The Illumina/Solexa platform (2/2)

- Some numbers:
  - Reads length: 35-100bp
  - 3G clusters per flow cell -> 600 Gb/run (HiSeq2000)
  - Time per run: 2 to 11 days
  - Platform specific sequencing error:
    - base miscalling ~1%

NGS output

- Millions of sequences -> size of file up toGBytes
- Output formats:
  - **FASTA**
  - **FASTQ** ≈ FASTA + quality score
    - Quality score = goodness of base call (higer is better)

FASTQ file example:

```
@SRR006511.105 8_1_663_27 length=36
ATAGCGGCACTGTTGGTTCGCTTGTTCCTTTGAGTC
+
IIII7II-9/0;+8I<03.+%-,&"+'($,#""'&"
@SRR006511.112 8_1_829_108 length=36
AGAATTTTATGTATCTGGATGCAATAAAAAATGATG
+
II@IIIIIIIIIIIII1DII>0<I?>869;64(+%
@SRR006511.490 8_1_351_672 length=36
AGCACCCGCCGTGTGTCCCCCATGCTCCACACCTCT
+
I0>0A,I2H):$)6)#4$.)>'.&.$)"%7"1%)&&
@SRR006511.632 8_1_79_187 length=36
ATGCCGAAAGGTATCGGTAAACCGTTGAAATTCTTC
+
IIIIII<I;II57G;II.I0**32.--)$32++9),
@SRR006511.726 8_1_300_437 length=36
ACCACGTGGACTTCCAGGACCATGAGGCCAAATTGG
+
I1B>:IIII)3,I&0-,;$(%&%1$+1"&($%"&#"
```

Read 1

Read 2

Read 3

Read 4

…

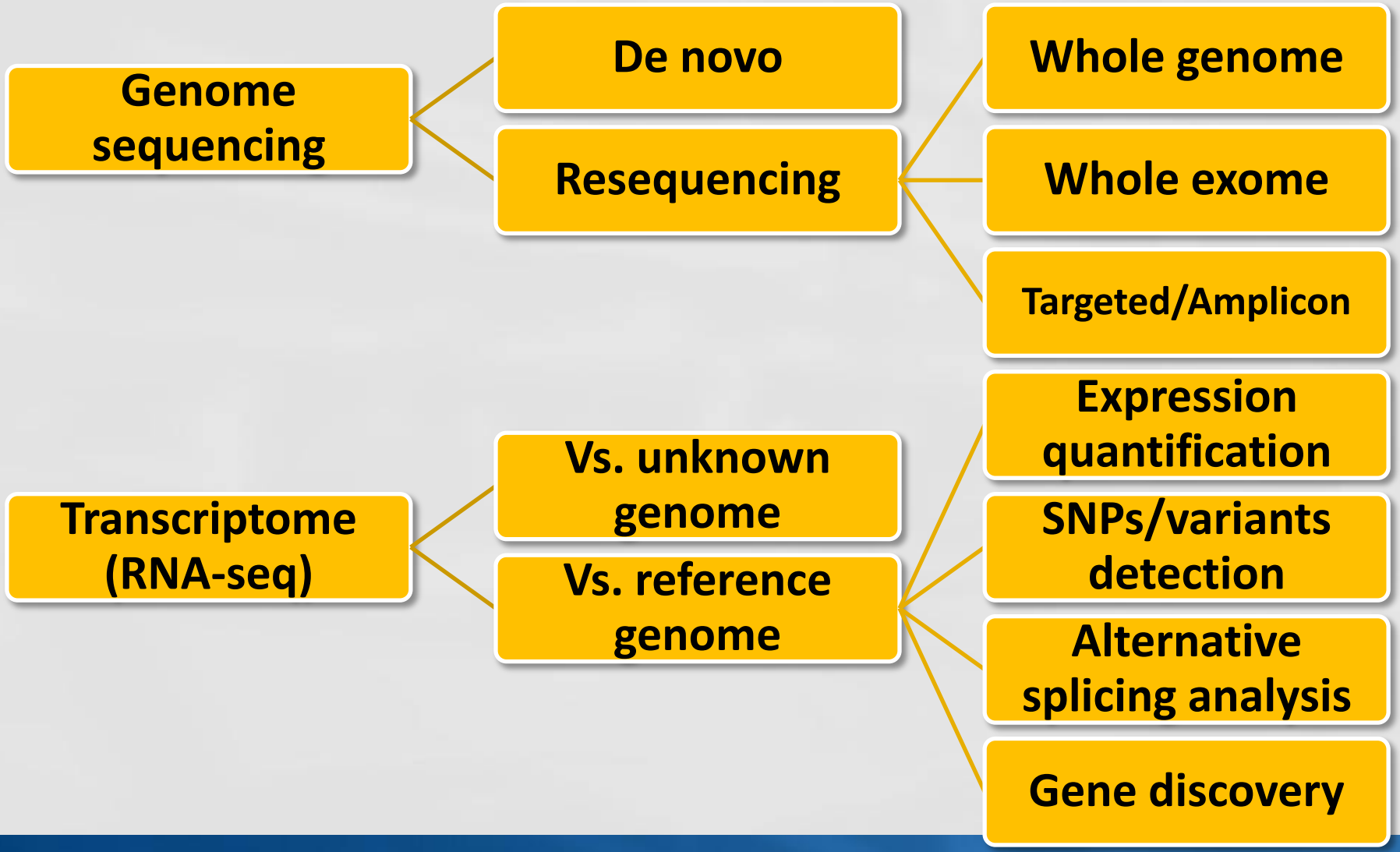
1) '@' sequence identifier and description
2) Raw sequence (in letters)
3) '+' sequence identifier (again) – opt.
4) Quality score per nucleot., char encoded

