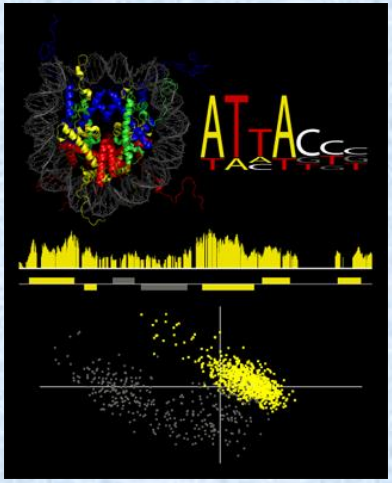
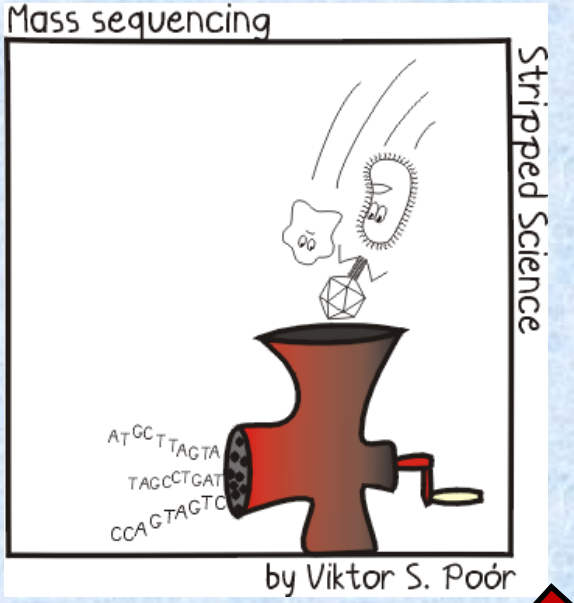
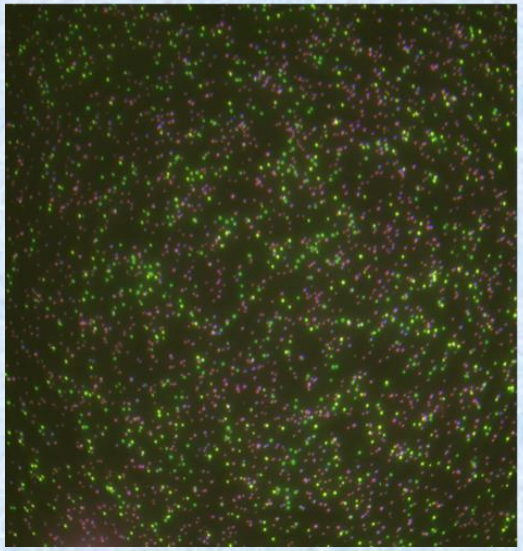

# MAPPING OR ALIGNMENT OF SHORT READS

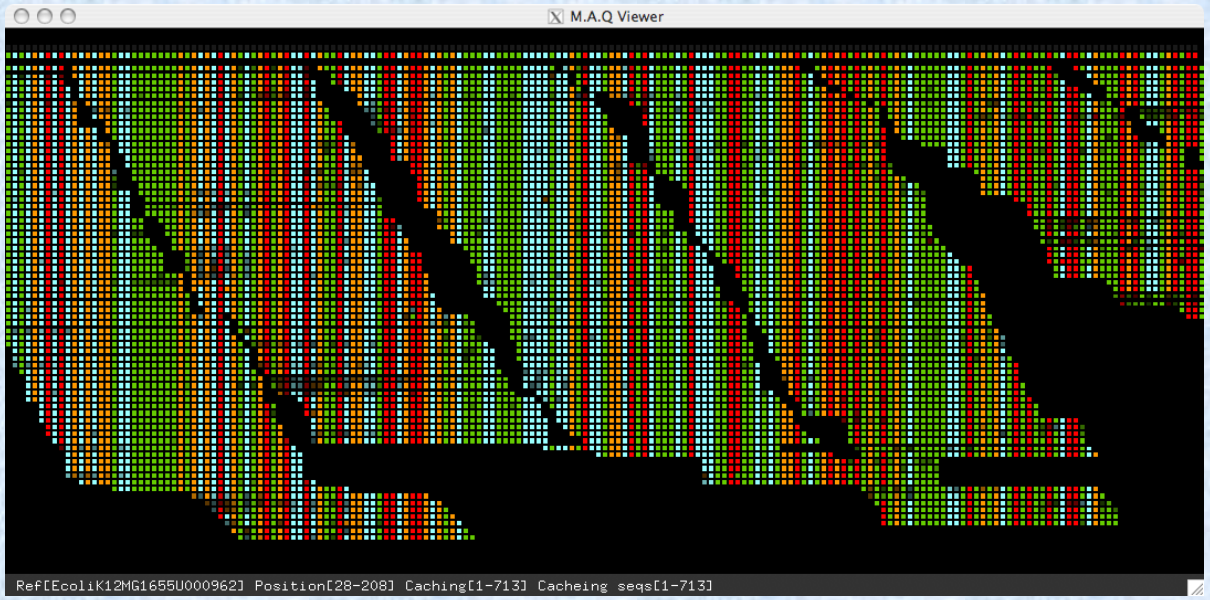


Ester Feldmesser

November 2019



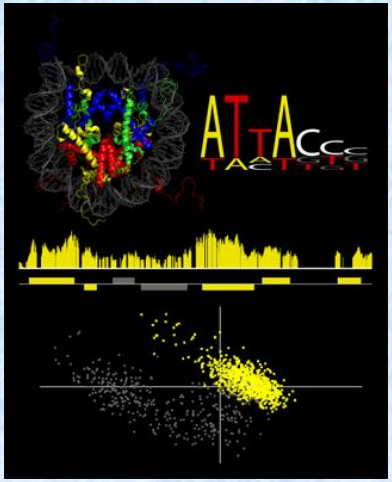






https://www.wiki.ed.ac.uk/download/attachments/36021339/maqscreenshot.png

# Outline


- Introduction: alignment
- Local and global alignments
- Blast
- Needs of NGS
- Short reads alignment
  - Burrows Wheeler transform
- Transcriptome alignment



