An Introduction to RNA-seq Analysis I

Gunnar Rätsch

Computational Biology Center
Memorial Sloan-Kettering Cancer Center



Memorial Sloan-Kettering
Cancer Center

cBio@MSKCC

Overview

Part I

- RNAs, RNA-seq & research questions
- Common analysis steps
- Read mapping
  - The problem
  - String data structures
  - Programs
  - Extensions
- Differences between read mappers

Part II

- Transcript quantification & biases
- Differential "expression"
- Transcript reconstruction by ...
  - alignment
  - assembly
- Genome annotation

Transcription & RNA Processing in Eukaryotes

- Newly synthesized pre-mRNA is capped.

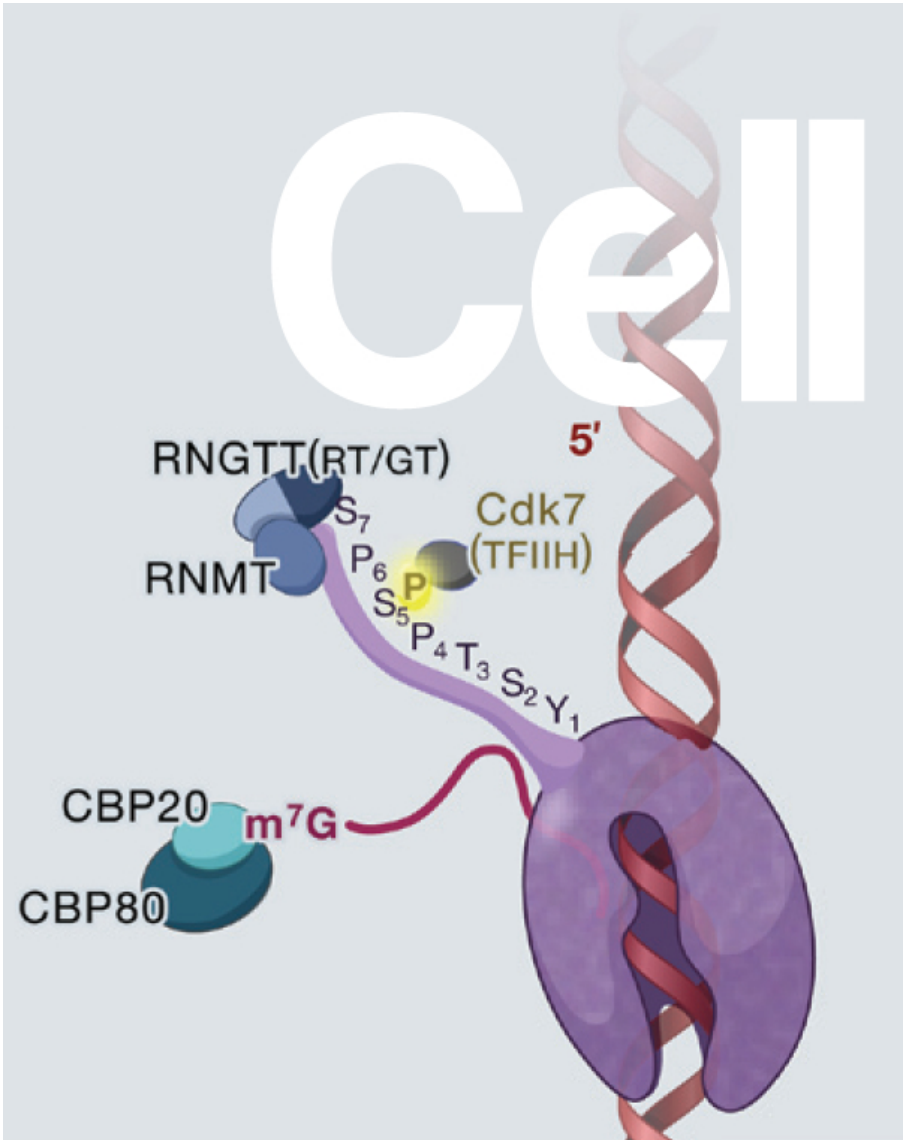
  [CBP20 & CBP80: cap-binding proteins]

- Introns are spliced from pre-mRNA.

  [U1, U2, U4-6: spliceosome

  SF1, U2AF, SR proteins: splicing factors]

- A polyA-tail is added to the 3' terminus of pre-mRNA.



[Bergkessel et al., 2009]

Transcription & RNA Processing in Eukaryotes

- Newly synthesized pre-mRNA is capped.

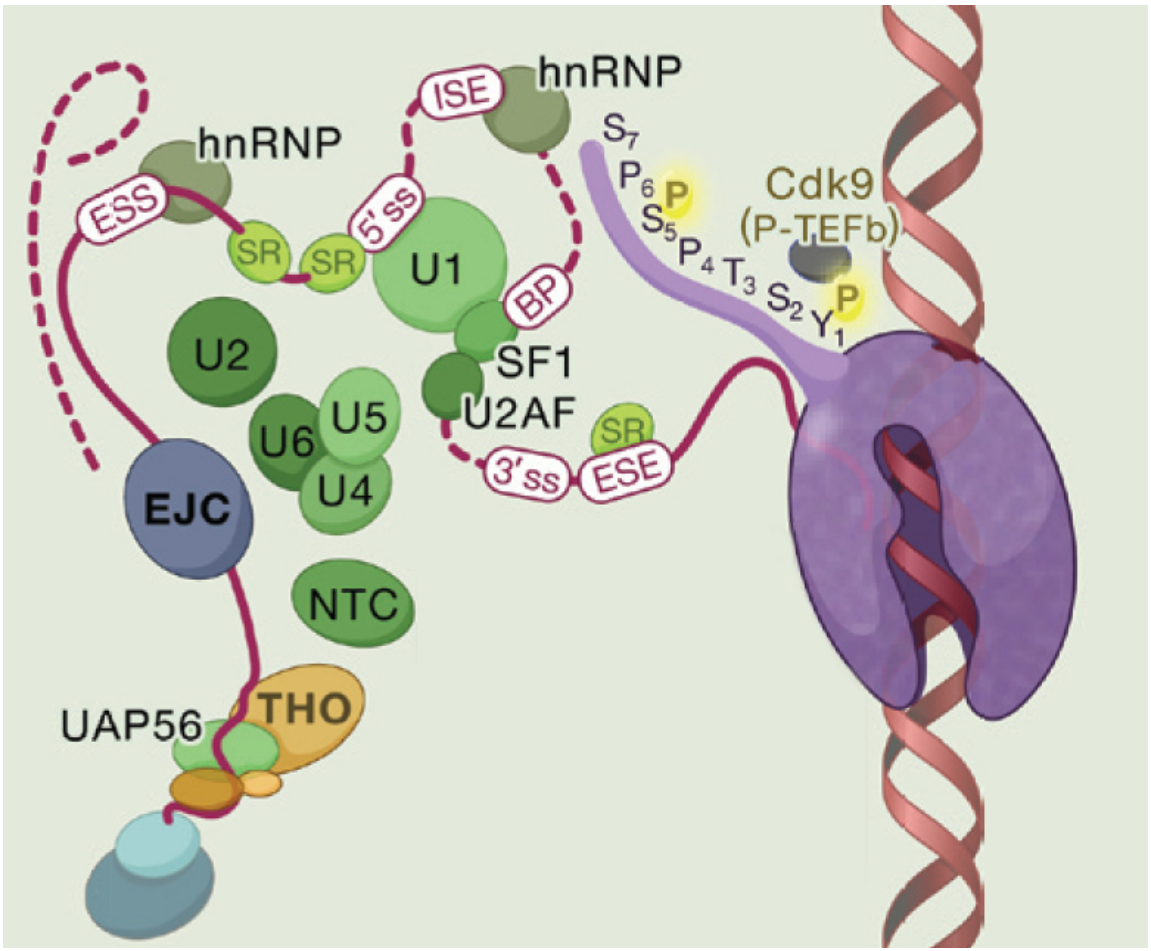
  [CBP20 & CBP80: cap-binding proteins]

- Introns are spliced from pre-mRNA.

  [U1, U2, U4-6: spliceosome

  SF1, U2AF, SR proteins: splicing factors]

- A polyA-tail is added to the 3' terminus of pre-mRNA.



[Bergkessel et al., 2009]

Transcription & RNA Processing in Eukaryotes

- Newly synthesized pre-mRNA is capped.

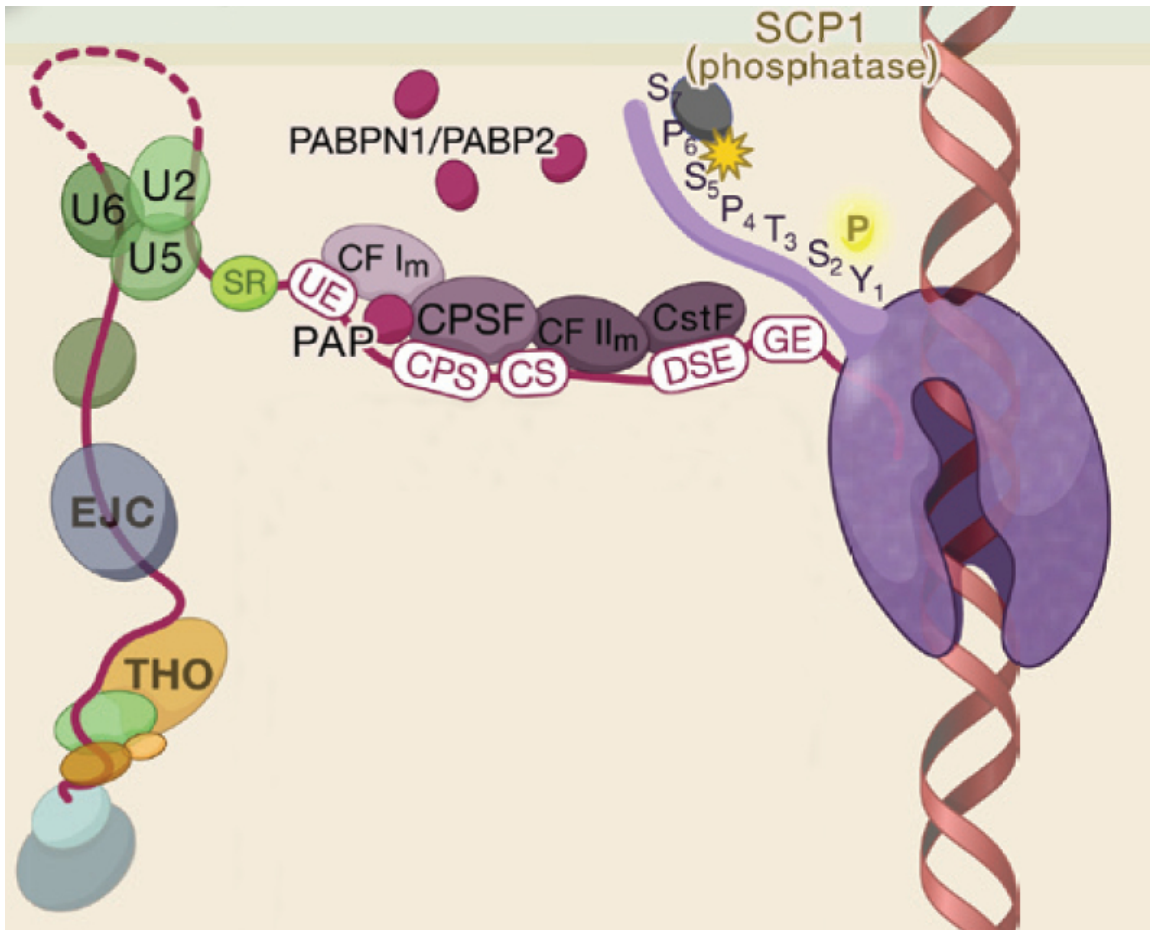
  [CBP20 & CBP80: cap-binding proteins]

- Introns are spliced from pre-mRNA.

  [U1, U2, U4-6: spliceosome

  SF1, U2AF, SR proteins: splicing factors]

- A polyA-tail is added to the 3' terminus of pre-mRNA.



