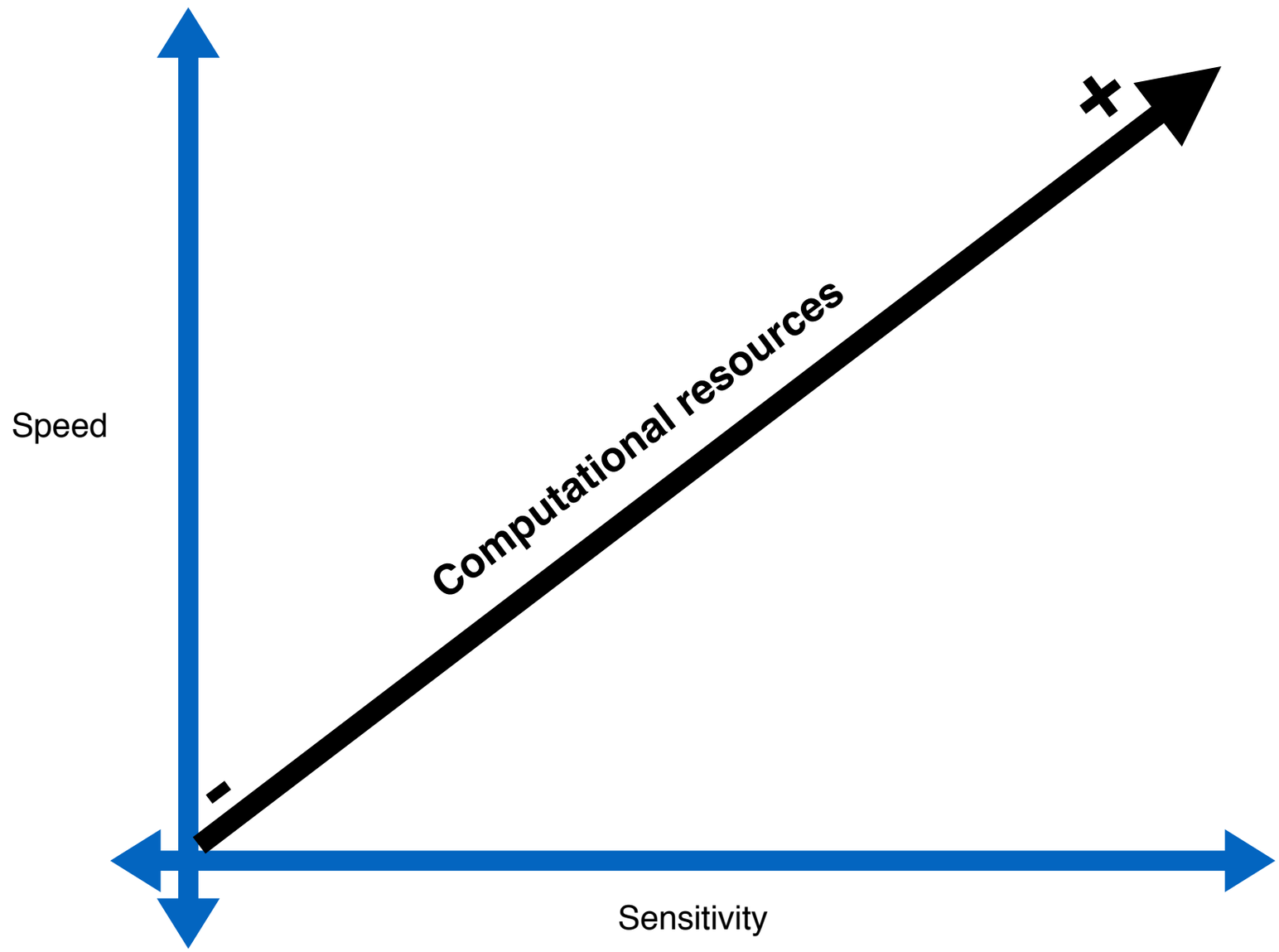
More differences with the reference, more difficult will be to align and to map

1. Basics of read mapping.

<u>Read Mapping Considerations</u>:

- Length of the read.
- Number of reads.
- Size of the reference.
- Differences with the reference

$$\textbf{MAPPING SENSITIVITY} = \frac{\textbf{Reads mapped correctly}}{\textbf{Reads mapped incorrectly}}$$



1. Basics of read mapping.

Read Mapping Methodologies:

- Hashing methods
- Suffix array methods

Reference (L=50)

ATGGCGTGGCAGCGACCAGTGACCAGTGACGTGTGCAGACGTGATATGCA

Read (L=23)

GACCAGTGACGTGTGCAGACCTG

Indexing with Kmer = 20 (31 Kmers)

```
ATGGCGTGGCAGCGACCAGT
TGGCGTGGCAGCGACCAGTG
GGCGTGGCAGCGACCAGTGA
GCGTGGCAGCGACCAGTGAC
CGTGGCAGCGACCAGTGACC
GTGGCAGCGACCAGTGACCA
TGGCAGCGACCAGTGACCAG
GGCAGCGACCAGTGACCAGT
GCAGCGACCAGTGACCAGTG
CAGCGACCAGTGACCAGTGA
AGCGACCAGTGACCAGTGAC
GCGACCAGTGACCAGTGACG
CGACCAGTGACCAGTGACGT
GACCAGTGACCAGTGACGTG
ACCAGTGACCAGTGACGTGT
CCAGTGACCAGTGACGTGTG
```

```
CAGTGACCAGTGACGTGTGC
AGTGACCAGTGACGTGTGCA
GTGACCAGTGACGTGTGCAG
TGACCAGTGACGTGTGCAGA
GACCAGTGACGTGTGCAGAC
ACCAGTGACGTGTGCAGACG
CCAGTGACGTGTGCAGACGT
CAGTGACGTGTGCAGACGTG
AGTGACGTGTGCAGACGTGA
GTGACGTGTGCAGACGTGAT
TGACGTGTGCAGACGTGATA
GACGTGTGCAGACGTGATAT
ACGTGTGCAGACGTGATATG
CGTGTGCAGACGTGATATGC
GTGTGCAGACGTGATATGCA
```