# Sequence output format

FASTQ format is a text-based format for storing both a biological sequence (usually nucleotide sequence) and its corresponding quality scores. (From Wikipedia)

Line 1: Unique ID for a sequencing read
Line 2: Sequences
Line 3:+
Line 4: Base calling quality score (Analogous to Phred scores but in ASCII value)

Example:

```
@HISEQ:126:H14YJADXX:1:1101:1118:2101 1:N:0:ATCACG
CTCCATAGTCAGAAACTTCAGCATGACAGTACCTCATGCTGCATCAGGTGATCATGAAAAGATTAC
+
@@?ADDDD?ADHDIIIIIIIEIIIGEFHC<?FH4C9E9BGAFIGH<DG9BD?@DGGEGHHG<DCBB
```

https://emea.support.illumina.com/bulletins/2016/04/fastq-files-explained.html

# Sequence alignment

A **sequence alignment** is a way of arranging the sequences of DNA, RNA, or protein to identify regions of similarity. Aligned sequences of nucleotide or amino acid residues are typically represented as rows within a matrix.
Gaps are inserted between the residues so that identical or similar characters are aligned in successive columns.

```
AAB24882        TYHMCQFHCRYVNNHSGEKLYECNERSKAFSCPSHLQCHKRRQIGEKTHEHNQCGKAFPT 60
AAB24881        --------------------YECNQCGKAFAQHSSLKCHYRTHIGEKPYECNQCGKAFSK 40
                                    ****: .***:  * *:** * :****.:* *******..

AAB24882        PSHLQYHERTHTGEKPYECHQCGQAFKKCSLLQRHKRTHTGEKPYE-CNQCGKAFAQ- 116
AAB24881        HSHLQCHKRTHTGEKPYECNQCGKAFSQHGLLQRHKRTHTGEKPYMNVINMVKPLHNS 98
                 **** *:***********:***:**.: .***************    :  *.: :
```

A sequence alignment, produced by ClustalW, of two human zinc finger proteins

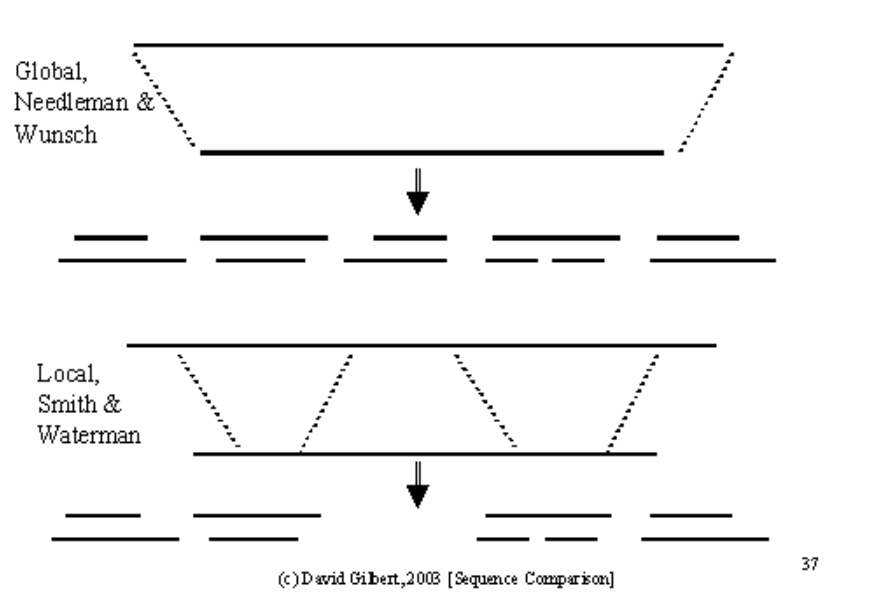
# Alignment coverage


→ Reference sequence

What software for alignment do you know?

# Local and global alignments

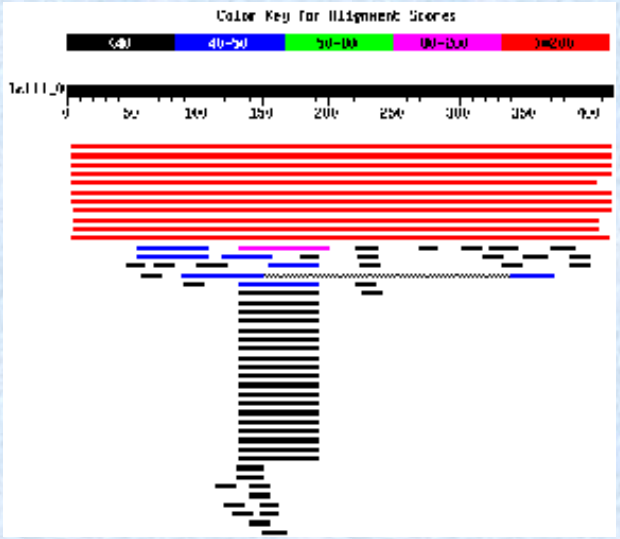
- **Global alignments** attempt to align every residue in every sequence
- Useful when the sequences in the query set are similar and of roughly equal size
- The FASTA software performs global alignments