[Bergkessel et al., 2009]

Key Research Questions

- Characterize an organism's full complement of genes
  - ⇒ Find new, possibly noncoding genes
  - ⇒ Compare genes among organisms, strains, individuals, tissues
  - ⇒ Identify alternative splice forms or transcript ends
- Understand transcriptome regulation
  - ⇒ Monitor transcriptome changes between tissues or in response to environmental changes (e.g. stress)
  - ⇒ Knock-out / knock-down analysis of regulators
    - ⇝ Identify regulated targets with significant expression changes
  - ⇒ Identify binding sites used in regulation (e.g. ChIP-seq)
- Understand differences between populations, individuals, cells
  - ⇒ QTL-Mapping of transcriptome-phenotypes
  - ⇒ Effect of rare or somatic mutations

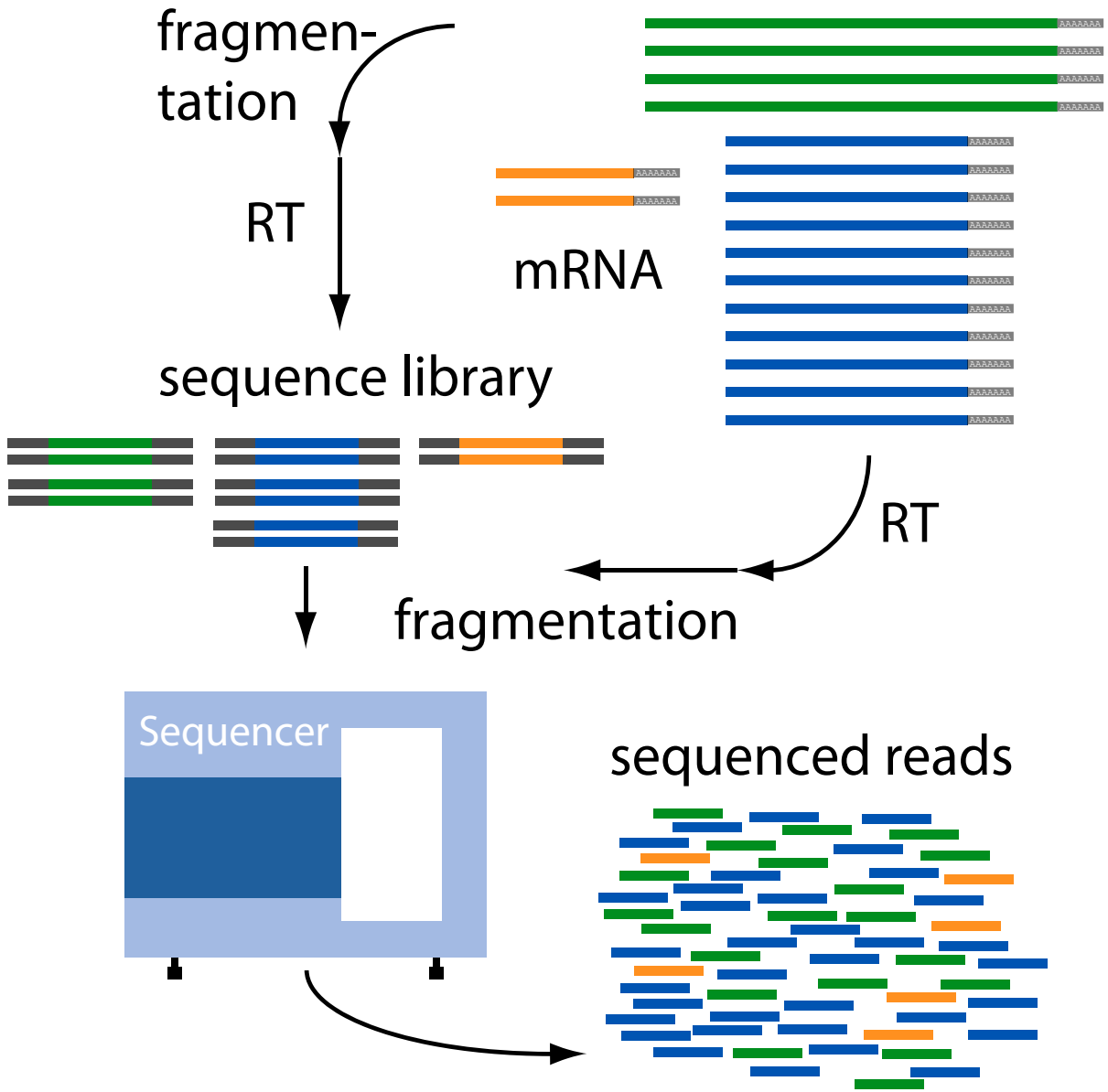
Key Research Questions

- Characterize an organism's full complement of genes
  - ⇒ Find new, possibly noncoding genes
  - ⇒ Compare genes among organisms, strains, individuals, tissues
  - ⇒ Identify alternative splice forms or transcript ends
- Understand transcriptome regulation
  - ⇒ Monitor transcriptome changes between tissues or in response to environmental changes (e.g. stress)
  - ⇒ Knock-out / knock-down analysis of regulators
    - ⇝ Identify regulated targets with significant expression changes
  - ⇒ Identify binding sites used in regulation (e.g. ChIP-seq)
- Understand differences between populations, individuals, cells
  - ⇒ QTL-Mapping of transcriptome-phenotypes
  - ⇒ Effect of rare or somatic mutations

Key Research Questions

- Characterize an organism's full complement of genes
  - ⇒ Find new, possibly noncoding genes
  - ⇒ Compare genes among organisms, strains, individuals, tissues
  - ⇒ Identify alternative splice forms or transcript ends
- Understand transcriptome regulation
  - ⇒ Monitor transcriptome changes between tissues or in response to environmental changes (e.g. stress)
  - ⇒ Knock-out / knock-down analysis of regulators
    - ⇝ Identify regulated targets with significant expression changes
  - ⇒ Identify binding sites used in regulation (e.g. ChIP-seq)
- Understand differences between populations, individuals, cells
  - ⇒ QTL-Mapping of transcriptome-phenotypes
  - ⇒ Effect of rare or somatic mutations

RNA-Seq Basics



Biological Question ⇒ Analysis

- New genome & variation — qualitative
  - Sequence DNA ⇒ assemble [⇒ align ⇒ detect]
  - Sequence ⇒ align to DNA ⇒ detect
- New gene/transcript — qualitative
  - Sequence RNA ⇒ assemble [⇒ align ⇒ detect]
  - Sequence RNA ⇒ align to DNA ⇒ detect
- Gene/transcript expression — quantitative
  - Sequence RNA ⇒ align to known RNAs ⇒ count
  - Sequence RNA ⇒ align to DNA ⇒ count
- DNA/RNA-Protein binding — quant- & qualitative
  - Binding assay ⇒ Sequence ⇒ align ⇒ identify peaks
- Methylation — quant- & qualitative
  - Bisulfite assay ⇒ Sequence DNA/RNA ⇒ align⇒ count
- ...

Biological Question ⇒ Analysis

- New genome & variation — qualitative
  - Sequence DNA ⇒ assemble [⇒ align ⇒ detect]
  - Sequence ⇒ align to DNA ⇒ detect
- New gene/transcript — qualitative
  - Sequence RNA ⇒ assemble [⇒ align ⇒ detect]
  - Sequence RNA ⇒ align to DNA ⇒ detect
- Gene/transcript expression — quantitative
  - Sequence RNA ⇒ align to known RNAs ⇒ count
  - Sequence RNA ⇒ align to DNA ⇒ count
- DNA/RNA-Protein binding — quant- & qualitative
  - Binding assay ⇒ Sequence ⇒ align ⇒ identify peaks
- Methylation — quant- & qualitative
  - Bisulfite assay ⇒ Sequence DNA/RNA ⇒ align⇒ count
- ...

Biological Question ⇒ Analysis

- New genome & variation — qualitative
  - Sequence DNA ⇒ assemble [⇒ align ⇒ detect]
  - Sequence ⇒ align to DNA ⇒ detect
- New gene/transcript — qualitative
  - Sequence RNA ⇒ assemble [⇒ align ⇒ detect]
  - Sequence RNA ⇒ align to DNA ⇒ detect
- Gene/transcript expression — quantitative
  - Sequence RNA ⇒ align to known RNAs ⇒ count
  - Sequence RNA ⇒ align to DNA ⇒ count
- DNA/RNA-Protein binding — quant- & qualitative
  - Binding assay ⇒ Sequence ⇒ align ⇒ identify peaks
- Methylation — quant- & qualitative
  - Bisulfite assay ⇒ Sequence DNA/RNA ⇒ align⇒ count
- ...

Biological Question ⇒ Analysis

- New genome & variation — qualitative
  - Sequence DNA ⇒ assemble [⇒ align ⇒ detect]
  - Sequence ⇒ align to DNA ⇒ detect
- New gene/transcript — qualitative
  - Sequence RNA ⇒ assemble [⇒ align ⇒ detect]
  - Sequence RNA ⇒ align to DNA ⇒ detect
- Gene/transcript expression — quantitative
  - Sequence RNA ⇒ align to known RNAs ⇒ count
  - Sequence RNA ⇒ align to DNA ⇒ count
- DNA/RNA-Protein binding — quant- & qualitative
  - Binding assay ⇒ Sequence ⇒ align ⇒ identify peaks
- Methylation — quant- & qualitative
  - Bisulfite assay ⇒ Sequence DNA/RNA ⇒ align⇒ count
- . . .

Biological Question ⇒ Analysis

- New genome & variation — qualitative
  - Sequence DNA ⇒ assemble [⇒ align ⇒ detect]
  - Sequence ⇒ align to DNA ⇒ detect
- New gene/transcript — qualitative
  - Sequence RNA ⇒ assemble [⇒ align ⇒ detect]
  - Sequence RNA ⇒ align to DNA ⇒ detect
- Gene/transcript expression — quantitative
  - Sequence RNA ⇒ align to known RNAs ⇒ count
  - Sequence RNA ⇒ align to DNA ⇒ count
- DNA/RNA-Protein binding — quant- & qualitative
  - Binding assay ⇒ Sequence ⇒ align ⇒ identify peaks
- Methylation — quant- & qualitative
  - Bisulfite assay ⇒ Sequence DNA/RNA ⇒ align⇒ count
- . . .

Biological Question ⇒ Analysis

- New genome & variation — qualitative
  - Sequence DNA ⇒ assemble [⇒ align ⇒ detect]
  - Sequence ⇒ align to DNA ⇒ detect
- New gene/transcript — qualitative
  - Sequence RNA ⇒ assemble [⇒ align ⇒ detect]
  - Sequence RNA ⇒ align to DNA ⇒ detect
- Gene/transcript expression — quantitative
  - Sequence RNA ⇒ align to known RNAs ⇒ count
  - Sequence RNA ⇒ align to DNA ⇒ count
- DNA/RNA-Protein binding — quant- & qualitative
  - Binding assay ⇒ Sequence ⇒ align ⇒ identify peaks
- Methylation — quant- & qualitative
  - Bisulfite assay ⇒ Sequence DNA/RNA ⇒ align⇒ count
- ...

Sequencing Genomes

- Goal: reconstruct chromosome sequences
- Sequence fragments of DNA
- Assemble fragments into whole genome (hard)
- Works best for long fragments
  - PacBio, 454
  - Long-insert size paired-end sequencing
- Hybrid strategies often work best, e.g.
  - Illumina for contigs, PacBio for scaffolds

RNA Transcripts and Library Preparation

There are many different kinds of RNAs:

- Protein-coding mRNAs
- Noncoding RNAs
  - Structural RNAs (e.g. rRNAs, tRNAs, ...)
  - Small RNAs (e.g. miRNAs, endogenous siRNAs, ...)
  - Antisense / promoter-associated transcripts
  - ...

Analysis of biological sample starts with sample/library preparation.

Depending on which RNAs should be targeted, different preparation strategies have to be used.

Sample/Library Preparation Choices

Directly sequencing total-RNA is suboptimal in most cases:

- rRNA, tRNAs constitute the largest fraction of RNA ($> 90\%$)

Sample preparation choices:

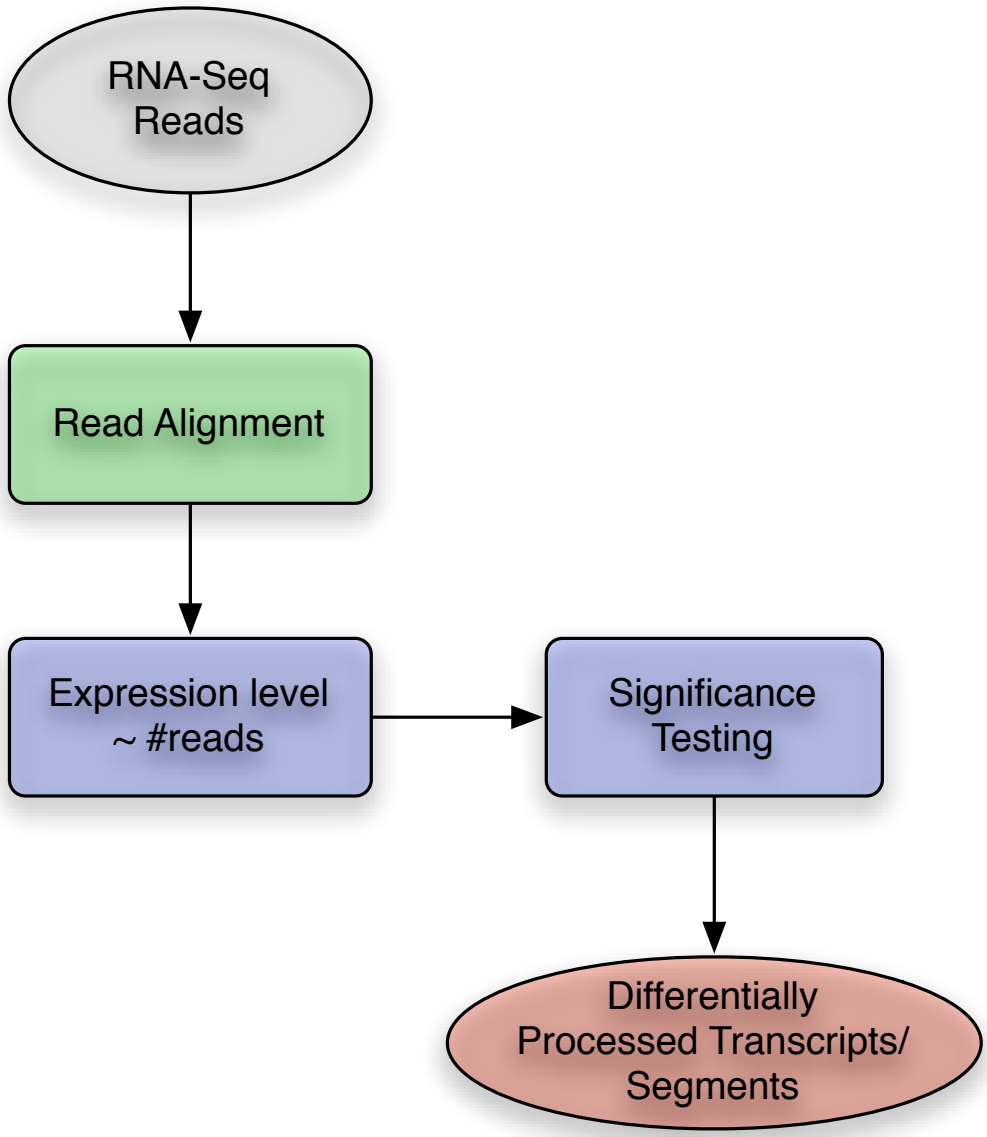
- ribo-minus (rRNA depletion, if it works)
- oligo-dT (selection of poly-adenylated transcripts)
- exonuclease treatment (degrade 5'-P RNAs)