Perfect seed hit

Extension

Kmer = 20 (3 seeds)

GACCAGTGACGTGTGCAGAC
ACCAGTGACGTGTGCAGACC
CCAGTGACGTGTGCAGACCT

Extension algorithms

- Needleman-Wunsch
- Smith-Waterman
- FSA
- …

		A	T	G	A	C	G	T	G	C
	0	-1	-2	-3	-4	-5	-6	-7	-8	-9
G	-1	-1	-2	-1	-4	-5	-4	-7	-6	-9
C	-2	-2	-2	-3	-2	-3	-6	-5	-8	-5
C	-3	-3	-3	-3	-4	-1	-4	-7	-6	-7
G	-2	-4	-4	-2	-4	-5	0	-5	-6	-7
T	-3	-3	-3	-5	-3	-5	-6	1	-6	-7
G	-4	-4	-4	-2	-6	-4	-4	-7	2	-7
C	-5	-5	-5	-5	-3	-5	-5	-5	-8	3
T	-4	-6	-4	-6	-6	-4	-6	-4	-6	-9
G	-5	-5	-7	-3	-8	-7	-3	-7	-3	-7

1. Basics of read mapping.

<u>Read Mapping Methodologies</u>:

- Hashing methods
- Suffix array methods

Observations:
Small K, more sensitive but slower
High K, more specific and faster but bigger databases

Reference (L=50)

ATGGCGTGGCAGCGACCAGTGACCAGTGACGTGTGCAGACGTGATATGCA

Indexing with Kmer = 20 (31 Kmers)

```
ATGGCGTGGCAGCGACCAGT
TGGCGTGGCAGCGACCAGTG
GGCGTGGCAGCGACCAGTGA
GCGTGGCAGCGACCAGTGAC
CGTGGCAGCGACCAGTGACC
GTGGCAGCGACCAGTGACCA
TGGCAGCGACCAGTGACCAG
GGCAGCGACCAGTGACCAGT
GCAGCGACCAGTGACCAGTG
CAGCGACCAGTGACCAGTGA
AGCGACCAGTGACCAGTGAC
GCGACCAGTGACCAGTGACG
CGACCAGTGACCAGTGACGT
GACCAGTGACCAGTGACGTG
ACCAGTGACCAGTGACGTGT
CCAGTGACCAGTGACGTGTG
```

```
CAGTGACCAGTGACGTGTGC
AGTGACCAGTGACGTGTGCA
GTGACCAGTGACGTGTGCAG
TGACCAGTGACGTGTGCAGA
GACCAGTGACGTGTGCAGAC
ACCAGTGACGTGTGCAGACG
CCAGTGACGTGTGCAGACGT
CAGTGACGTGTGCAGACGTG
AGTGACGTGTGCAGACGTGA
GTGACGTGTGCAGACGTGAT
TGACGTGTGCAGACGTGATA
GACGTGTGCAGACGTGATAT
ACGTGTGCAGACGTGATATG
CGTGTGCAGACGTGATATGC
GTGTGCAGACGTGATATGCA
```

Read (L=23)

GACCAGTGACGTGTGCAGACCTG

Kmer = 20 (3 seeds)

GACCAGTGACGTGTGCAGAC
ACCAGTGACGTGTGCAGACC
CCAGTGACGTGTGCAGACCT

Perfect seed hit

Extension

Extension algorithms

- Needleman-Wunsch
- Smith-Waterman
- FSA
- …

		A	T	G	A	C	G	T	G	C
	0	-1	-2	-3	-4	-5	-6	-7	-8	-9
G	-1	-1	-2	-1	-4	-5	-4	-7	-6	-9
C	-2	-2	-2	-3	-2	-3	-6	-5	-8	-5
C	-3	-3	-3	-3	-4	-1	-4	-7	-6	-7
G	-2	-4	-4	-2	-4	-5	0	-5	-6	-7
T	-3	-3	-3	-5	-3	-5	-6	1	-6	-7
G	-4	-4	-4	-2	-6	-4	-4	-7	2	-7
C	-5	-5	-5	-5	-3	-5	-5	-5	-8	3
T	-4	-6	-4	-6	-6	-4	-6	-4	-6	-9
G	-5	-5	-7	-3	-8	-7	-3	-7	-3	-7

- Exercise 7.1: Decompose the following sequence in Kmers of 10-mers

ATGGCGTAGGTGACCAGTGA

1. Basics of read mapping.

Read Mapping Methodologies:

Burrows–Wheeler indexes

- Hashing methods
- Suffix array methods

The **Burrows–Wheeler transform (BWT, also called block-sorting compression) rearranges a character string into runs of similar characters**. This is useful for **compression**, since it tends to be easy to compress a string that has runs of repeated characters by techniques such as move-to-front transform and run-length encoding. More importantly, the transformation is reversible, without needing to store any additional data except the position of the first original character. The BWT is thus a "free" method of improving the efficiency of text compression algorithms, costing only some extra computation.

Transformation				
1. Input	2. All rotations	3. Sort into lexical order	4. Take the last column	5. Output
^BANANA|	^BANANA| |^BANANA A|^BANAN NA|^BANA ANA|^BAN NANA|^BA ANANA|^B BANANA|^	**A**NANA|^B **AN**A|^BAN **A**|^BANAN **B**ANANA|^ **N**ANA|^BA **N**A|^BANA ^BANANA| |^BANANA	ANANA|^**B** ANA|^BA**N** A|^BANA**N** BANANA|^ NANA|^B**A** NA|^BAN**A** ^BANANA| |^BANAN**A**	BNN^AA|A

https://en.wikipedia.org/wiki/Burrows%E2%80%93Wheeler_transform

1. Basics of read mapping.

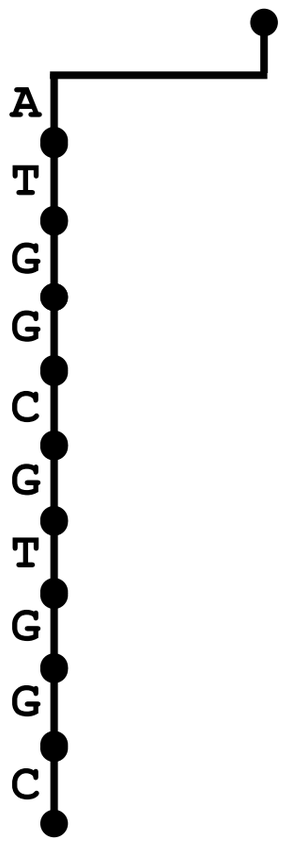
Read Mapping Methodologies:

- Hashing methods
- Suffix array methods

Reference (L=10)

ATGGCGTGGC

Build a suffix tree



1. Basics of read mapping.

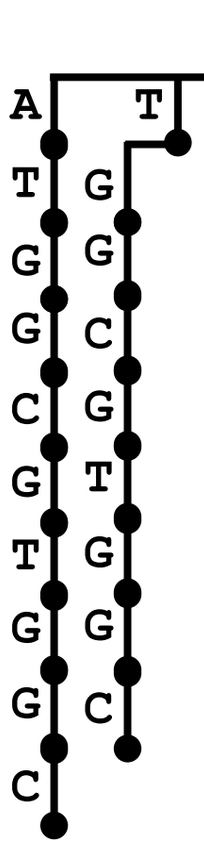
Read Mapping Methodologies:

- Hashing methods
- Suffix array methods

Reference (L=10)

ATGGCGTGGC

Build a suffix tree



1. Basics of read mapping.

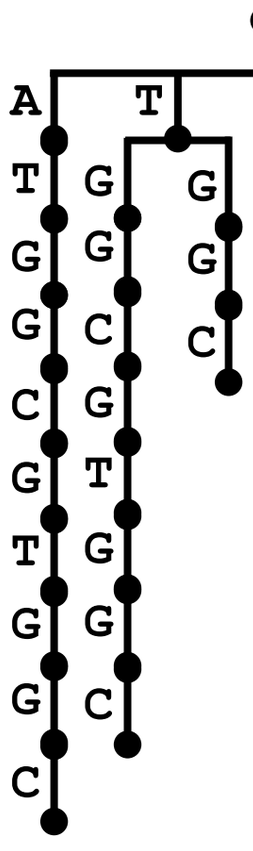
Read Mapping Methodologies:

- Hashing methods
- Suffix array methods

Reference (L=10)

ATGGCGTGGC

Build a suffix tree



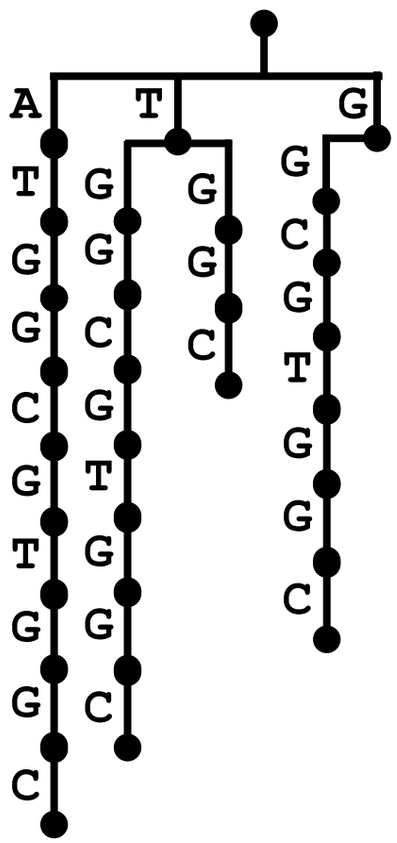
1. Basics of read mapping.

Read Mapping Methodologies:

- Hashing methods
- Suffix array methods

Reference (L=10)

Build a suffix tree



1. Basics of read mapping.

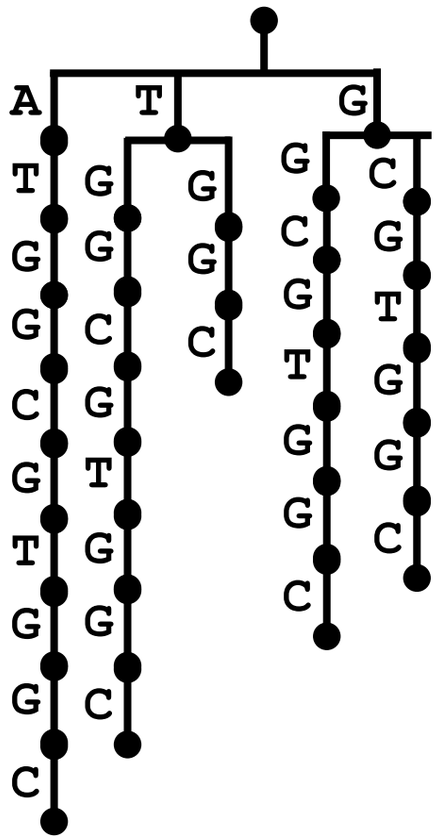
Read Mapping Methodologies:

- Hashing methods
- Suffix array methods

Reference (L=10)

ATGGCGTGGC

Build a suffix tree



1. Basics of read mapping.

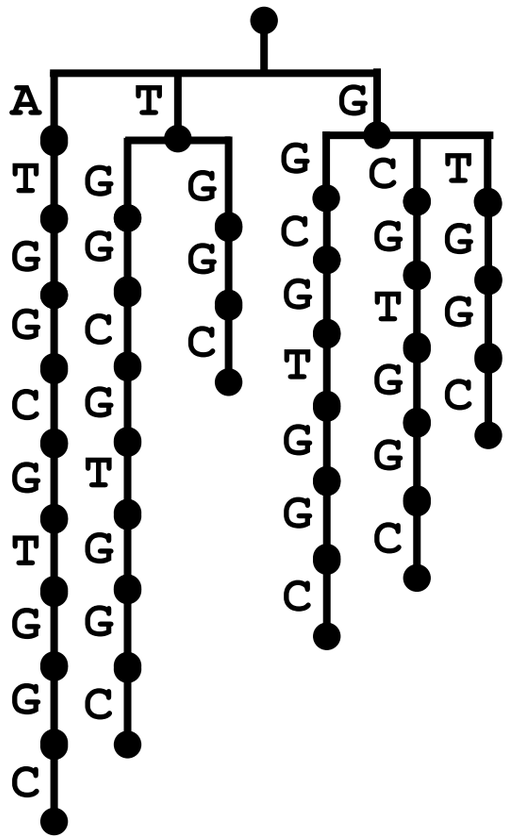
Read Mapping Methodologies:

- Hashing methods
- Suffix array methods

Reference (L=10)

ATGGCGTGGC

Build a suffix tree



1. Basics of read mapping.

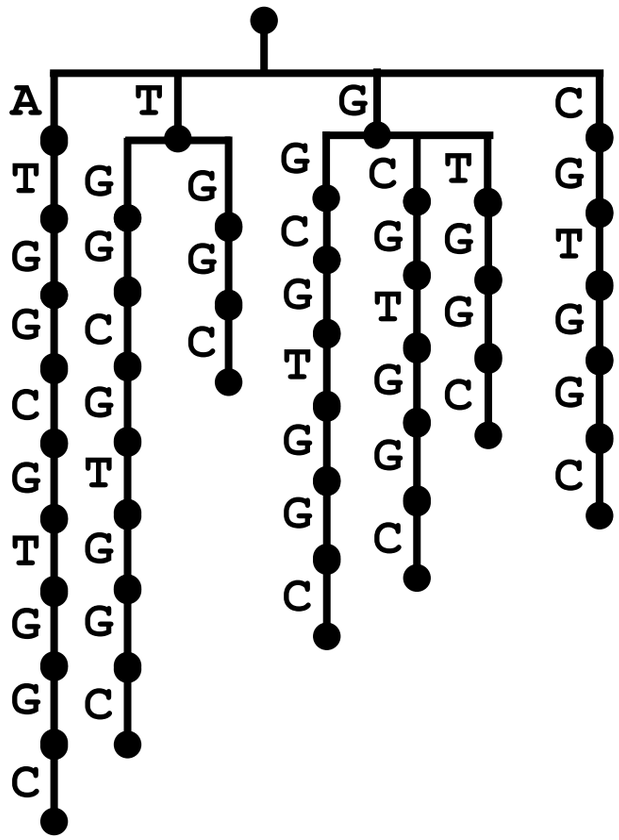
Read Mapping Methodologies:

- Hashing methods
- Suffix array methods

Reference (L=10)

ATGGCGTGGC

Build a suffix tree



1. Basics of read mapping.

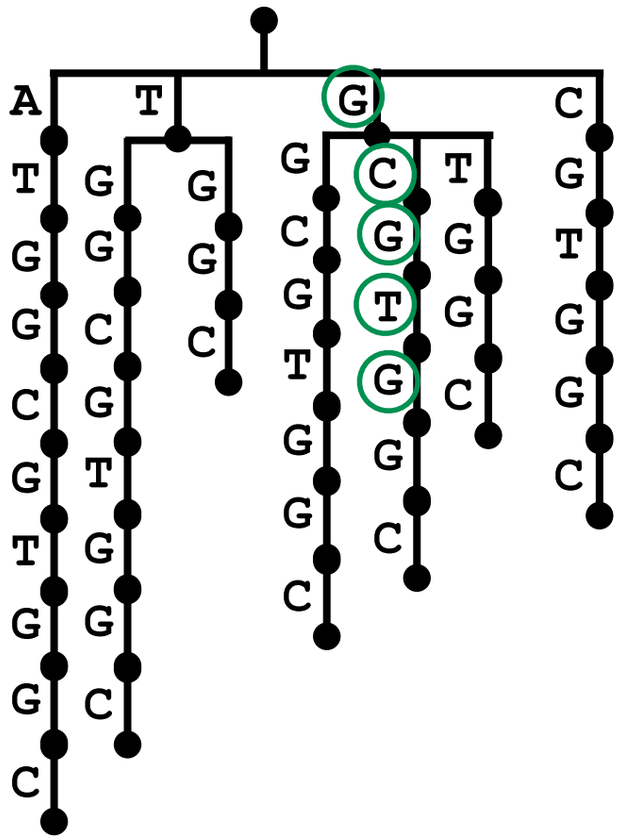
Read Mapping Methodologies:

- Hashing methods
- Suffix array methods

Reference (L=10)

ATGGCGTGGC

Build a suffix tree



Read (L=5)

GCGTG

Follow the tree

1. Basics of read mapping.

<u>Read Mapping Methodologies</u>:

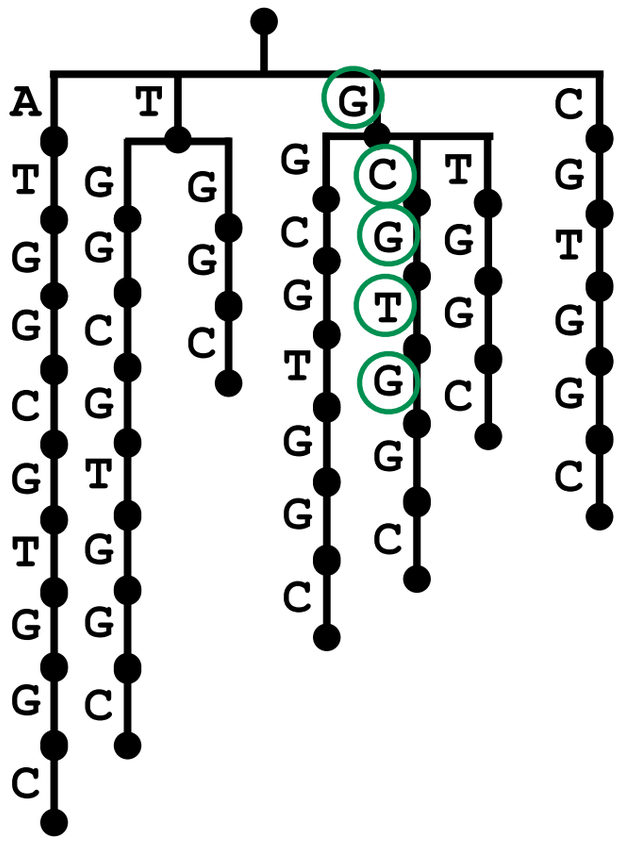
- Hashing methods
- Suffix array methods