Phred Quality Score

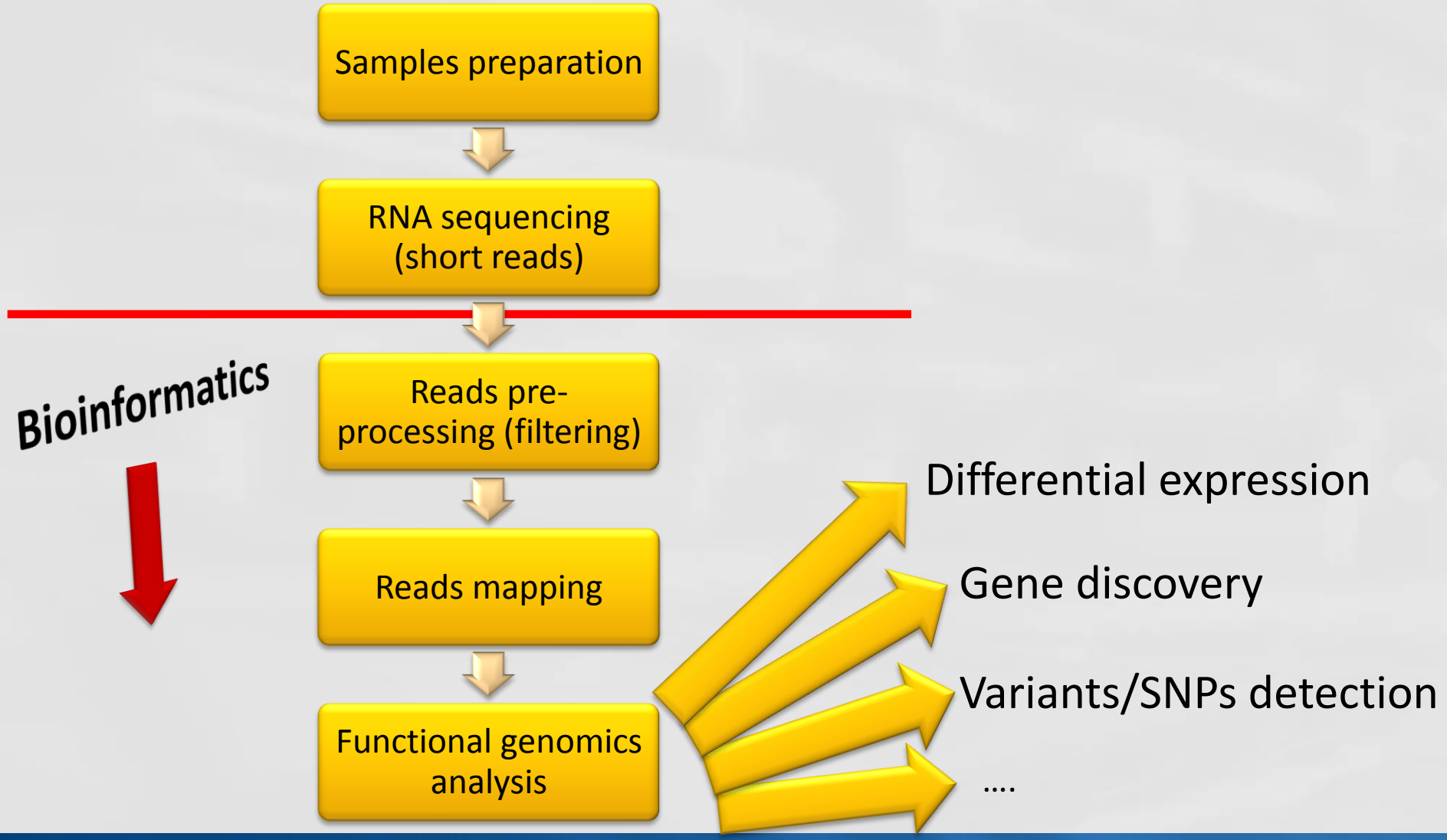
$$Q = -10 \log_{10} P \qquad P = 10^{\frac{-Q}{10}}$$