- **Local alignments** look for regions of similarity or similar sequence motifs within their larger sequence context
- Useful for dissimilar sequences
- BLAST performs local alignment

# Blast (Basic Local Alignment Search Tool)

- Locates **short matches** (common words)
- Searches for high scoring sequence alignments between these matches in the database using a heuristic approach that approximates the Smith-Waterman algorithm
- BLAST algorithm uses a heuristic approach that is less accurate than the Smith-Waterman but over 50 times faster.
- In typical usage, the query sequence is much smaller than the database (query ~1000 nucleotides and the database ~ several billion nucleotides.



# Needs for NGS

- Many of the next-generation sequencing projects begin with a known, or so-called ‘reference’, genome. This process is known as aligning or ‘mapping’ the read to the reference.
- There are many programs that perform both spliced and unspliced alignment for the older Sanger-style capillary reads.
- These programs neither scale up to the much greater volumes of data produced by short read sequencers nor scale down to the short read lengths.
- Aligning the reads from ChIP-Seq or RNA-Seq experiments can take hundreds or thousands of central processing unit (CPU) hours using conventional software tools such as BLAST or BLAT.

*Nature Biotechnology* **27**, 455 - 457 (2009)

# Comparison between generations of alignment programs

| Blast | New programs |
| --- | --- |
| Alignment of a protein to large databases | Alignment of DNA reads within the genome sequenced |
| Alignment of one or a few sequences | Alignment of millions of reads |
| Usually the sequence is long | Sequence reads are short |
| To find some degree of homology | To find almost perfect matches (up to two mismatches, short indels) |
| Permits gaps | Take into account splice junctions |
| Algorithms not fully optimized | Algorithms optimized for speed and memory |

# Short read aligners: Differences

Alignment tools differ in

- speed
- suitability for use on computer clusters
- memory requirements
- accuracy
    - Is a good match always found?
    - What is the maximum number of allowed mismatches?
- ease of use

http://marray.economia.unimi.it/2009/material/lectures/L5_HTS_Overview.pdf

# Short read aligners: Differences

Alignment tools also differ in whether they can

- make use of base-call quality scores
- estimate alignment quality
- detect small indels
- report multiple matches
- work with longer than normal reads
- match in color space (for SOLiD systems)
- align data from methylation experiments
- deal with splice junctions

http://marray.economia.unimi.it/2009/material/lectures/L5_HTS_Overview.pdf

# Short read alignment: Algorithms

Short-read aligners use indexing to speed up mapping

- use the Burrows-Wheeler transform (BWT)

BWT seems to be the winning idea (very fast, sufficiently accurate), and is used by the newest tools (Bowtie, SOAPv2, BWA, Hisat).

http://marray.economia.unimi.it/2009/material/lectures/L5_HTS_Overview.pdf

# Burrows–Wheeler transform

- **Burrows–Wheeler transform** (**BWT**, also called **block-sorting compression**), is an algorithm used in data compression techniques.
- It was invented by Michael Burrows and David Wheeler in 1994 while working at DEC Systems Research Center in Palo Alto, California.
- When a character string is transformed by the BWT, none of its characters change value.
- The transformation permutes the order of the characters.
- If the original string had several substrings that occurred often, then the transformed string will have several places where a single character is repeated multiple times in a row.
- This is useful for compression.