Library preparation options (depend on sequencing technology)

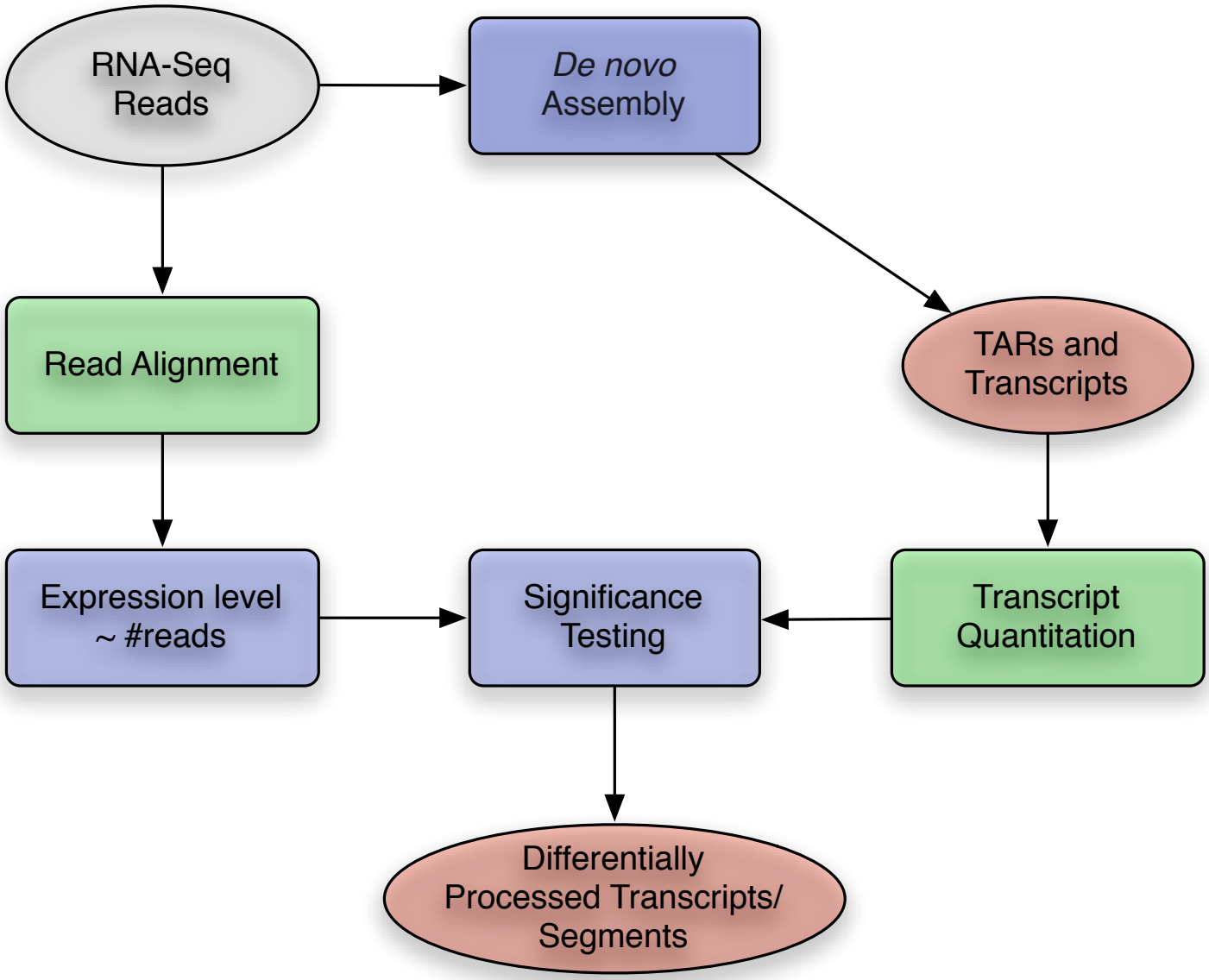
- Strand information
- paired-end, mate-pair sequencing

Most of these steps distort RNA transcript concentrations.

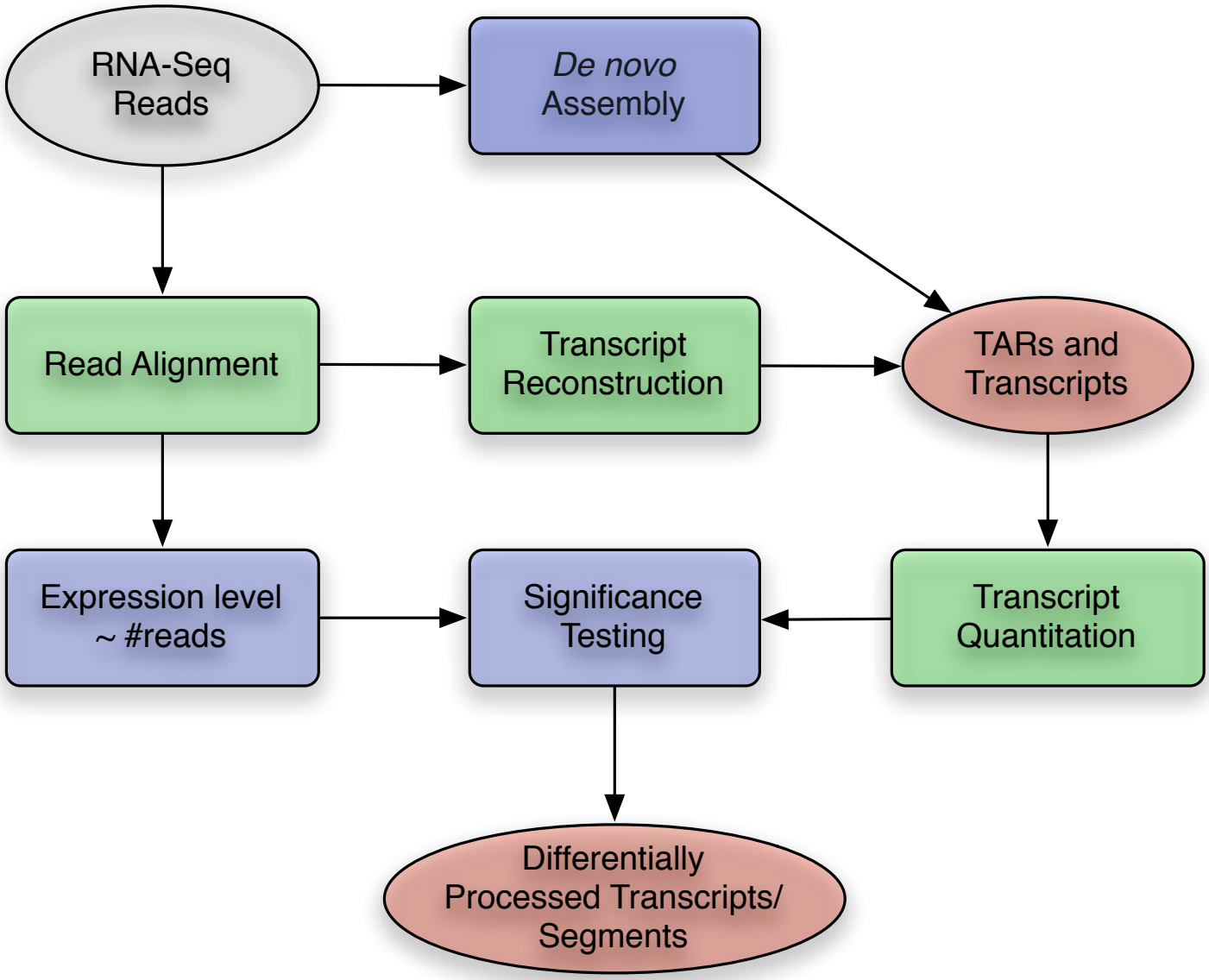
Common RNA-Seq Analysis Steps

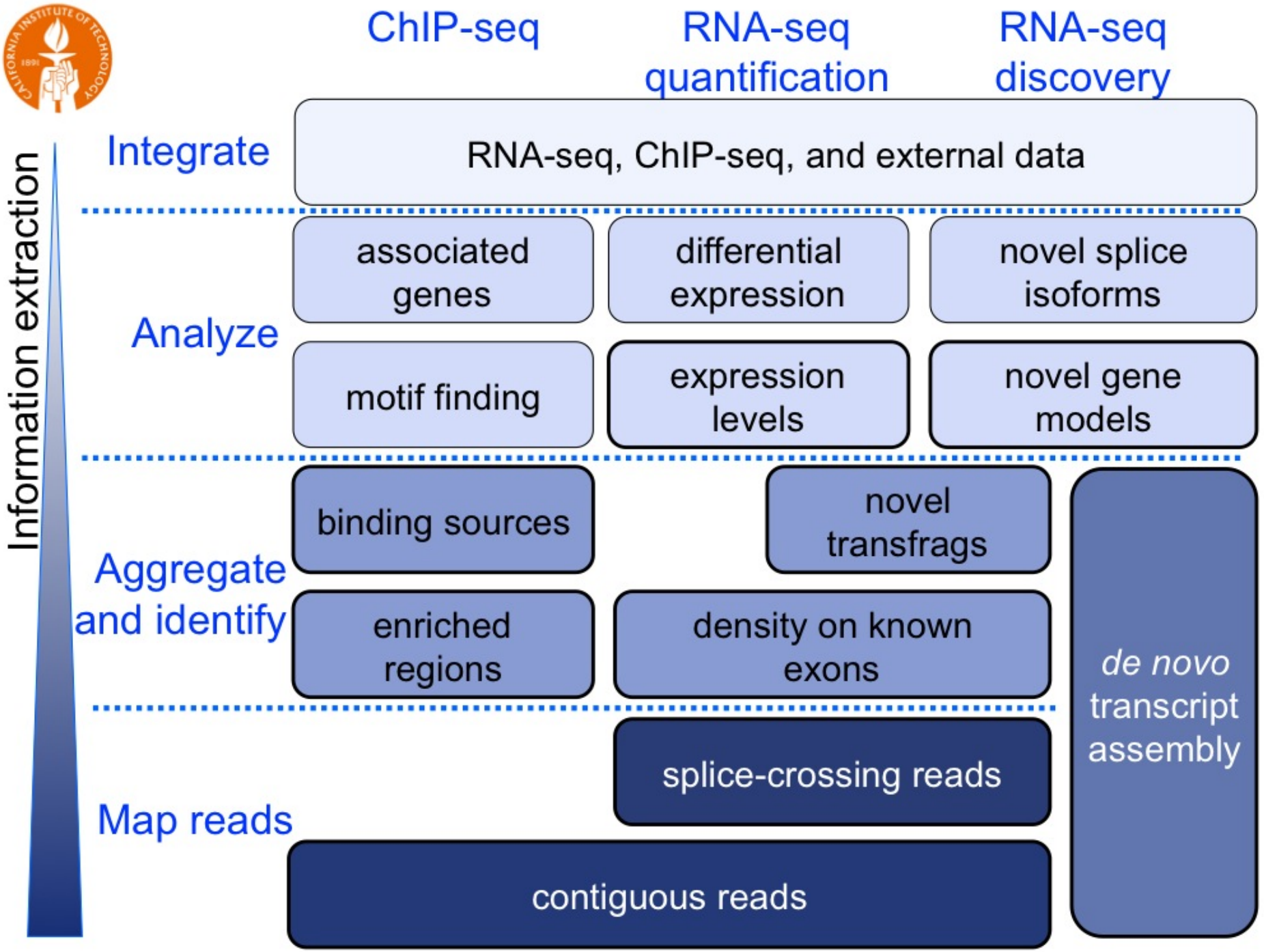


Common RNA-Seq Analysis Steps



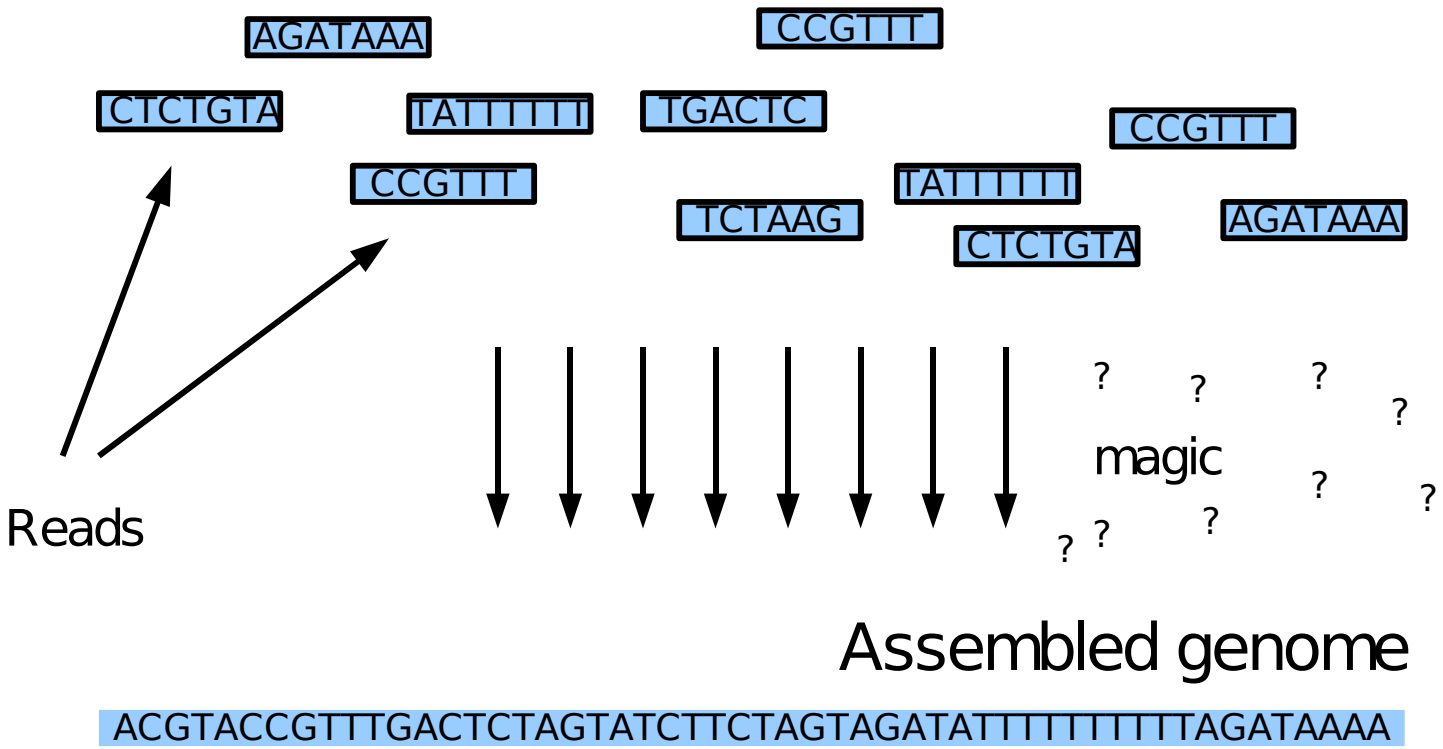
Common RNA-Seq Analysis Steps




ChIP-seq
RNA-seq quantification
RNA-seq discovery
Information extraction
Integrate
RNA-seq, ChIP-seq, and external data
Analyze
associated genes
differential expression
novel splice isoforms
motif finding
expression levels
novel gene models
Aggregate and identify
binding sources
novel transfrags
enriched regions
density on known exons
de novo transcript assembly
Map reads
splice-crossing reads
contiguous reads