Observations:
Very fast
With errors the search space increase quickly

Reference (L=10)

ATGGCGTGGC

Build a suffix tree



Read (L=5)

GCGTG

Follow the tree

- Exercise 7.2: Build a suffix tree with the following sequence

ATGGCGTCGGT

Outline of Topics

1. Basics of read mapping.

2. Short read mapping.

3. Long read mapping.

4. Mapping of transcriptomic reads.

5. Visualization of mapped reads.

6. Uses and analysis.

2. Short read mapping.

Read Mapping Software:

Name	Type	Input	Output
Bowtie	Short sequences	Fasta, Fastq	Sam
BWA	Short sequences	Fasta, Fastq	Sam
Novoalign	Short sequences	Fasta, Fastq	Sam
SOAP	Short sequences	Fasta, Fastq	Sam
Stampy	Short sequences	Fasta, Fastq	Sam

2. Short read mapping.

Read Mapping Software:

Fast gapped-read alignment with Bowtie 2

Ben Langmead & Steven L Salzberg

Affiliations | Contributions | Corresponding author

Nature Methods **9**, 357–359 (2012) | doi:10.1038/nmeth.1923

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

```
Usage:
  bowtie [options]* <ebwt> {-1 <m1> -2 <m2> | --12 <r> | <s>} [<hit>]

<m1>    Comma-separated list of files containing upstream mates (or the
        sequences themselves, if -c is set) paired with mates in <m2>
<m2>    Comma-separated list of files containing downstream mates (or the
        sequences themselves if -c is set) paired with mates in <m1>
<r>     Comma-separated list of files containing Crossbow-style reads.  Can be
        a mixture of paired and unpaired.  Specify "-" for stdin.
<s>     Comma-separated list of files containing unpaired reads, or the
        sequences themselves, if -c is set.  Specify "-" for stdin.
<hit>   File to write hits to (default: stdout)
```

2. Short read mapping.

Read Mapping Software:

III. SAM/BAM

SAM (and its binary form BAM) format is designed to store read mapping information to a reference. It has 11 columns.

Alignment

```
Coor     12345678901234  5678901234567890123456789012345
ref      AGCATGTTAGATAA**GATAGCTGTGCTAGTAGGCAGTCAGCGCCAT

+r001/1        TTAGATAAAGGATA*CTG
+r002        aaaAGATAA*GGATA
+r003      gcctaAGCTAA
+r004                  ATAGCT..............TCAGC
-r003
-r001/2
```

Sam file

```
@HD VN:1.3 SO:coordinate
@SQ SN:ref LN:45
r001  163  ref   7  30  8M2I4M1D3M  =  37   39  TTAGATAAAGGATACTG  *
r002    0  ref   9  30  3S6M1P1I4M  *   0    0  AAAAGATAAGGATA     *
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           *
```

Col	Field
1	QNAME
2	FLAG
3	RNAME
4	POS
5	MAPQ
6	CIGAR
7	RNEXT
8	PNEXT
9	TLEN
10	SEQ
11	QUAL

2. Short read mapping.

Read Mapping Software:

III. SAM/BAM

SAM (and its binary form BAM) format is designed to store read mapping information to a reference. It has 11 columns.

The 2nd column: FLAG defines the status of the read mapping.

Col	Field
1	QNAME
2	FLAG
3	RNAME
4	POS
5	MAPQ
6	CIGAR
7	RNEXT
8	PNEXT
9	TLEN
10	SEQ
11	QUAL

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

- Flag = 4 means 0x4 read unmapped
- Flag = 16 means 0x10 read reverse strand
- Flag = 83 means 0x1 read paired, 0x2 read mapped proper pair, 0x10 read reverse strand and 0x40 first in pair

http://picard.sourceforge.net/explain-flags.html

- Exercise 7.3: Translate the following SAM flags

  - f = 5
  - f = 73
  - f = 161
  - f = 289

2. Short read mapping.

Sam/Bam file manipulation:

Oxford Journals › Science & Mathematics › Bioinformatics › Volume 25, Issue 16 › Pp. 2078-2079.

The Sequence Alignment/Map format and SAMtools

Heng Li[1,†], Bob Handsaker[2,†], Alec Wysoker[2], Tim Fennell[2], Jue Ruan[3], Nils Homer[4], Gabor Marth[5], Goncalo Abecasis[6], Richard Durbin[1,*] and 1000 Genome Project Data Processing Subgroup[7]

+ Author Affiliations

« Previous | Next Article »
Table of Contents

This Article

Bioinformatics (2009) 25 (16): 2078-2079.
doi: 10.1093/bioinformatics/btp352
First published online: June 8, 2009

http://samtools.sourceforge.net/

```
Usage:   samtools <command> [options]

Command: view        SAM<->BAM conversion
         sort        sort alignment file
         mpileup     multi-way pileup
         depth       compute the depth
         faidx       index/extract FASTA
         tview       text alignment viewer
         index       index alignment
         idxstats    BAM index stats (r595 or later)
         fixmate     fix mate information
         flagstat    simple stats
         calmd       recalculate MD/NM tags and '=' bases
         merge       merge sorted alignments
         rmdup       remove PCR duplicates
         reheader    replace BAM header
         cat         concatenate BAMs
         bedcov      read depth per BED region
         targetcut   cut fosmid regions (for fosmid pool only)
         phase       phase heterozygotes
         bamshuf     shuffle and group alignments by name
```