http://en.wikipedia.org/wiki/Burrows–Wheeler_transform

# Example

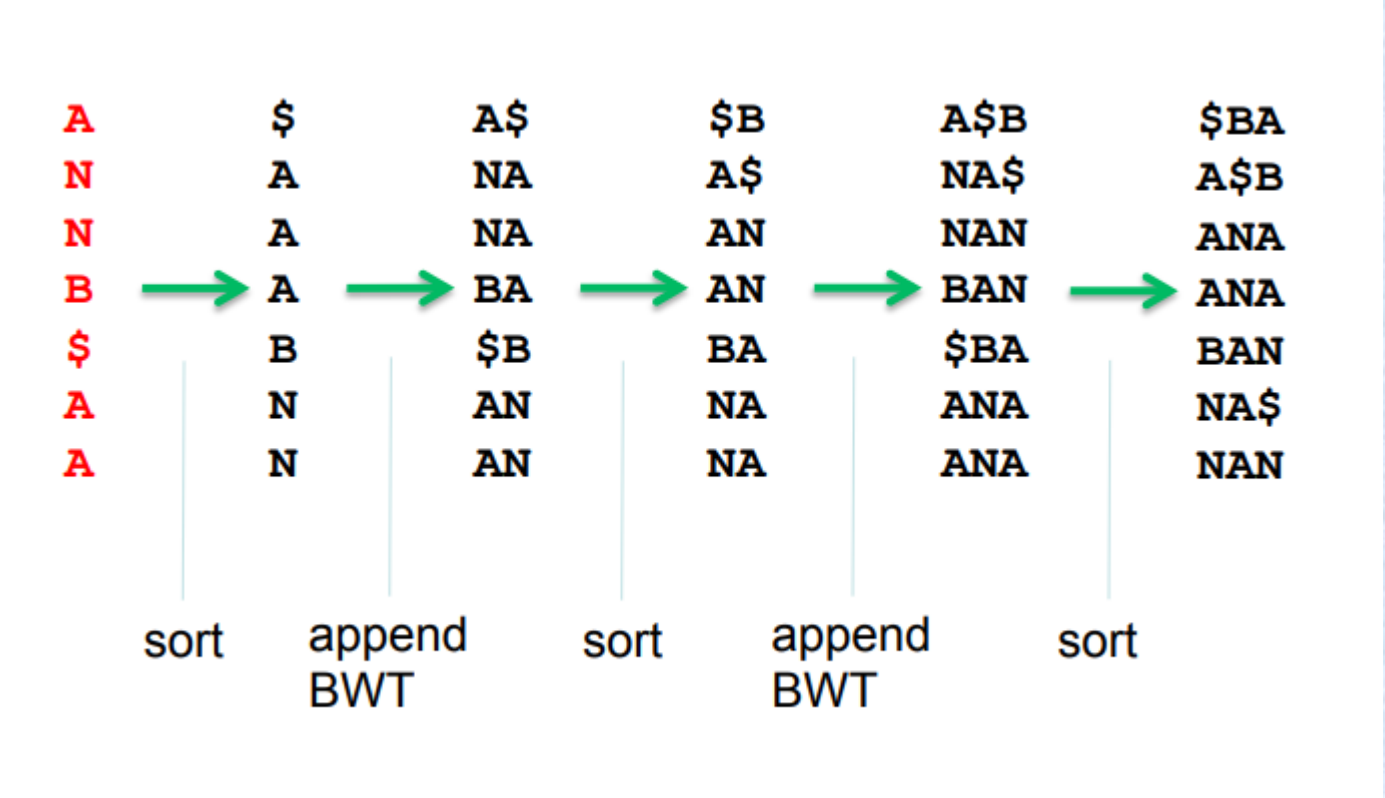
The transform is done by sorting all rotations of the text, then taking the last column. For example, the text "BANANA$" is transformed into "ANNB$AA" through these steps (the red $ character indicates the 'EOF' pointer):

| Transformation | | | |
|---|---|---|---|
| Input | All Rotations | Sort the Rows | Output |
| BANANA$ | BANANA$<br>ANANA$B<br>NANA$BA<br>ANA$BAN<br>NA$BANA<br>A$BANAN<br>$BANANA | $BANANA<br>A$BANAN<br>ANA$BAN<br>ANANA$B<br>BANANA$<br>NA$BANA<br>NANA$BA | ANNB$AA |

BWT not only generates a more easily encoded string
It is reversible, allowing the original data to be re-generated

Modified from http://en.wikipedia.org/wiki/Burrows–Wheeler_transform

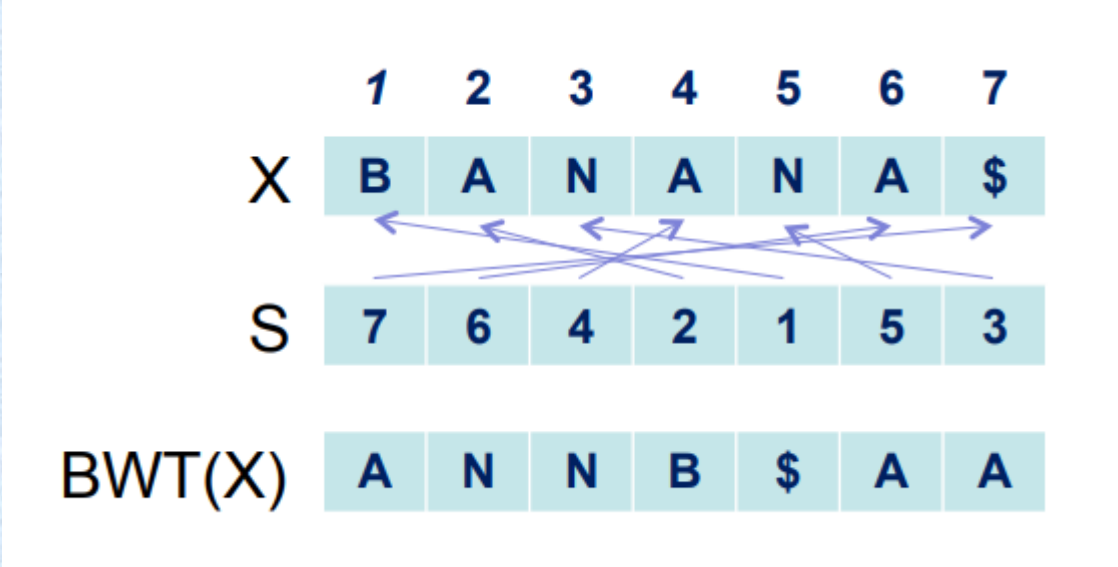
# Getting back the original string




- The last column sorted by characters return the first column
- The first and last columns together give you all *pairs* of successive characters , where pairs are taken cyclically so that the last and first character form a pair.
- Sorting the list of pairs gives the first and second columns.

https://web.stanford.edu/class/cs262/presentations/lecture4.pdf

# Keeping the original string location




http://blog.thegrandlocus.com/2016/07/a-tutorial-on-burrows-wheeler-indexing-methods

# Bowtie algorithm

More details in https://www.youtube.com/watch?v=z5EDLODQPtg

**The places to look for the read**

```
@HWI-EAS412_4:1:1:1376:380
AATGATACGGCGACCACCGAGATCTA
```

http://mged12-deep-sequencing-analysis.wikispaces.com/file/view/Cole_MGED_tutorial_slides.pdf