Basic Reads Analysis

- Assembly
  ⇒ generate contigs
- Mapping/Alignments
  ⇒ map/align reads back to a known genome



[Sutskever, 2008]

Basic Reads Analysis

- Assembly
  ⇒ generate contigs
- Mapping/Alignments
  ⇒ map/align reads back to a known genome

```
DNA          ...GCAAACCAGTGACCTGACTACTACGTCGTAACGTACACGGTAGCT...
Read 1           AACCAGTGACCTGACTACTACGTCGTAACGTA
Read 2364         ACCAGTGACCTGACTACTACGTCGTAACGTAC
Read 32442         CCAGTGACCTGACTACTACGTCGTAACGTACA
...                 CAGTGACCTGACTACTACGTCGTAACGTACAC
                     AGTGACCTGACTACTACGTCGTAACGTACACG
```

Problem: hundreds of millions of reads of short length

⇒ Big computational challenge

Costs of sequencing are comparable with costs of analysis!

≈ $1000 for sequencing (1 lane) vs. ≈ $250 for analysis (cloud)
(Analysis often performed more than once . . .)

Basic Reads Analysis

- Assembly
  ⇒ generate contigs
- Mapping/Alignments
  ⇒ map/align reads back to a known genome

```
DNA          ...GCAAACCAGTGACCTGACTACTACGTCGTAACGTACACGGTAGCT...
Read 1            AACCAGTGACCTGACTACTACGTCGTAACGTA
Read 2364          ACCAGTGACCTGACTACTACGTCGTAACGTAC
Read 32442          CCAGTGACCTGACTACTACGTCGTAACGTACA
...                  CAGTGACCTGACTACTACGTCGTAACGTACAC
                      AGTGACCTGACTACTACGTCGTAACGTACACG
```

Problem: hundreds of millions of reads of short length

⇒ Big computational challenge

Costs of sequencing are comparable with costs of analysis!

≈ $1000 for sequencing (1 lane) vs. ≈ $250 for analysis (cloud)
(Analysis often performed more than once . . .)

Basic Reads Analysis

- Assembly
  ⇒ generate contigs
- Mapping/Alignments
  ⇒ map/align reads back to a known genome

```
DNA          ...GCAAACCAGTGACCTGACTACTACGTCGTAACGTACACGGTAGCT...
Read 1            AACCAGTGACCTGACTACTACGTCGTAACGTA
Read 2364          ACCAGTGACCTGACTACTACGTCGTAACGTAC
Read 32442          CCAGTGACCTGACTACTACGTCGTAACGTACA
...                  CAGTGACCTGACTACTACGTCGTAACGTACAC
                      AGTGACCTGACTACTACGTCGTAACGTACACG
```

Problem: hundreds of millions of reads of short length

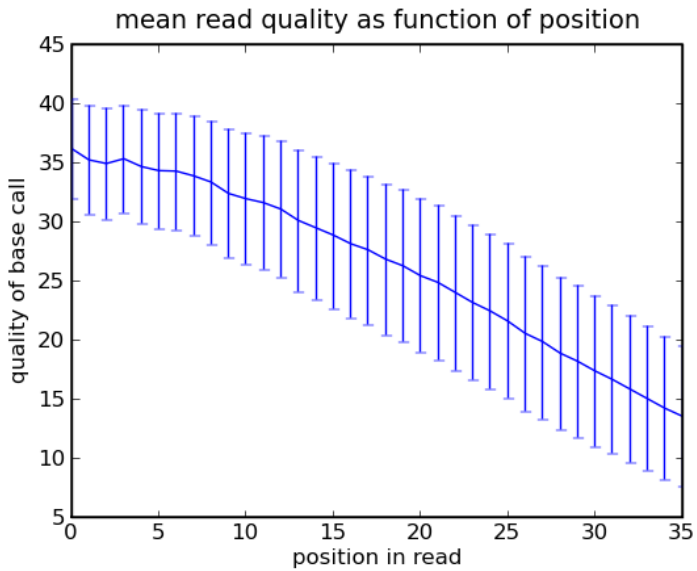
⇒ Big computational challenge

Costs of sequencing are comparable with costs of analysis!

≈ \$1000 for sequencing (1 lane) vs. ≈ \$250 for analysis (cloud)
(Analysis often performed more than once . . .)

Basic Read Analysis II

- Experiment leads to millions of reads $36 - 200$nt
- Reads may have a position-wise varying quality



- Quality corresponds to error probability:

$$q = -10 \cdot log_{10}\left(\frac{p}{1-p}\right)$$

- Example: $p = 10^{-3}$ per base corresponds to $q = 30$
- Typical read error probability for Illumina reads $\approx 1\%$

Read Analysis - Mapping

Read mapping problem

For each read find its target regions on the reference genome such that there are at most k mismatches/indels between read and target.

- Global/local alignment of all reads prohibitive
- A read stems from a certain small region
- Find this region and then do an alignment
  - (Spaced) seeds
  - Suffix trees/arrays
  - Burrows-Wheeler
- Common tools: bowtie [Langmead et al., 2009], bwa [Li and Durbin, 2009, 2010], *GenomeMapper* [Schneeberger et al., 2009a], *Shrimp* [Rumble et al., 2009], *SOAP(2)* [Li et al., 2009], *VMATCH*, MAQ [Li et al., 2008], ELAND, segemehl [Hoffmann et al., 2009], . . . (≈50 more)
- Main issues:
  - Accuracy
  - Speed
  - Memory Consumption

Read Analysis - Mapping

Read mapping problem

For each read find its target regions on the reference genome such that there are at most k mismatches/indels between read and target.

- Global/local alignment of all reads prohibitive
- A read stems from a certain small region
- Find this region and then do an alignment
  - (Spaced) seeds
  - Suffix trees/arrays
  - Burrows-Wheeler
- Common tools: bowtie [Langmead et al., 2009], bwa [Li and Durbin, 2009, 2010], *GenomeMapper* [Schneeberger et al., 2009a], *Shrimp* [Rumble et al., 2009], *SOAP(2)* [Li et al., 2009], *VMATCH*, MAQ [Li et al., 2008], ELAND, segemehl [Hoffmann et al., 2009], . . . (≈50 more)
- Main issues:
  - Accuracy
  - Speed
  - Memory Consumption

Read Analysis - Mapping

Read mapping problem

For each read find its target regions on the reference genome such that there are at most k mismatches/indels between read and target.

- Global/local alignment of all reads prohibitive
- A read stems from a certain small region
- Find this region and then do an alignment
  - (Spaced) seeds
  - Suffix trees/arrays
  - Burrows-Wheeler
- Common tools: bowtie [Langmead et al., 2009], bwa [Li and Durbin, 2009, 2010], *GenomeMapper* [Schneeberger et al., 2009a], *Shrimp* [Rumble et al., 2009], *SOAP(2)* [Li et al., 2009], *VMATCH*, MAQ [Li et al., 2008], ELAND, segemehl [Hoffmann et al., 2009], . . . (≈50 more)
- Main issues:
  - Accuracy
  - Speed
  - Memory Consumption

Read Analysis - Mapping

Read mapping problem

For each read find its target regions on the reference genome such that there are at most k mismatches/indels between read and target.

- Global/local alignment of all reads prohibitive
- A read stems from a certain small region
- Find this region and then do an alignment
  - (Spaced) seeds
  - Suffix trees/arrays
  - Burrows-Wheeler
- Common tools: bowtie [Langmead et al., 2009], bwa [Li and Durbin, 2009, 2010], *GenomeMapper* [Schneeberger et al., 2009a], *Shrimp* [Rumble et al., 2009], *SOAP(2)* [Li et al., 2009], *VMATCH*, MAQ [Li et al., 2008], ELAND, segemehl [Hoffmann et al., 2009], . . . (≈50 more)
- Main issues:
  - Accuracy
  - Speed
  - Memory Consumption

Read Analysis - Mapping

Read mapping problem

For each read find its target regions on the reference genome such that there are at most k mismatches/indels between read and target.

- Global/local alignment of all reads prohibitive
- A read stems from a certain small region
- Find this region and then do an alignment
  - (Spaced) seeds
  - Suffix trees/arrays
  - Burrows-Wheeler
- Common tools: bowtie [Langmead et al., 2009], bwa [Li and Durbin, 2009, 2010], *GenomeMapper* [Schneeberger et al., 2009a], *Shrimp* [Rumble et al., 2009], *SOAP(2)* [Li et al., 2009], *VMATCH*, MAQ [Li et al., 2008], ELAND, segemehl [Hoffmann et al., 2009], . . . (≈50 more)
- Main issues:
  - Accuracy
  - Speed
  - Memory Consumption

Data Structures for Efficiently Finding Strings

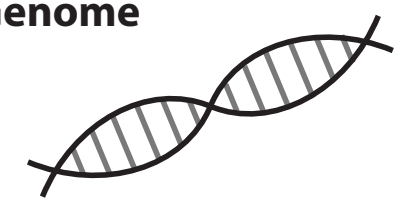
	k-mer index	Suffix Tree	Suffix Array	BWT & FM Index
Space	$4L + \|\Sigma\|^k$	$\approx 12L$	$\approx 4L$	$\ll L$
Time	$O(p)$	$O(p \log L)$	$O(p \log L)$	$O(p + N \log L)$

Indexing based on k-mers

Steps to generate a k-mer index

- Take the genome as a linear string of alphabet A, C, G, T
- Generate a list of all words of length k in the genome string (k-mer) and create a unique list
- Store for each k-mer all positions in the genome where it occurs