Phred Quality Score	Prob. Of Incorrect Base Call	Base Call Accuracy
10	1 in 10	90%
20	1 in 100	99%
30	1 in 1000	99.9

NGS applications

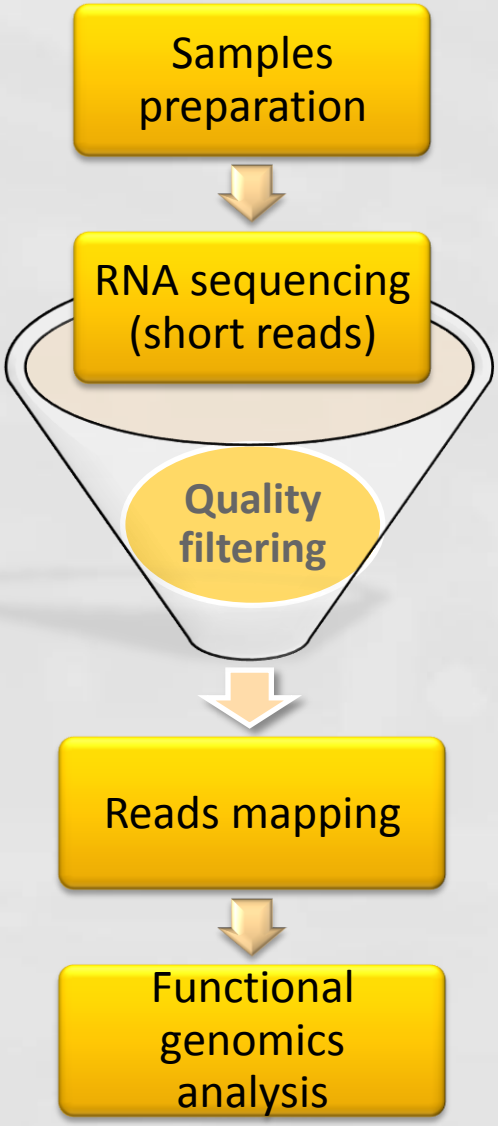


RNA-seq data processing: main steps



Reads pre-processing

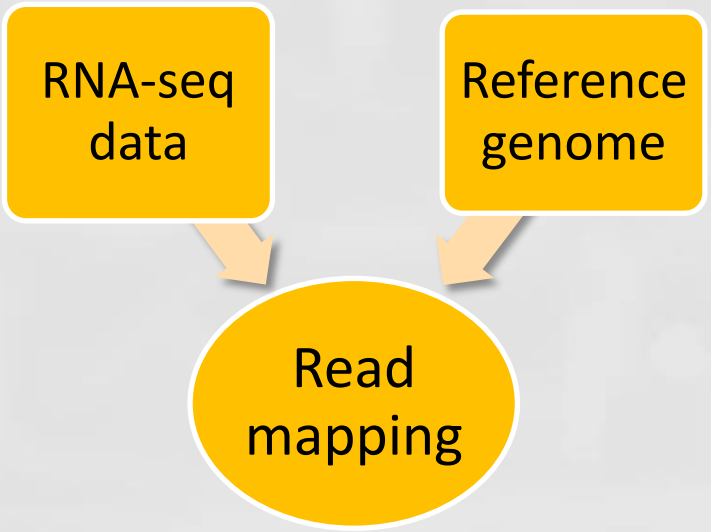
- Quality is better than quantity
- Discard possible noise sources:
  - Adapter removal
  - Genetic contaminants removal
  - Reads quality filtering by checking:
    - Overall quality score of read
    - Quality of the sequence (‘N’s removal)



Reads mapping

- Mapping = arranging and aligning the reads to a reference genome
- We map to identify the expressed genomic regions
- Problems:
  - Reference transcriptome/genome must be available
  - Splicing/alternative splicing events within reads
  - High computational complexity:
    - Lots of reads
    - Small length of reads
    - Large genomic regions

Input requirements

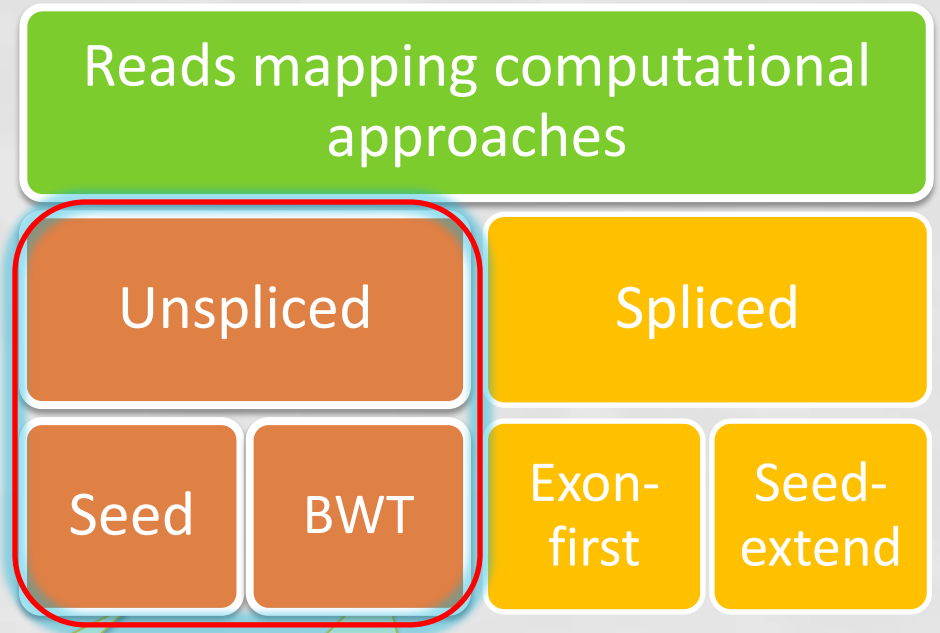


Reads mapping computational approaches

(*) M.Garber, M. G. Grabherr, M. Guttman& C. Trapnell. *Computational methods for transcriptome annotation and quantification usingRNA-seq*, Nature Methods 8, 469–477 (2011)

Unspliced aligners

- No large gap alignment (reads are not “spliced”)
- Look for perfect matches
- Align to reference transcriptome
- Ideal for quantification pourposes
- Identify only known exons