Outline of Topics

1. Basics of read mapping.

2. Short read mapping.

3. Long read mapping.

4. Mapping of transcriptomic reads.

5. Visualization of mapped reads.

6. Uses and analysis.

3. Long read mapping.

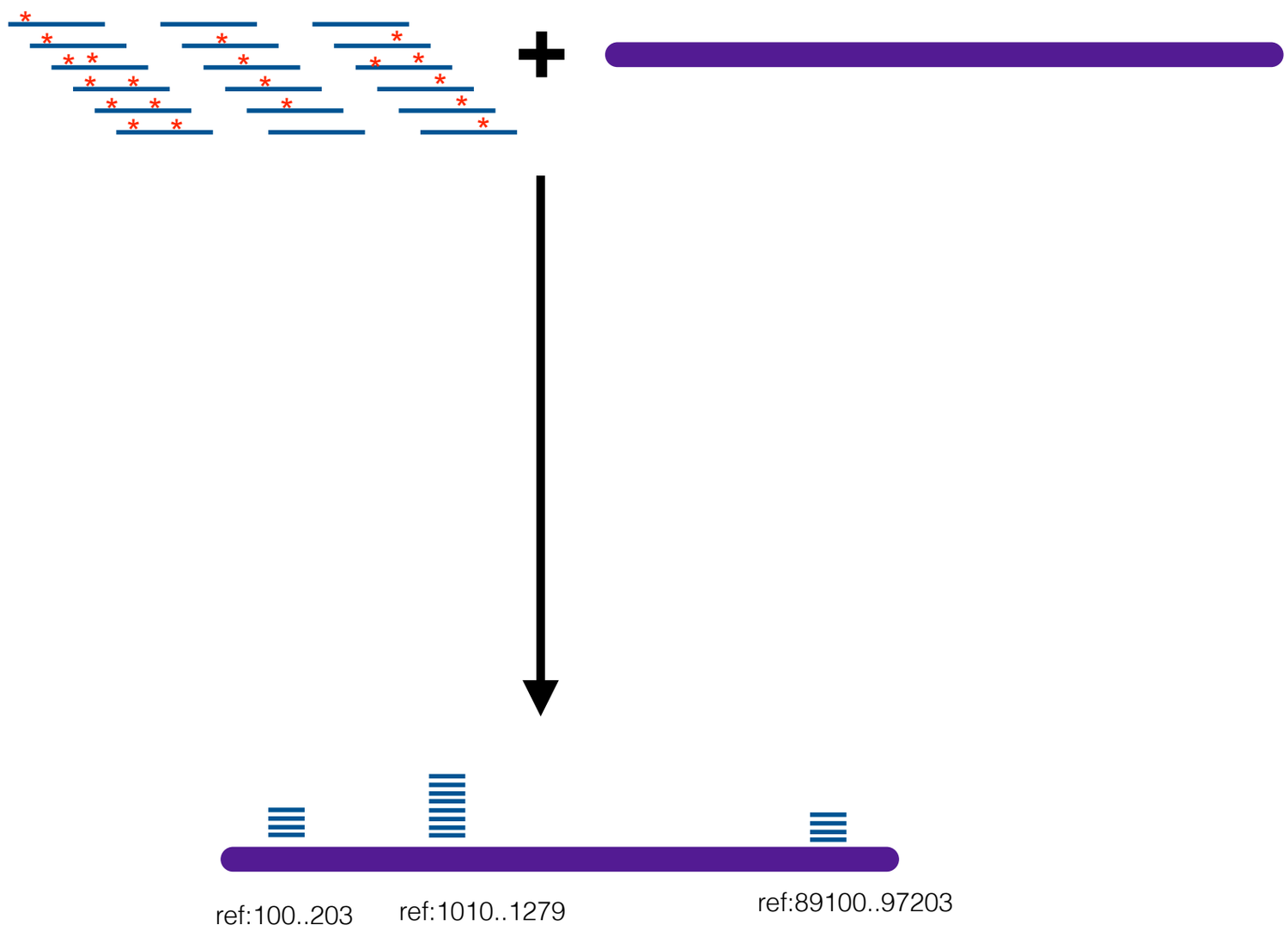
Long Read Mapping (Traditional tools):

Name	Type	Input	Output
Mauve	Long sequences	Fasta, GenBank	backbone (positions) XMFA (alignments)
LastZ/MultiZ	Long sequences	Fasta	several (maf, sam…)
Blast	Medium sequences	Fasta	Blast formats (0 text file, 6 tabular file)
Blat	Medium sequences	Fasta	Blast formats + Blat tabular format

3. Long read mapping.

Long Read Mapping:

Long reads produced by NGS are “noisy" (full or errors).



3. Long read mapping.

Long Read Mapping Software:

Name	Type	Input	Output
BlasR	Long sequences	Fasta, Fastq	Sam
GMap	Long sequences	Fasta, Fastq	Sam
Minimap2	Long sequences	Fasta, Fastq	Sam
NGMLR	Long sequences	Fasta, Fastq	Sam
LordFAST	Long sequences	Fasta, Fastq	Sam