Genome



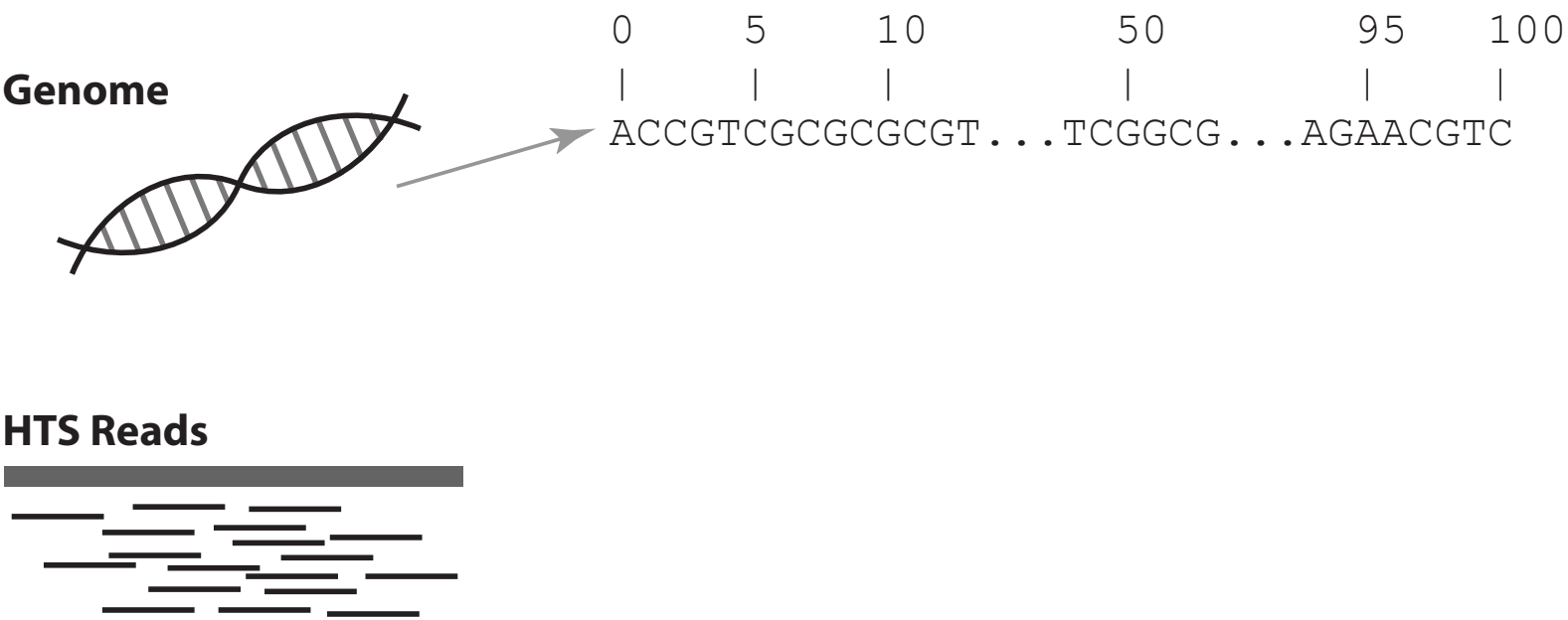
HTS Reads



Indexing based on k-mers

Steps to generate a k-mer index

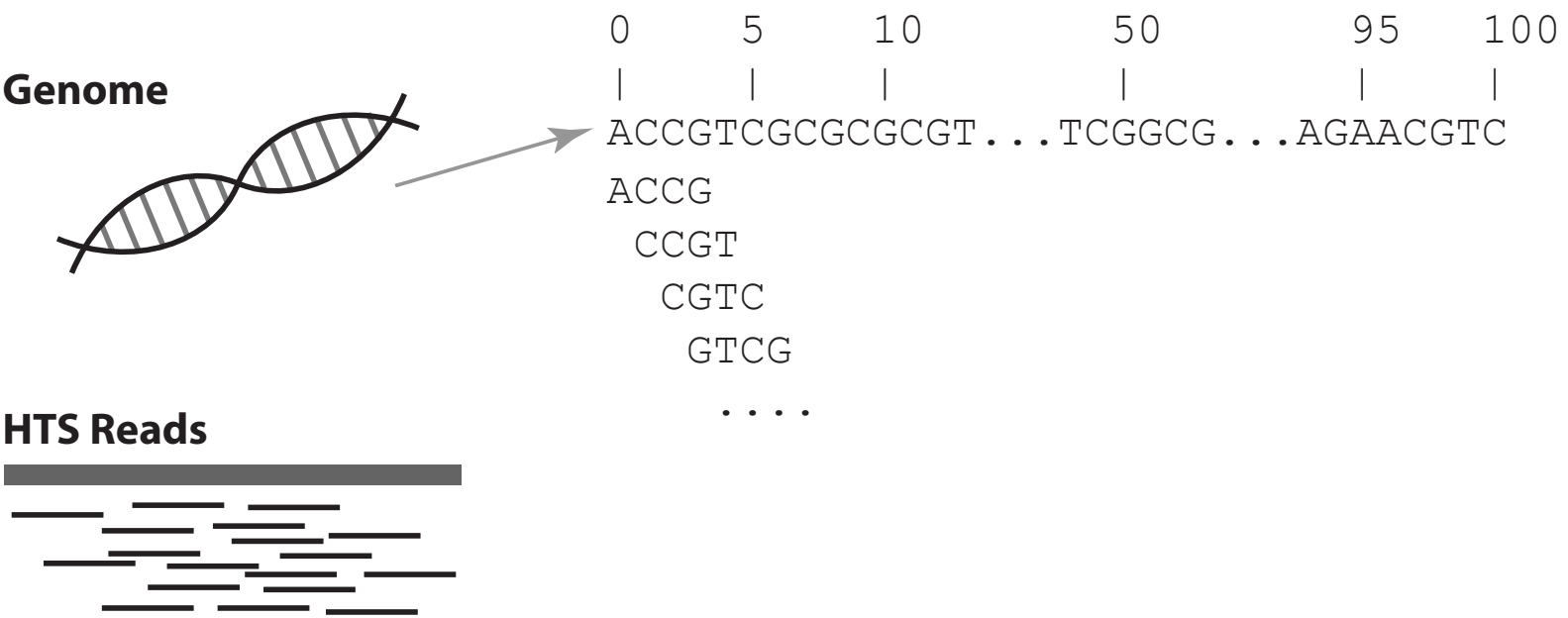
- Take the genome as a linear string of alphabet A, C, G, T
- Generate a list of all words of length k in the genome string (k-mer) and create a unique list
- Store for each k-mer all positions in the genome where it occurs



Indexing based on k-mers

Steps to generate a k-mer index

- Take the genome as a linear string of alphabet A, C, G, T
- Generate a list of all words of length k in the genome string (k-mer) and create a unique list
- Store for each k-mer all positions in the genome where it occurs



Indexing based on k-mers

Steps to generate a k-mer index

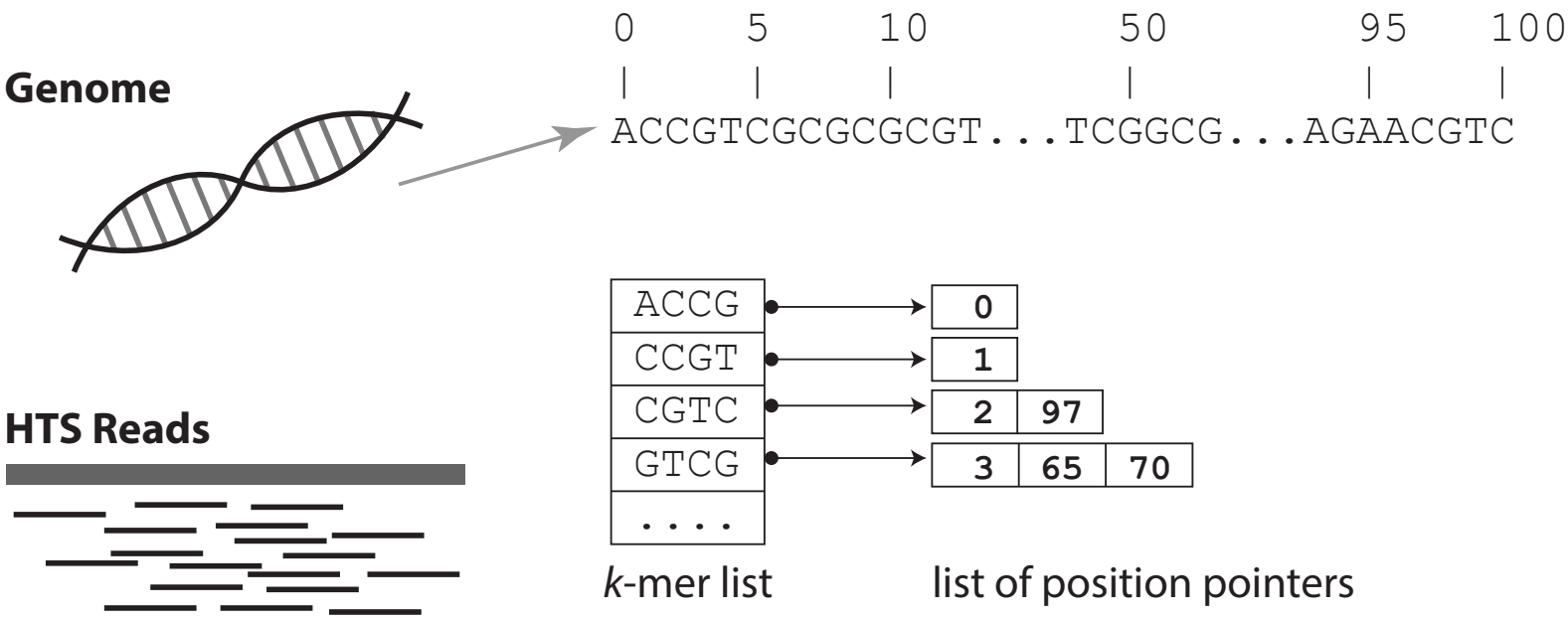
- Take the genome as a linear string of alphabet A, C, G, T
- Generate a list of all words of length k in the genome string (k-mer) and create a unique list
- Store for each k-mer all positions in the genome where it occurs



Indexing based on k-mers

Steps to generate a k-mer index

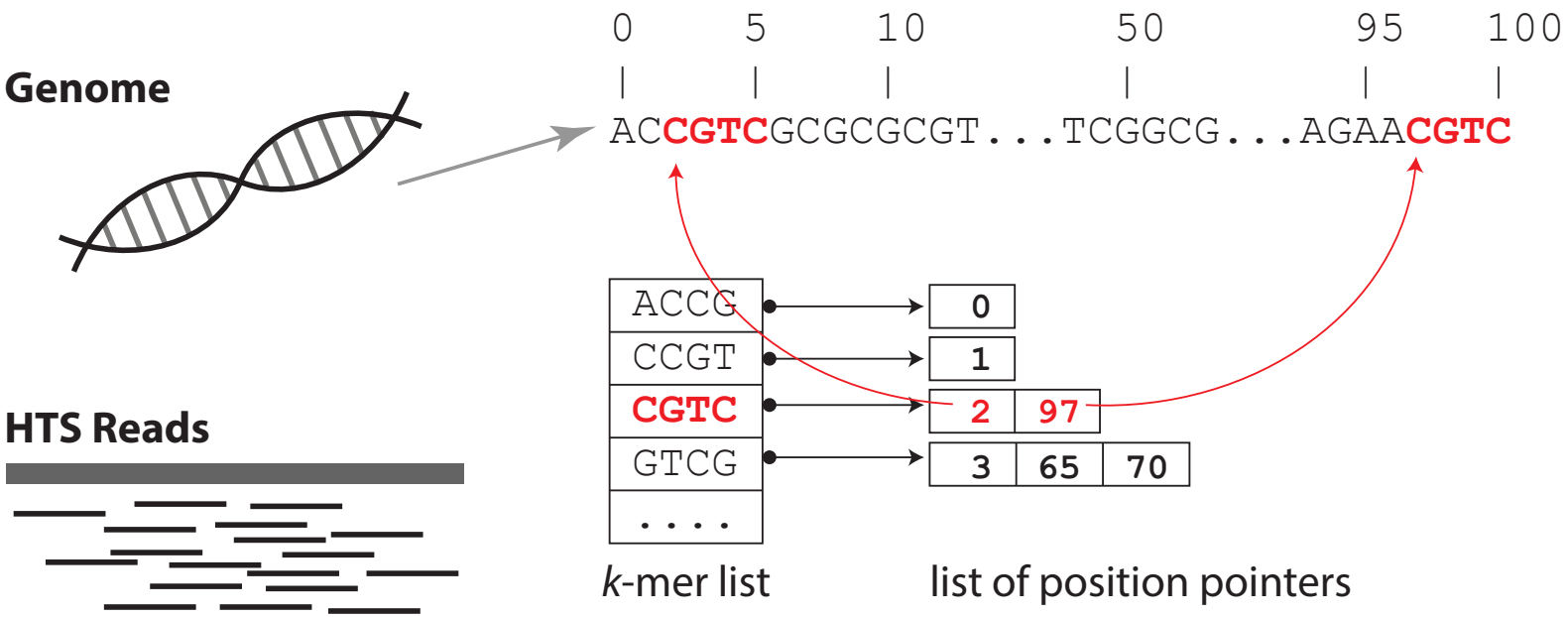
- Take the genome as a linear string of alphabet A, C, G, T
- Generate a list of all words of length k in the genome string (k-mer) and create a unique list
- Store for each k-mer all positions in the genome where it occurs



Indexing based on k-mers

Steps to generate a k-mer index

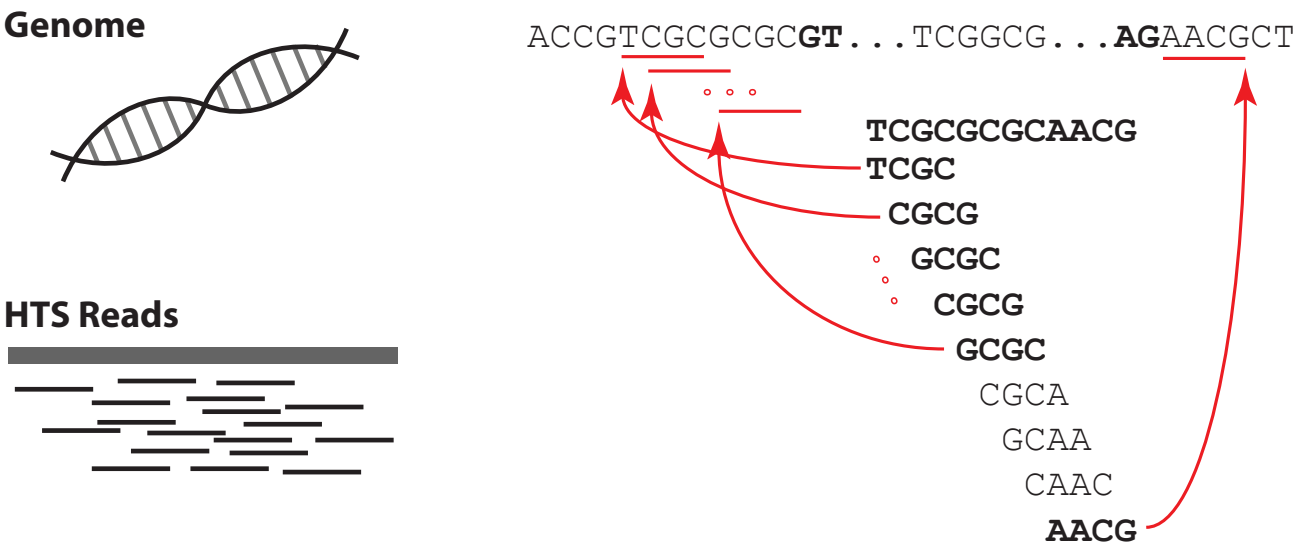
- Take the genome as a linear string of alphabet A, C, G, T
- Generate a list of all words of length k in the genome string (k-mer) and create a unique list
- Store for each k-mer all positions in the genome where it occurs



Read Alignment with PALMapper

Case for DNA read mapping:

- Alignments based on GenomeMapper
- Based on a *k*-mer index, long seeds (16) are best suited for large genomes
- Connect seed hits into segments that trigger local alignments