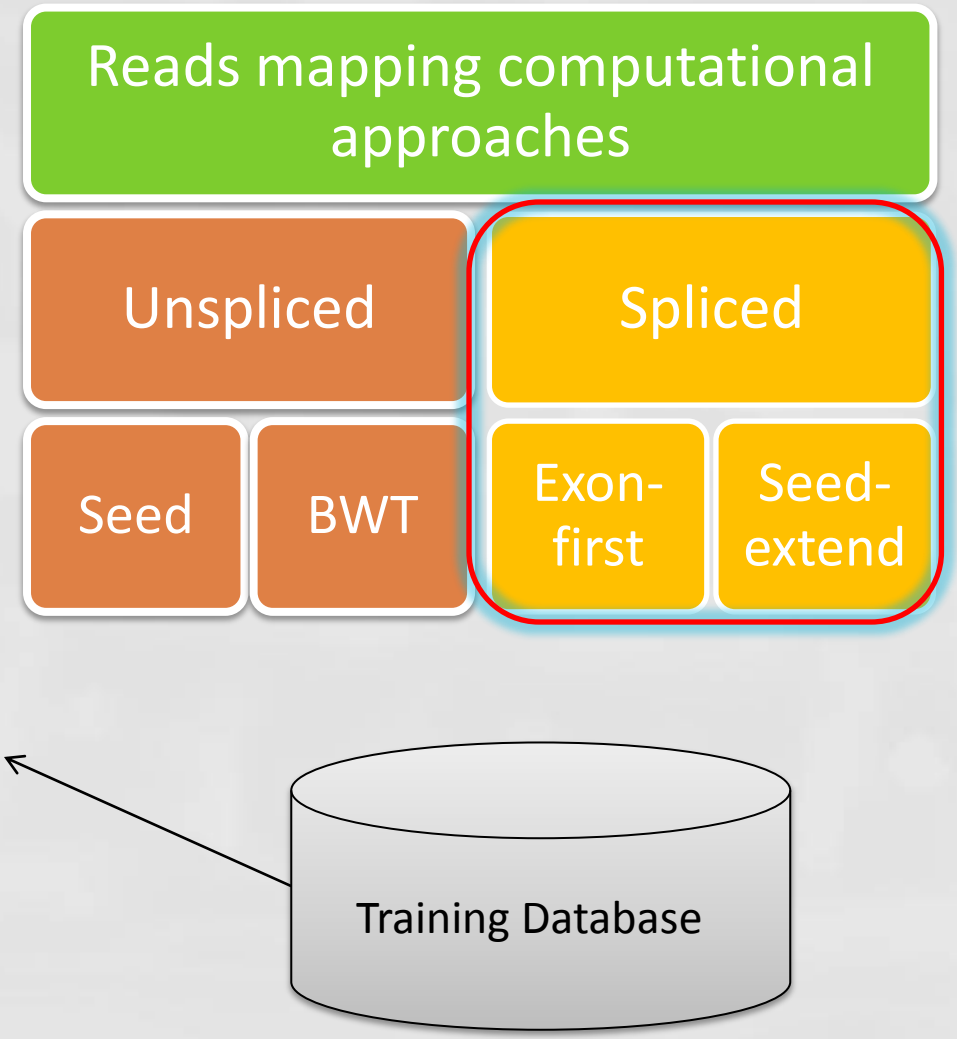


Perfect match for a ‘seed’ ->try to extend the alignment (Smith-Waterman)
•SHRiMP, Stampy, MAQ