# Bowtie algorithm

**The places to look for the read**

```
@HWI-EAS412_4:1:1:1376:380
AATGATACGGCGACCACCGAGATCTA
```

http://mged12-deep-sequencing-analysis.wikispaces.com/file/view/Cole_MGED_tutorial_slides.pdf

# Bowtie algorithm

**The places to look for the read**

```
@HWI-EAS412_4:1:1:1376:380
AATGATACGGCGACCACCGAGATCTA
```

http://mged12-deep-sequencing-analysis.wikispaces.com/file/view/Cole_MGED_tutorial_slides.pdf

# Bowtie algorithm

**The places to look for the read**

```
@HWI-EAS412_4:1:1:1376:380
AATGATACGGCGACCACCGAGATCTA
```

http://mged12-deep-sequencing-analysis.wikispaces.com/file/view/Cole_MGED_tutorial_slides.pdf

# Bowtie algorithm

**The places to look for the read**

```
@HWI-EAS412_4:1:1:1376:380
AATGATACGGCGACCACCGAGATCTA
```

http://mged12-deep-sequencing-analysis.wikispaces.com/file/view/Cole_MGED_tutorial_slides.pdf

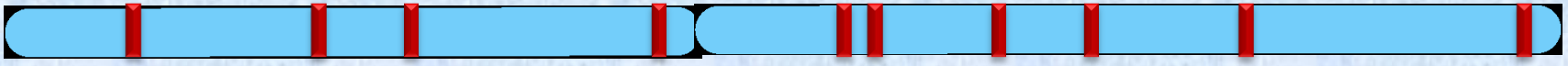
# Bowtie algorithm

**The places to look for the read**

```
@HWI-EAS412_4:1:1:1376:380
AATGATACGGCGACCACCGAGATCTA
```

http://mged12-deep-sequencing-analysis.wikispaces.com/file/view/Cole_MGED_tutorial_slides.pdf

# Bowtie algorithm

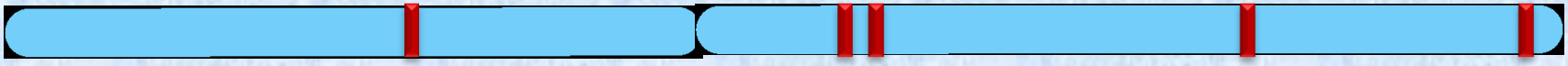


**The places to look for the read**

```
@HWI-EAS412_4:1:1:1376:380
AATGATACGGCGACCACCGAGATCTA
```

http://mged12-deep-sequencing-analysis.wikispaces.com/file/view/Cole_MGED_tutorial_slides.pdf

# Bowtie algorithm

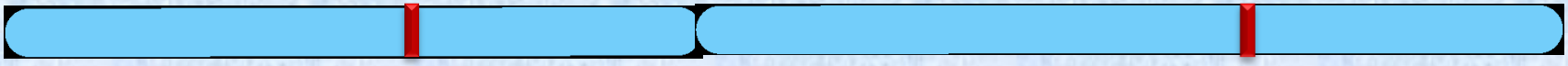


**The places to look for the read**

```
@HWI-EAS412_4:1:1:1376:380
AATGATACGGCGACCACCGAGATCTA
```

http://mged12-deep-sequencing-analysis.wikispaces.com/file/view/Cole_MGED_tutorial_slides.pdf

# Bowtie algorithm



**The places to look for the read**

```
@HWI-EAS412_4:1:1:1376:380
AATGATACGGCGACCACCGAGATCTA
```

http://mged12-deep-sequencing-analysis.wikispaces.com/file/view/Cole_MGED_tutorial_slides.pdf

# Bowtie algorithm

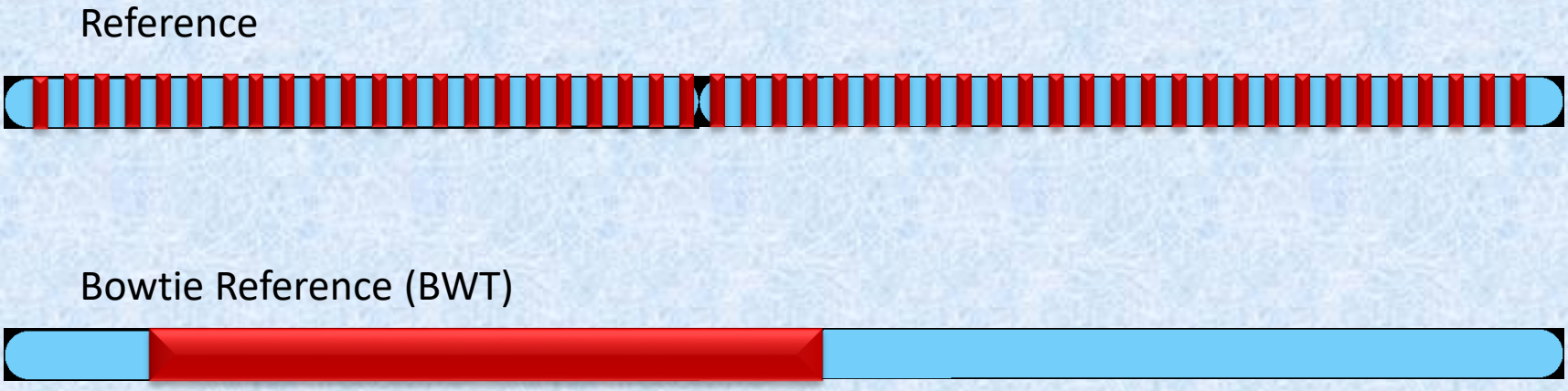
Reference

Bowtie Reference (BWT)

```
@HWI-EAS412_4:1:1:1376:380
AATGATACGGCGACCACCGAGATCTA
```

http://mged12-deep-sequencing-analysis.wikispaces.com/file/view/Cole_MGED_tutorial_slides.pdf