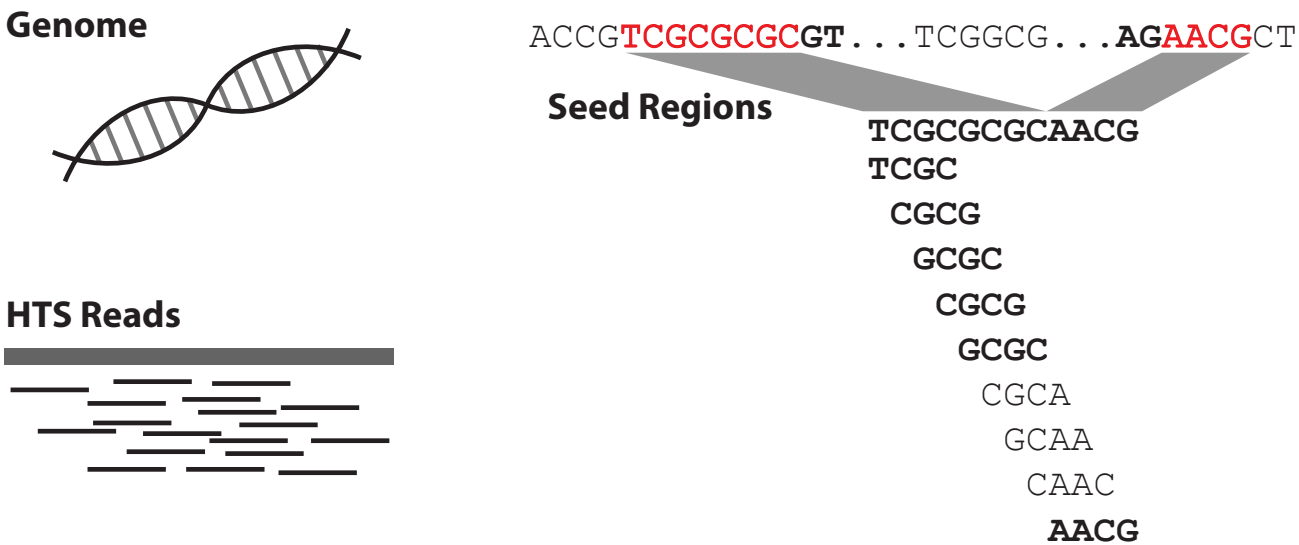


[Schneeberger et al., 2009b, Jean et al., 2010]

Read Alignment with PALMapper

Case for DNA read mapping:

- Alignments based on GenomeMapper
- Based on a *k*-mer index, long seeds (16) are best suited for large genomes
- Connect seed hits into segments that trigger local alignments



[Schneeberger et al., 2009b, Jean et al., 2010]

Spaced Seeding*

Use the same number of positions k, but ...

- ... introduce gaps in the word
- ... use different gap-patterns for different words
- ... use different spacing of the words

```
0     5     10          50          95    100
|     |     |           |           |     |
ACCGTCGCGCGCGT...TCGGCG...AGAACGTC
AC--TC--GC
 CC--CG--CG
  CG--GC--GC
       ...
```

Spaced Seeding*

Use the same number of positions k, but ...

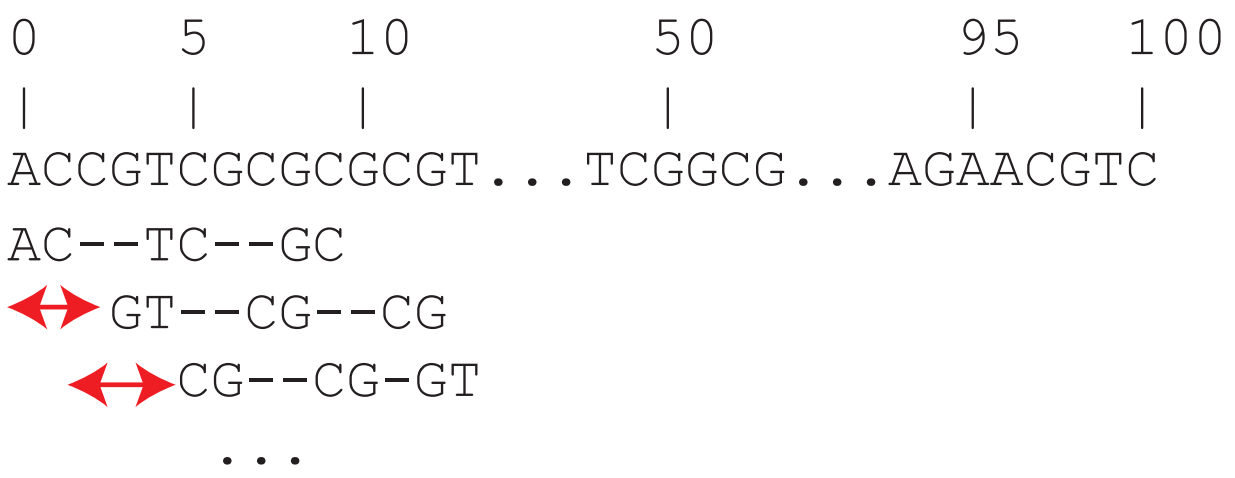
- ... introduce gaps in the word
- ... use different gap-patterns for different words
- ... use different spacing of the words

```
0     5     10          50          95    100
|     |     |           |           |     |
ACCGTCGCGCGCGT...TCGGCG...AGAACGTC
AC--TC--GC
 CCG-----CGC
  C--CGCG--C
       ...
```

Spaced Seeding*

Use the same number of positions k, but ...

- ... introduce gaps in the word
- ... use different gap-patterns for different words
- ... use different spacing of the words



Suffix of a String

Given a string $S = \text{TAGAGA}$, a suffix S_i is the substring of S that starts at position i:

Example

$S = \text{TAGAGA}$

$S_2 = \text{AGAGA}$
$S_5 = \text{GA}$

Suffix Trees

Building a Suffix Tree

- Start with full string S and an empty root
- Add successively each suffix, such that edge labels on path from root to leaf spell the suffix; label leaf with position
- Use existing edges where possible

1234567

string S ⟶ TAGAGA$



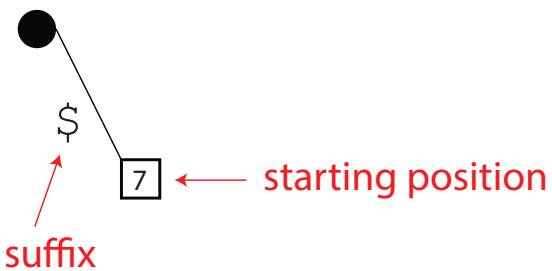
[Weiner, 1973]

Suffix Trees

Building a Suffix Tree

- Start with full string S and an empty root
- Add successively each suffix, such that edge labels on path from root to leaf spell the suffix; label leaf with position
- Use existing edges where possible

```
1234567
TAGAGA$
      $  7
```



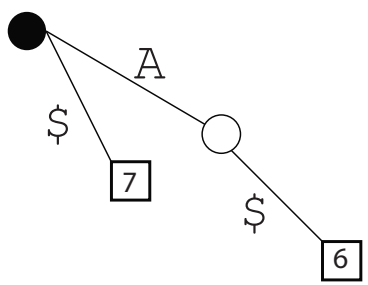
[Weiner, 1973]

Suffix Trees

Building a Suffix Tree

- Start with full string S and an empty root
- Add successively each suffix, such that edge labels on path from root to leaf spell the suffix; label leaf with position
- Use existing edges where possible

```
1234567
TAGAGA$
      $ 7
     A$ 6
```

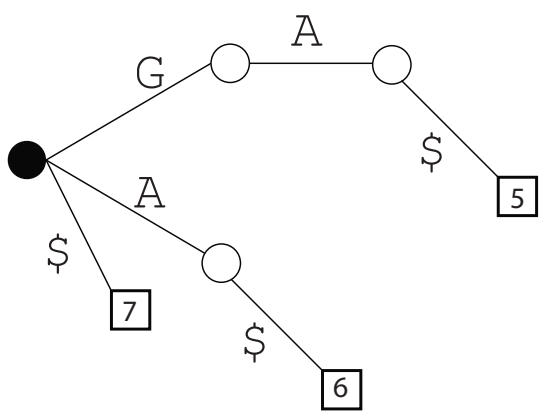


[Weiner, 1973]

Suffix Trees

Building a Suffix Tree

- Start with full string S and an empty root
- Add successively each suffix, such that edge labels on path from root to leaf spell the suffix; label leaf with position
- Use existing edges where possible



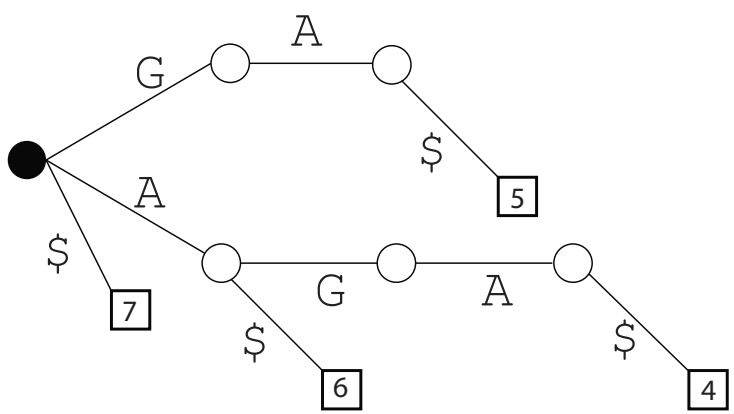

```
1234567
TAGAGA$
      $  7
     A$  6
    GA$  5
```

[Weiner, 1973]

Suffix Trees

Building a Suffix Tree

- Start with full string S and an empty root
- Add successively each suffix, such that edge labels on path from root to leaf spell the suffix; label leaf with position
- Use existing edges where possible



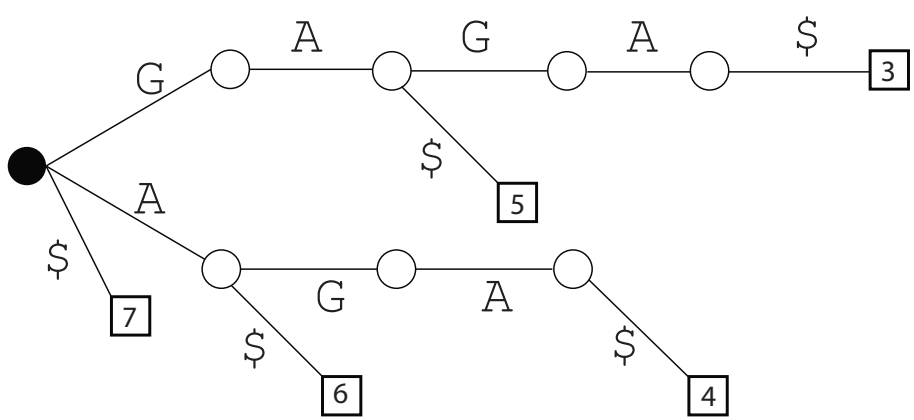

```
1234567
TAGAGA$
      $  7
     A$  6
    GA$  5
   AGA$  4
```

[Weiner, 1973]

Suffix Trees

Building a Suffix Tree

- Start with full string S and an empty root
- Add successively each suffix, such that edge labels on path from root to leaf spell the suffix; label leaf with position
- Use existing edges where possible



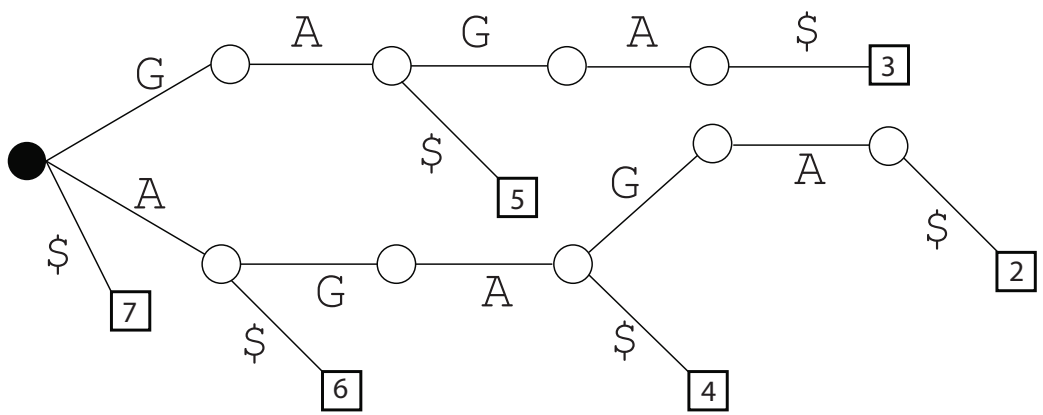

```
1234567
TAGAGA$
      $  7
     A$  6
    GA$  5
   AGA$  4
  GAGA$  3
```

[Weiner, 1973]

Suffix Trees

Building a Suffix Tree

- Start with full string S and an empty root
- Add successively each suffix, such that edge labels on path from root to leaf spell the suffix; label leaf with position
- Use existing edges where possible