Outline of Topics

1. Basics of read mapping.

2. Short read mapping.

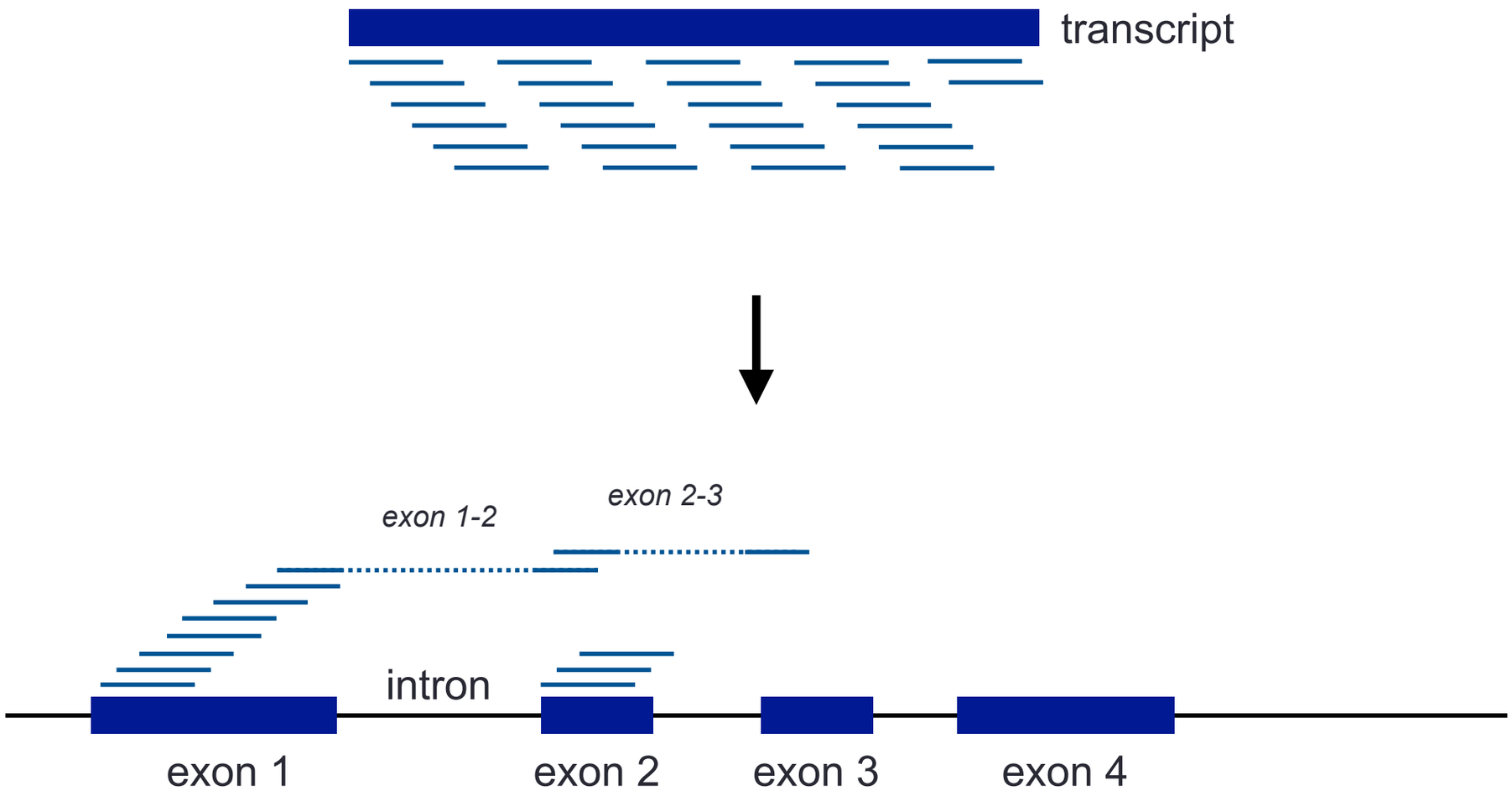
3. Long read mapping.

4. Mapping of transcriptomic reads.

5. Visualization of mapped reads.

6. Uses and analysis.

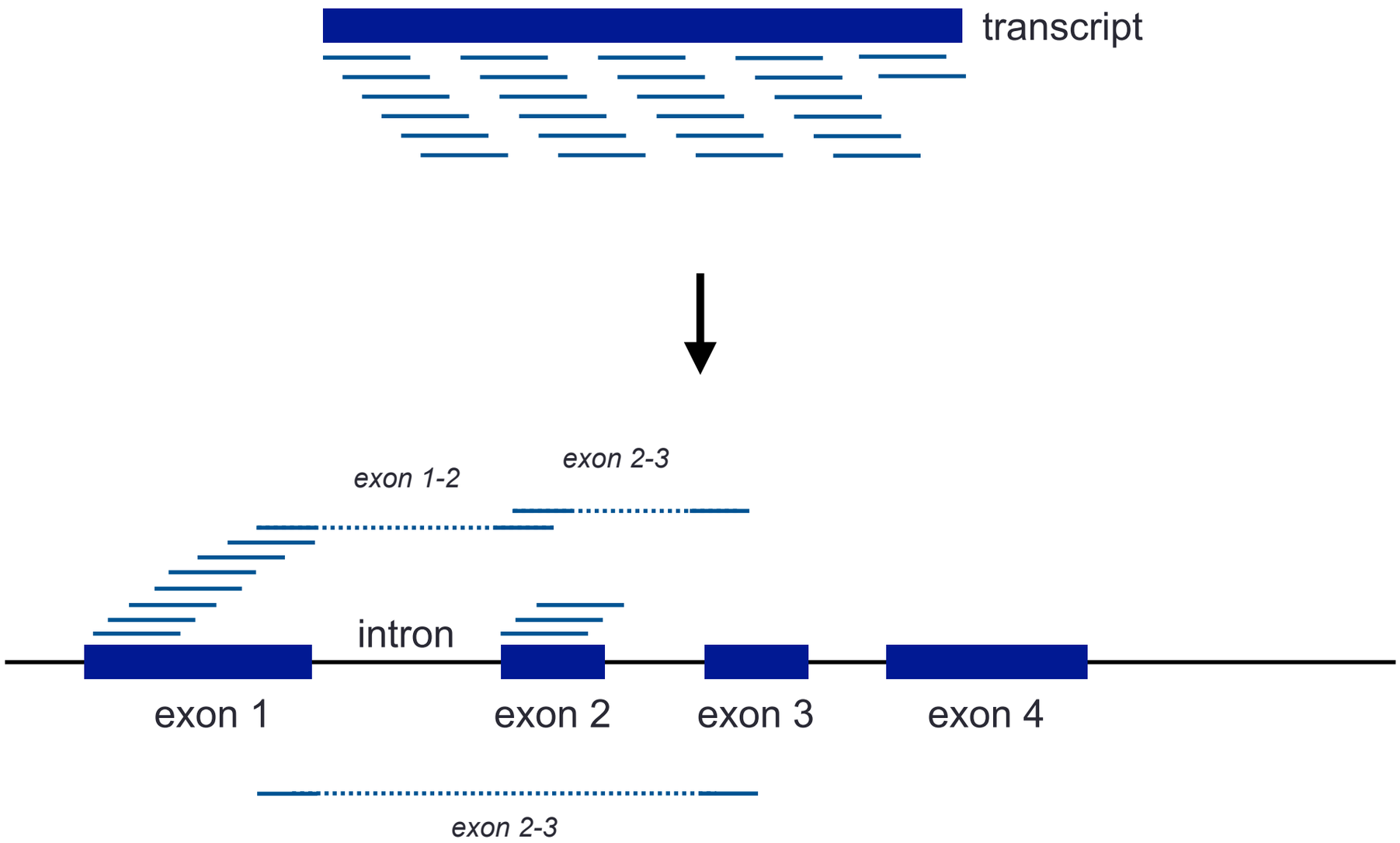
4. Mapping of transcriptomic reads.