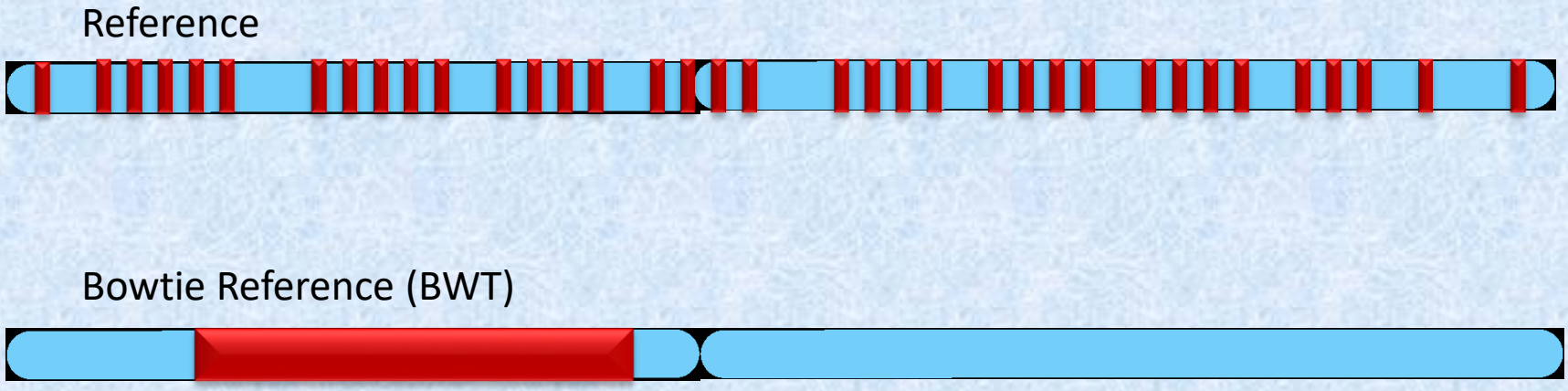
# Bowtie algorithm




```
@HWI-EAS412_4:1:1:1376:380
AATGATACGGCGACCACCGAGATCTA
```

http://mged12-deep-sequencing-analysis.wikispaces.com/file/view/Cole_MGED_tutorial_slides.pdf

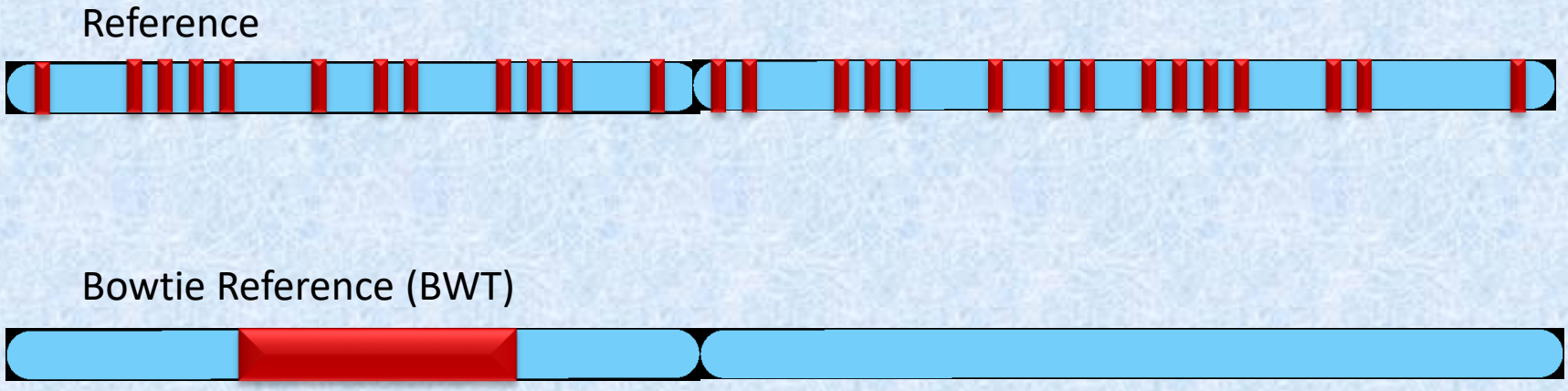
# Bowtie algorithm




```
@HWI-EAS412_4:1:1:1376:380
AATGATACGGCGACCACCGAGATCTA
```

http://mged12-deep-sequencing-analysis.wikispaces.com/file/view/Cole_MGED_tutorial_slides.pdf

# Bowtie algorithm

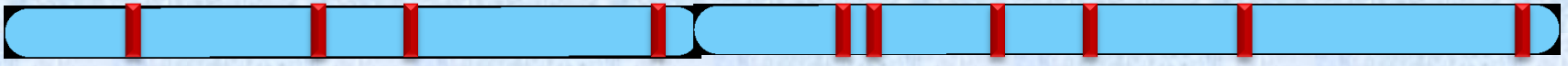



```
@HWI-EAS412_4:1:1:1376:380
AATGATACGGCGACCACCGAGATCTA
```