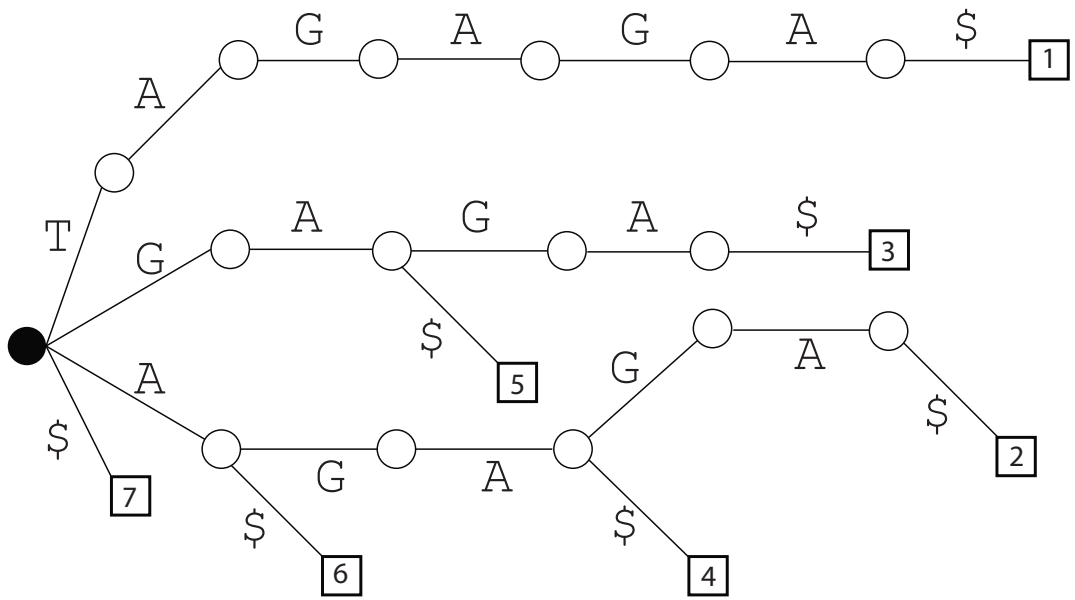

```
1234567
TAGAGA$
      $  7
     A$  6
    GA$  5
   AGA$  4
  GAGA$  3
 AGAGA$  2
```

[Weiner, 1973]

Suffix Trees

Building a Suffix Tree

- Start with full string S and an empty root
- Add successively each suffix, such that edge labels on path from root to leaf spell the suffix; label leaf with position
- Use existing edges where possible



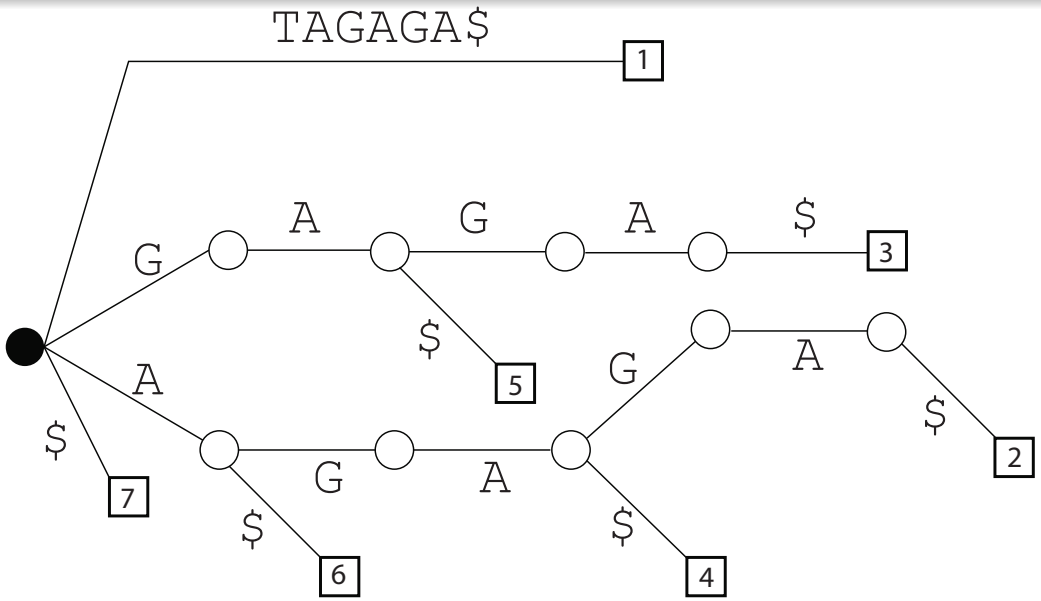

```
1234567
TAGAGA$
      $  7
     A$  6
    GA$  5
   AGA$  4
  GAGA$  3
 AGAGA$  2
TAGAGA$  1
```

[Weiner, 1973]

Suffix Trees

Building a Suffix Tree (cont)

- Contract edges that have no branches and concatenate labels
- Repeat for all edges
- The number of leaves does not change!



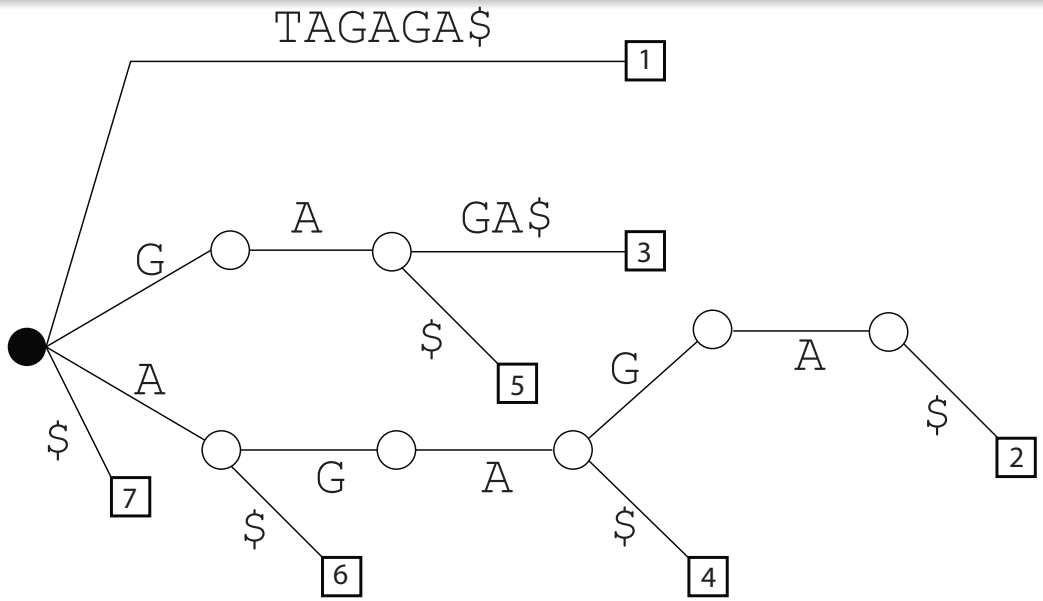

```
1234567
TAGAGA$
      $ 7
     A$ 6
    GA$ 5
   AGA$ 4
  GAGA$ 3
 AGAGA$ 2
TAGAGA$ 1
```

[Weiner, 1973]

Suffix Trees

Building a Suffix Tree (cont)

- Contract edges that have no branches and concatenate labels
- Repeat for all edges
- The number of leaves does not change!



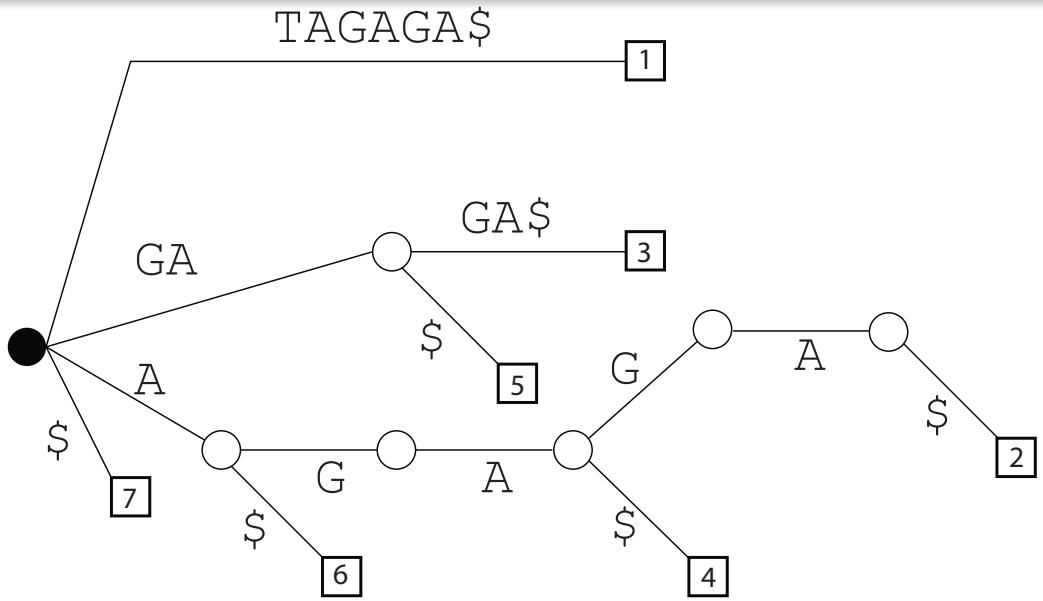

```
1234567
TAGAGA$
      $ 7
     A$ 6
    GA$ 5
   AGA$ 4
  GAGA$ 3
 AGAGA$ 2
TAGAGA$ 1
```

[Weiner, 1973]

Suffix Trees

Building a Suffix Tree (cont)

- Contract edges that have no branches and concatenate labels
- Repeat for all edges
- The number of leaves does not change!



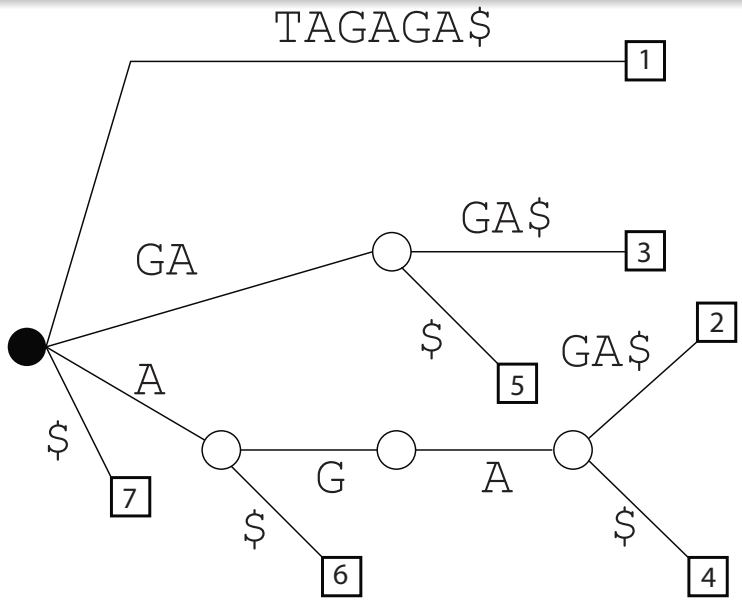

```
1234567
TAGAGA$
      $ 7
     A$ 6
    GA$ 5
   AGA$ 4
  GAGA$ 3
 AGAGA$ 2
TAGAGA$ 1
```

[Weiner, 1973]

Suffix Trees

Building a Suffix Tree (cont)

- Contract edges that have no branches and concatenate labels
- Repeat for all edges
- The number of leaves does not change!



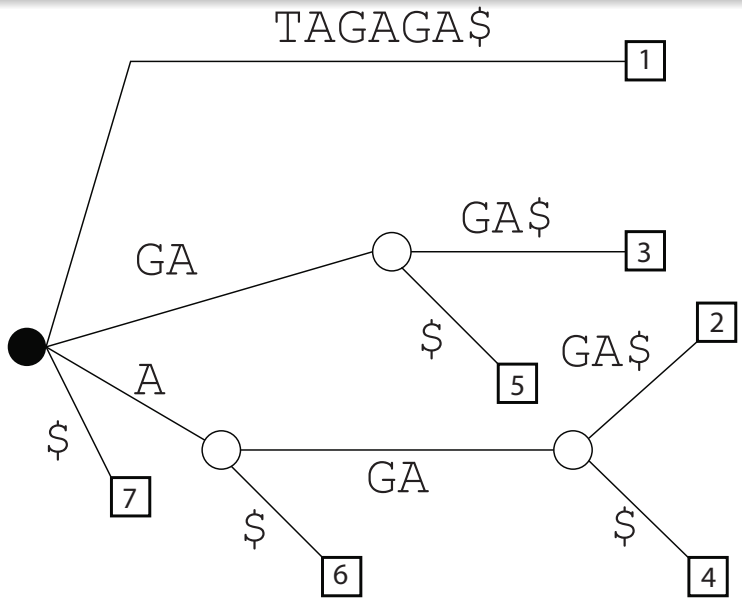

```
1234567
TAGAGA$
      $ 7
     A$ 6
    GA$ 5
   AGA$ 4
  GAGA$ 3
 AGAGA$ 2
TAGAGA$ 1
```

[Weiner, 1973]

Suffix Trees

Building a Suffix Tree (cont)

- Contract edges that have no branches and concatenate labels
- Repeat for all edges
- The number of leaves does not change!