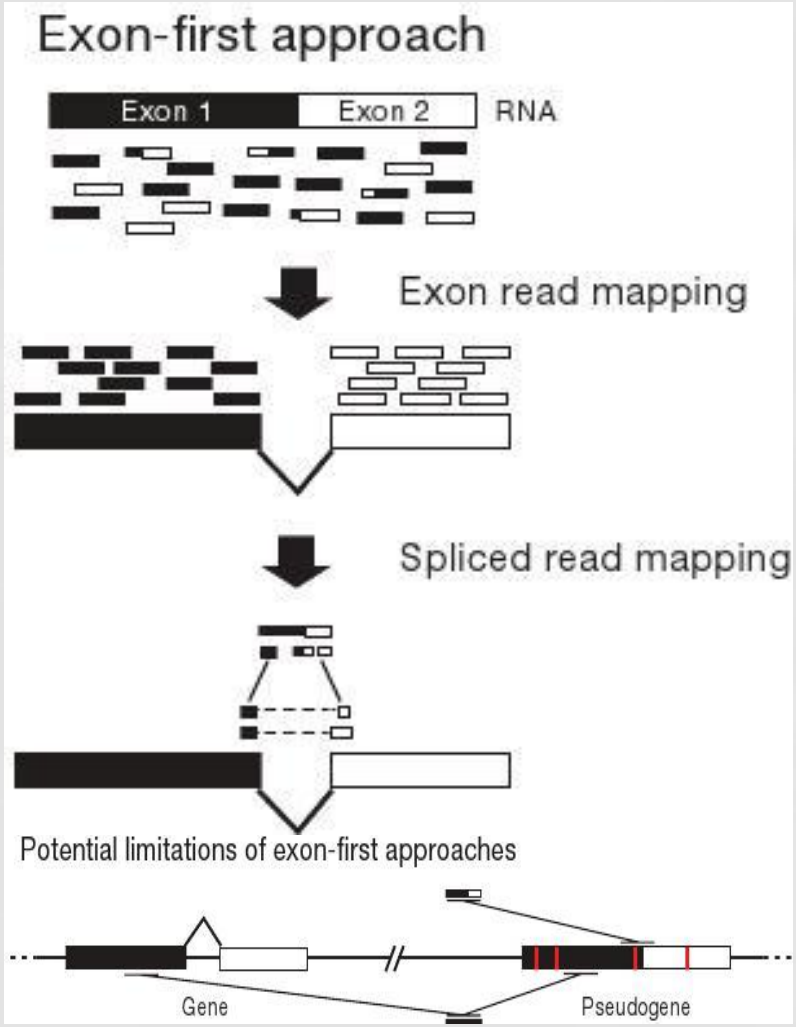
Genome compression by Burrows-Wheeler Transform ->very efficent structure for finding perfect matches
•**Bowtie**, BWA

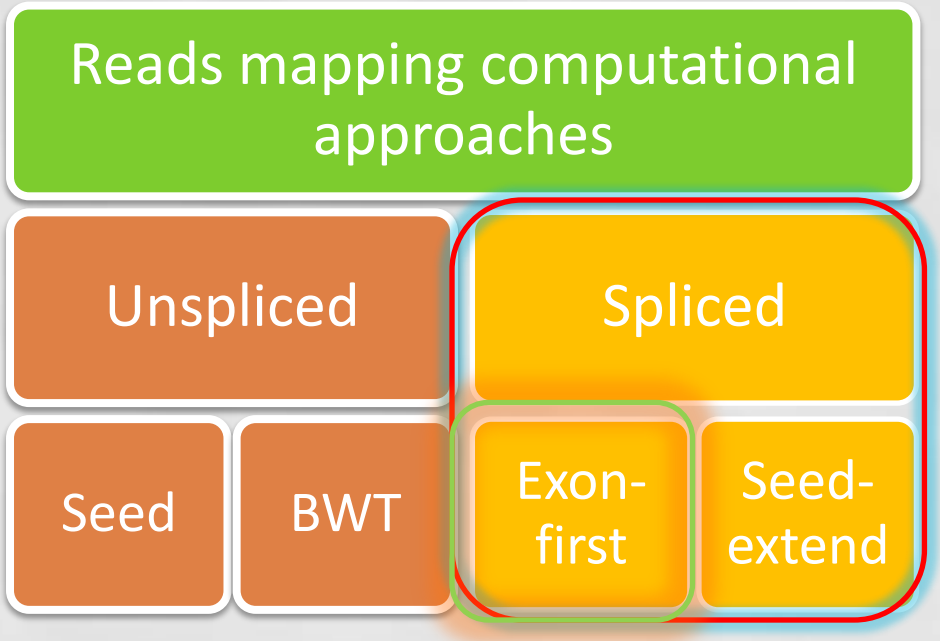
Spliced aligners

- Allow large gap alignments
- Reads can be fragmented and aligned even when spanned by introns
- New splicing sites (exons) prediction (ex: by machine learning or probabilistic methods)