Transcriptome read alignment needs total into account:

- **Intron-exon boundaries**

4. Mapping of transcriptomic reads.



Transcriptome read alignment needs total into account:

- **Intron-exon boundaries**
- **Alternative Splicings**

4. Mapping of transcriptomic reads.

Read Mapping Software for Transcriptome:

Name	Type	Input	Output
Tophat2	Short sequences	Fasta, Fastq	Sam
Hisat2	Short sequences	Fasta, Fastq	Sam
STAR	Short sequences	Fasta, Fastq	Sam
Salmon	Short sequences	Fasta, Fastq	Sam

4. Mapping of transcriptomic reads.

Read Mapping Software for Transcriptome:

Protocol | Published: 11 August 2016

Transcript-level expression analysis of RNA-seq experiments with HISAT, StringTie and Ballgown

Mihaela Pertea, Daehwan Kim, Geo M Pertea, Jeffrey T Leek & Steven L Salzberg

Nature Protocols **11**, 1650–1667(2016) | Cite this article

Outline of Topics

1. Basics of read mapping.

2. Short read mapping.

3. Long read mapping.

4. Mapping of transcriptomic reads.

5. Visualization of mapped reads.

6. Uses and analysis.

5. Visualization of mapped reads.

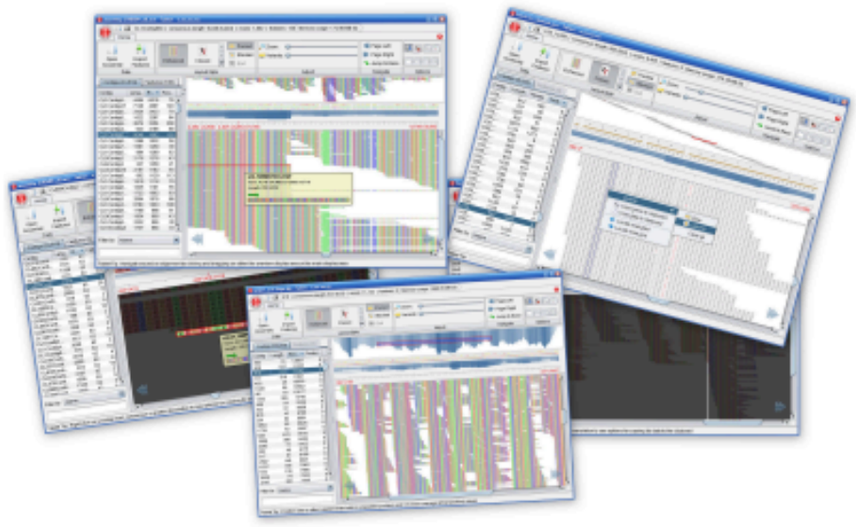
Tablet

https://ics.hutton.ac.uk/tablet/

Tablet is a lightweight, high-performance graphical viewer for next generation sequence assemblies and alignments.

Tablet

Tablet is a lightweight, high-performance graphical viewer for next generation sequence assemblies and alignments.



5. Visualization of mapped reads.

IGV, Integrative Genomic Viewer

http://software.broadinstitute.org/software/igv/

The Integrative Genomics Viewer (IGV) is a high-performance visualization tool for interactive exploration of large, integrated genomic datasets. It supports a wide variety of data types, including array-based and next-generation sequence data, and genomic annotations.

Downloads

Integrative Genomics Viewer - IGV 2.4

Install IGV

Use one of the following 4 options to install and run the current version of IGV. ***NOTE: IGV 2.4.x requires Java 8 or later.***

1. Download and unzip the **Mac App Archive**, then double-click the IGV application to run it. The application can be moved to the *Applications* folder, or anywhere else.

2. Download and unzip the **Windows Zip Archive**, then double-click the *igv.bat* file to start IGV. A black console window will appear, followed by the IGV application. **Note**: Windows users with **high resolution screens** should use this version -- it includes a modified Java executable for use with high-resolution screens.

3. Download and unzip the **Binary Distribution archive**. IGV is launched from a command prompt -- follow the instructions in the *readme* file. To launch IGV on Mac or Linux use the shell script *igv.sh*. On Windows use *igv.bat*.

4. Click on one of the *Launch* buttons below to download a .jnlp file and execute the file using **Java Web Start** (JWS). ***NOTE: this option does not work with Java 9.***

- **Mac users**: If you are notified of security errors that prevent launching IGV, try the following:
  - Right-click on the downloaded .jnlp file; select *Open With > Java Web Start;* dismiss the warnings.
  - After IGV has been run this way at least once from the .jnlp file, you can double-click on the file to launch.
- **Windows users:** To run with more than 1.2 GB of memory on Windows you must install 64-bit Java. ***Most Windows installs do not include 64-bit Java by default, even if the operating system is 64-bit***. Attempting to use the 2GB or greater launch options with 32-bit Java will result in the error *"could not create virtual machine"*.

Launch	Launch	Launch	Launch
Launch with **750 MB**	Launch with **1.2 GB** (Max usable memory for Windows with 32-bit Java)	Launch with **2 GB** (Max usable memory for 32-bit MacOS)	Launch with **10 GB** (Only for large memory machines with 64-bit Java)

Outline of Topics

1. Basics of read mapping.

2. Short read mapping.

3. Long read mapping.

4. Mapping of transcriptomic reads.

5. Visualization of mapped reads.

6. Uses and analysis.

Transcriptomic analysis

Epigenetic Analysis

Comparative genomics

. . .