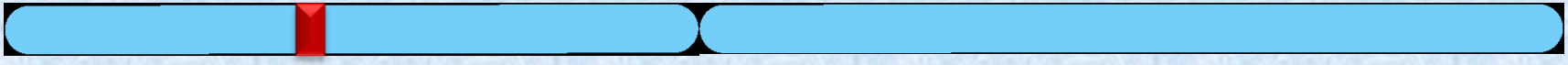
http://mged12-deep-sequencing-analysis.wikispaces.com/file/view/Cole_MGED_tutorial_slides.pdf

# Bowtie algorithm

Reference



Bowtie Reference (BWT)

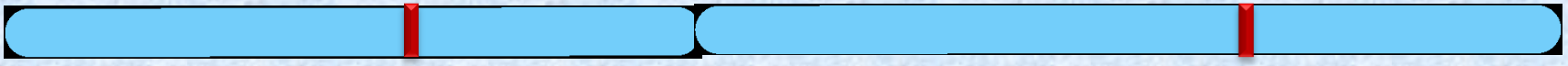


```
@HWI-EAS412_4:1:1:1376:380
AATGATACGGCGACCACCGAGATCTA
```

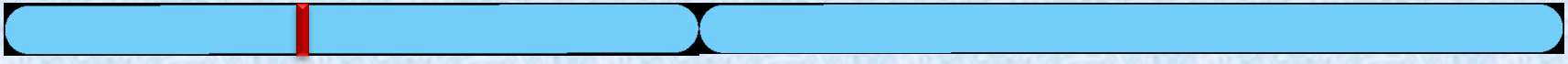
http://mged12-deep-sequencing-analysis.wikispaces.com/file/view/Cole_MGED_tutorial_slides.pdf

# Bowtie algorithm

Reference



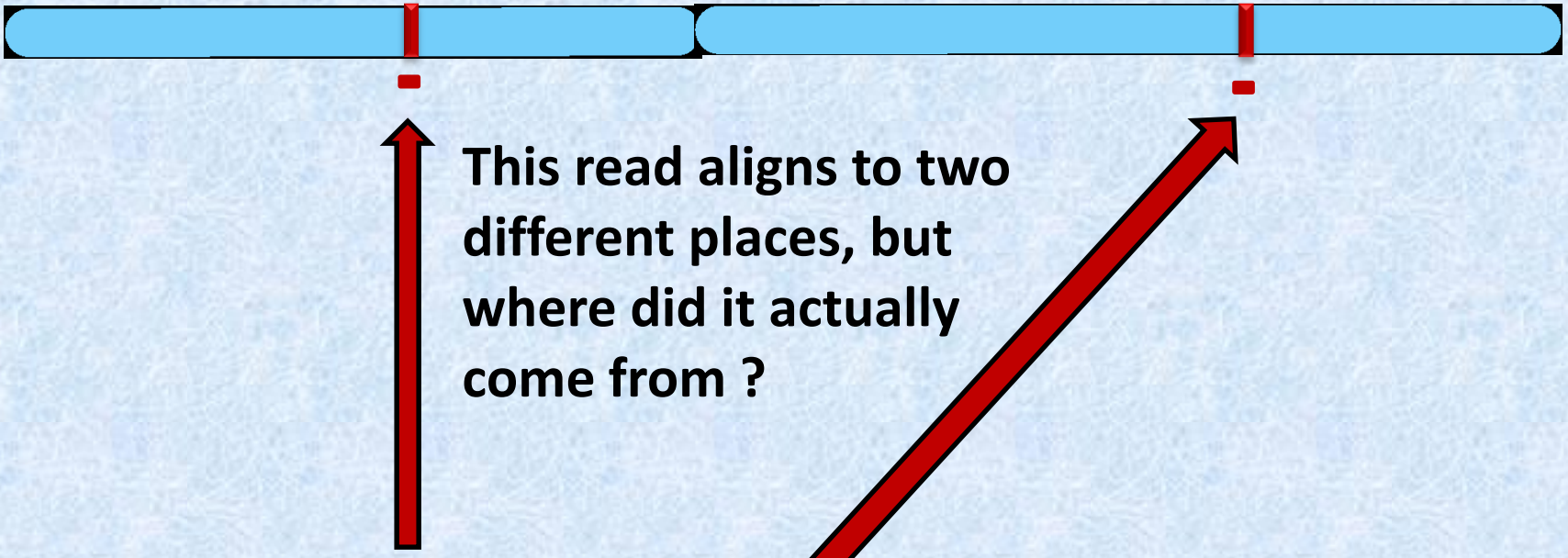
Bowtie Reference (BWT)



```
@HWI-EAS412_4:1:1:1376:380
AATGATACGGCGACCACCGAGATCTA
```

http://mged12-deep-sequencing-analysis.wikispaces.com/file/view/Cole_MGED_tutorial_slides.pdf