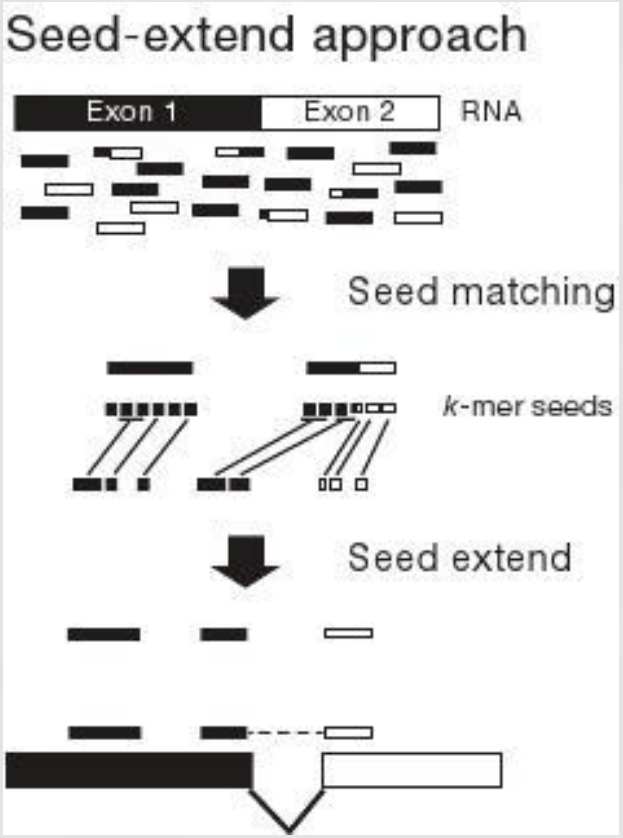
Exon-first

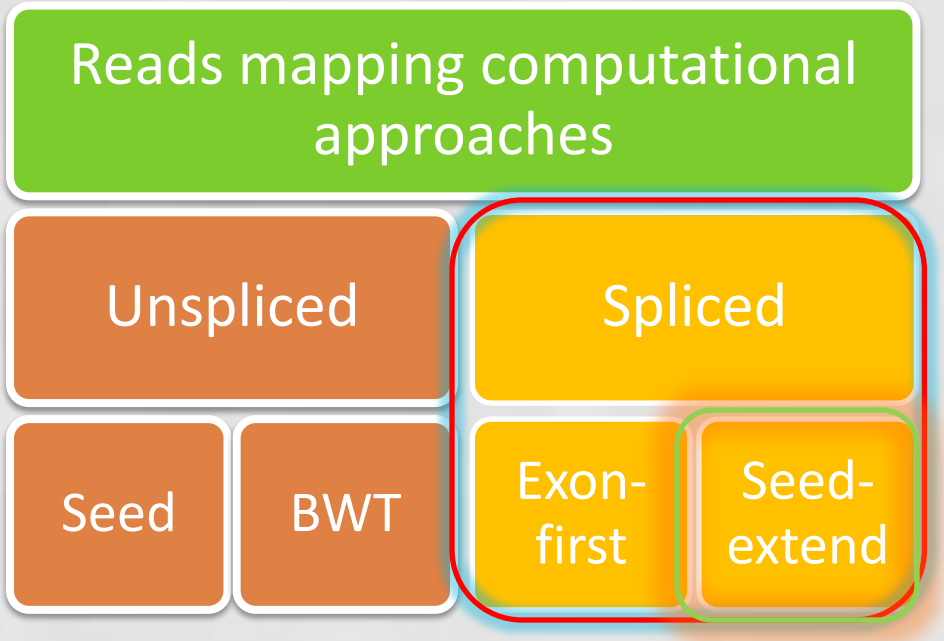




- Two steps process:
  1. Align with unspliced aligner
  2. Fragment remaining reads and map the fragments
- Caveat: retrotransposed pseudogenes
- Ex: **TopHat**, SpliceMap, MapSplice

Seed-extend





- Single step:
  - Fragment each read and map the fragments
- Computationally more expensive
  - Need parallelized computing
- Not influenced by retrotransposed pseudogenes
- Ex: **GSNAP**, QPALMA

SAM Format

```
Coor     12345678901234  5678901234567890123456789012345
ref      AGCATGTTAGATAA**GATAGCTGTGCTAGTAGGCAGTCAGCGCCAT

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

The corresponding SAM format is:

```
@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         *
```