# Picking the right alignment

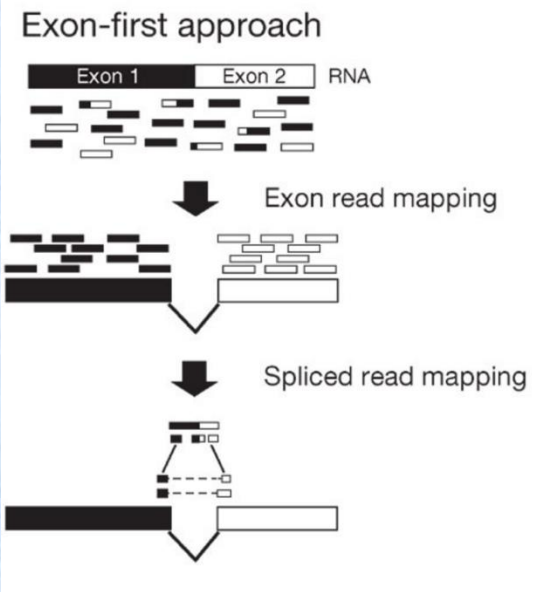


**This read aligns to two different places, but where did it actually come from ?**

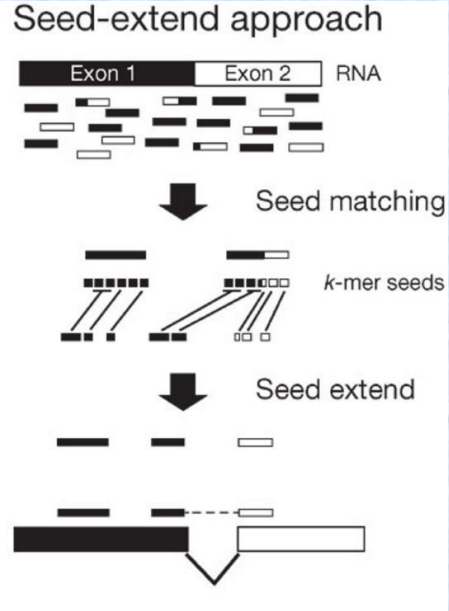
```
@HWI-EAS412_4:1:1:1376:380
AATGATACGGCGACCACCGAGATCTA
```

http://mged12-deep-sequencing-analysis.wikispaces.com/file/view/Cole_MGED_tutorial_slides.pdf

# Transcriptome alignment




- Map full, unspliced reads (exonic reads).
- Remaining reads are divided into smaller pieces and mapped to the genome.
- An extension process extends mapped pieces to find candidate splice sites to support a spliced alignment.




- Store a map of all small words (*k*-mers) in an efficient lookup data structure.
- Each read is divided into *k*-mers, which are mapped to the genome via the lookup structure.
- Mapped *k*-mers are extended into larger alignments, which may include gaps flanked by splice sites.

http://www.nature.com/nmeth/journal/v8/n6/fig_tab/nmeth.1613_F1.html

# Popular aligners

| For DNA | For RNA |
|---|---|
| Bowtie2 | TopHat2 |
| BWA | Star |
| BWA-MEM | hisat2 |

Why do we need different aligners for DNA and RNA?

**We need to allow for gaps because of the exon – intron gene structure**

Would you prefer to have local or global alignments?

**Depending on the application and read length**

# Popular aligners

| For DNA | For RNA |
|---|---|
| Bowtie2 | TopHat2 |
| BWA | Star |
| BWA-MEM | hisat2 |

**DNA Aligners**

- They can use global or local alignments
- BWA-MEM uses both global and local alignments
- Bowtie2 and both BWA allow indels
- BWA-MEM:
  - Fitted for aligning sequence reads or long query sequences against a large reference genome
  - It automatically chooses between local and end-to-end alignments, supports paired-end reads and performs chimeric alignment
  - It is permissive to long gaps up to tens of bp for 100bp reads, or up to several hundred bp (tunable) for contig alignment

# Popular aligners

| For DNA | For RNA |
|---|---|
| Bowtie2 | TopHat2 |
| BWA | Star |
| BWA-MEM | Hisat2 |

Exon-first approach

Seed searching step (Maximum Mappable Prefix) and clustering/stitching/scoring step.

Double index and combined approach of anchor and extension

## RNA Aligners

- **Star features**
  - Outperformed other aligners by a factor of >50 in mapping speed in 2012
  - Detects canonical junctions *de novo*
  - Can discover non-canonical splices and chimeric (fusion) transcripts
  - Can deal with long reads (up to full transcripts)
- **Hisat2 features**
  - The fastest today
  - The lowest memory requirement
  - High sensitivity and accuracy of splice site detection

# SAM format

HWI-ST808:87:C068VACXX:2:1101:1234:2199 16 chr5 177482768 2 100M * 0 0
TGACGGTCCATTCCCGGGCTCGATGCCGGAAAAACCCCTTGGCCCGCCGGAAGGGCAGGCACATGGGCATAGGTAAGCGGAAGGGTACAGCCAATGCACG
######@CA?5&DBB@@9BA99<7@98:(?@?5)5(@?<807DCBHFHGBIIIHHHEF@@FB?3GIIIIGGGGEIGFIIIHEFBIGFFFBHFEDDAA:=:
AS:i:-29 XS:i:-32 XN:i:0 XM:i:6 XO:i:0 XG:i:0 NM:i:6 MD:Z:35A26G3C7G3A18C2 YT:Z:UU

Alignments are reported in a compact representation: SAM format

```
0     61G9EAAXX100520:5:100:10095:16477  (read name)
1     83  (FLAGS stored as bit fields; 83 = 00001010011 )
2     chr1  (alignment target)
3     51986 (position alignment starts)
4     38
5     46M  (Compact description of the alignment in CIGAR format)
6     =
7     51789
8     -264
9     CCCAAACAAGCCGAACTAGCTGATTTGGCTCGTAAAGACCCGGAAA
10    ###CB?=ADDBCBCDEEFFDEFFFDEFFGDBEFGEDGCFGFGGGGG
11    MD:Z:67
12    NH:i:1
13    HI:i:1
14    NM:i:0
15    SM:i:38
16    XQ:i:40
17    X2:i:0
```

(read sequence, oriented according to the forward alignment)

(base quality values)

(Metadata)

SAM format specification: http://samtools.sourceforge.net/SAM1.pdf

# QUESTIONS?

# Thank you