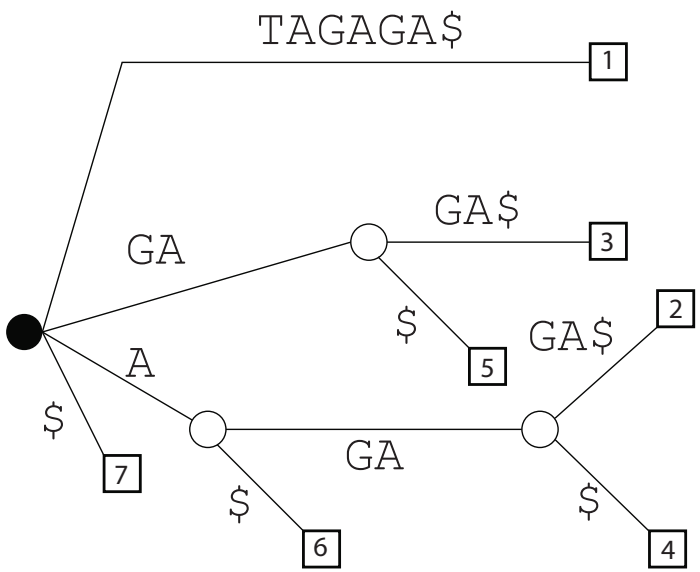

```
1234567
TAGAGA$
      $ 7
     A$ 6
    GA$ 5
   AGA$ 4
  GAGA$ 3
 AGAGA$ 2
TAGAGA$ 1
```

[Weiner, 1973]

Suffix Trees

Pattern Matching

- Start at root
- Traverse tree according to pattern
- Stop, if leaf or end of pattern is reached
- Find start positions of P in S in the leaves below of current sub-tree

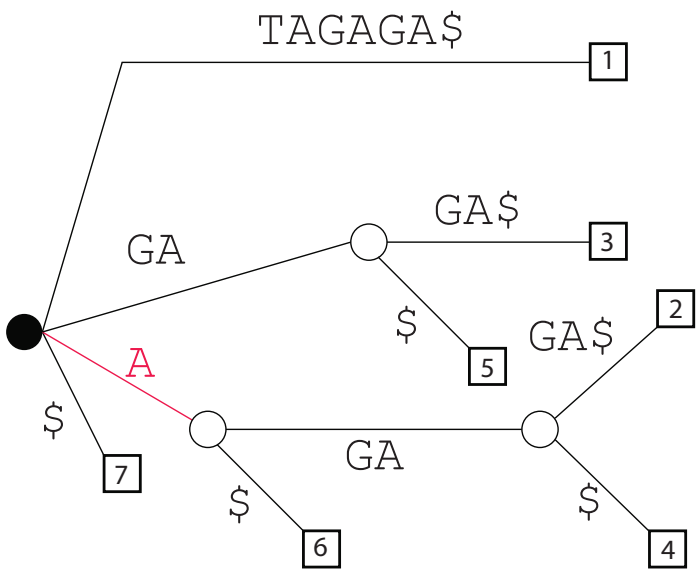


S = TAGAGA
P = AGA

Suffix Trees

Pattern Matching

- Start at root
- Traverse tree according to pattern
- Stop, if leaf or end of pattern is reached
- Find start positions of P in S in the leaves below of current sub-tree



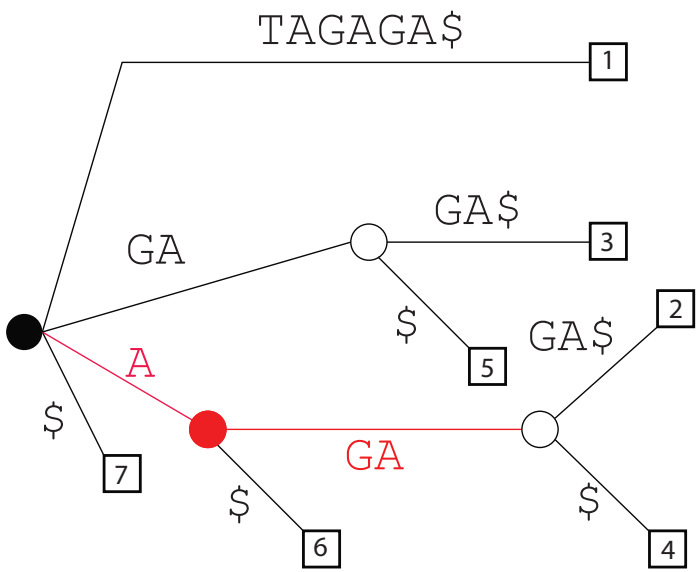
S = TAGAGA
P = AGA

A

Suffix Trees

Pattern Matching

- Start at root
- Traverse tree according to pattern
- Stop, if leaf or end of pattern is reached
- Find start positions of P in S in the leaves below of current sub-tree



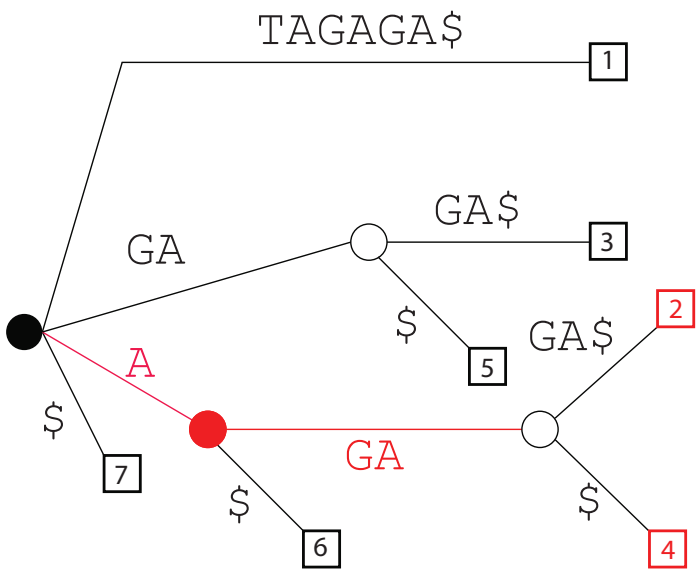
S = TAGAGA
P = AGA

A
AGA

Suffix Trees

Pattern Matching

- Start at root
- Traverse tree according to pattern
- Stop, if leaf or end of pattern is reached
- Find start positions of P in S in the leaves below of current sub-tree



S = TAGAGA
P = AGA

A
AGA
⟹ positions 2 and 4

Suffix Trees

Complexity

Construction for string S of length n:

- $O(n)$ for a constant size alphabet ($O(n \log n)$ general)
- On-line construction after Ukkonen $O(n)$

Tasks:

- Search in S for a pattern P of length p: $O(p)$
- Find matching statistics for P with p longest match between S and P: $\Theta(p)$

Search complexity is defined by the length of the pattern! Very useful for alignment.

[McCreight, 1976, Ukkonen, 1995]

Suffix Trees

Complexity

Construction for string S of length n:

- $O(n)$ for a constant size alphabet ($O(n \log n)$ general)
- On-line construction after Ukkonen $O(n)$

Tasks:

- Search in S for a pattern P of length p: $O(p)$
- Find matching statistics for P with p longest match between S and P: $\Theta(p)$

Search complexity is defined by the length of the pattern! Very useful for alignment.

[McCreight, 1976, Ukkonen, 1995]

Suffix Arrays

Definition and Generation

- Sorted list of all suffixes of a string S
- Generate list of suffixes, sort them lexicographical and
- Can be generated by a depth-first traversal of the suffix tree

TAGAGA position in S

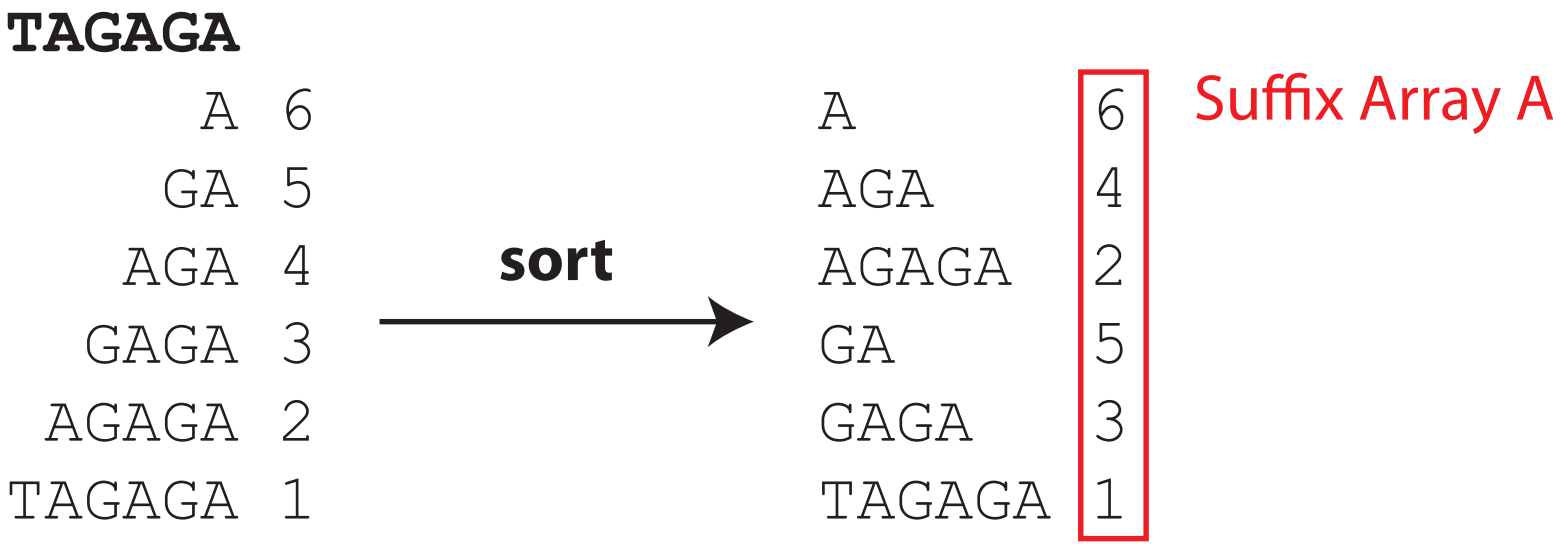
```
     A  6
    GA  5
   AGA  4
  GAGA  3
 AGAGA  2
TAGAGA  1
```

[Manber and Myers, 1993]

Suffix Arrays

Definition and Generation

- Sorted list of all suffixes of a string S
- Generate list of suffixes, sort them lexicographical and
- Can be generated by a depth-first traversal of the suffix tree

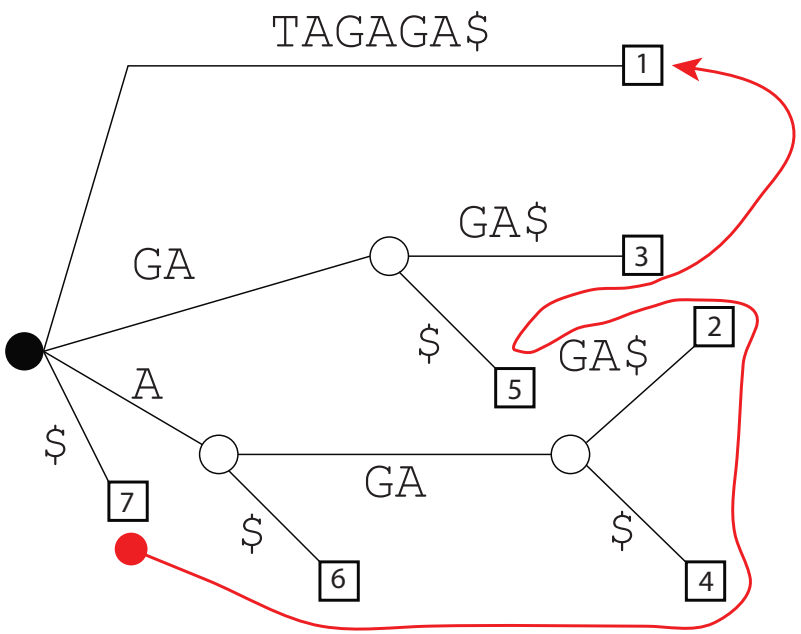


[Manber and Myers, 1993]

Suffix Arrays

Definition and Generation

- Sorted list of all suffixes of a string S
- Generate list of suffixes, sort them lexicographical and
- Can be generated by a depth-first traversal of the suffix tree



Suffix Array

A = 7 6 4 2 5 3 1

[Manber and Myers, 1993]

Suffix Arrays

Pattern Matching

- **Suffix Array** A: $A[k]$ is the start pos. of k-th smallest suffix in $\{S_0, S_1, ..., S_{n-1}\}$, where S_i is the suffix at pos. i in S
- **Search Pattern** P: is a substring of length $p \leq n$
- **Left Boundary** $L_P = min(k : P \leq S_{A[k]} \text{ or } k = n)$
- **Right Boundary** $R_P = max(k : S_{A[k]} \leq P \text{ or } k = -1)$

Suffix Arrays

Pattern Matching

- **Suffix Array** A: $A[k]$ is the start pos. of k-th smallest suffix in $\{S_0, S_1, ..., S_{n-1}\}$, where S_i is the suffix at pos. i in S
- **Search Pattern** P: is a substring of length $p \leq n$
- **Left Boundary** $L_P = min(k : P \leq S_{A[k]} \text{ or } k = n)$
- **Right Boundary** $R_P = max(k : S_{A[k]} \leq P \text{ or } k = -1)$

Suffix Arrays

Pattern Matching

- **Suffix Array** A: $A[k]$ is the start pos. of k-th smallest suffix in $\{S_0, S_1, ..., S_{n-1}\}$, where S_i is the suffix at pos. i in S
- **Search Pattern** P: is a substring of length $p \leq n$
- **Left Boundary** $L_P = min(k : P \leq S_{A[k]} \text{ or } k = n)$
- **Right Boundary** $R_P = max(k : S_{A[k]} \leq P \text{ or } k = -1)$

Suffix Arrays

Pattern Matching

- **Suffix Array** A: $A[k]$ is the start pos. of k-th smallest suffix in $\{S_0, S_1, ..., S_{n-1}\}$, where S_i is the suffix at pos. i in S
- **Search Pattern** P: is a substring of length $p \leq n$
- **Left Boundary** $L_P = min(k : P \leq S_{A[k]} \text{ or } k = n)$
- **Right Boundary** $R_P = max(k : S_{A[k]} \leq P \text{ or } k = -1)$

A	6
AGA	4
AGAGA	2
GA	5
GAGA	3
TAGAGA	1

P = AGA

L_P = 2 ==> A[2] = 4 ==> S_4= AGA

R_P = 3 ==> A[3] = 2 ==> S_2= AGAGA

Suffix Arrays*

Search and Complexity

- L_P and R_P fully define all matches of P in S
- Boundaries can be efficiently computed with binary search, pattern matching in $O(|P| + log|S|)$ time
- Construction and storage of A in $O(|S|)$

Burrows-Wheeler Transformation (BWT)

How does it work?

- Start with the full genome string (TAGAGA)
- Generate all rotations of the genome string
- Sort the matrix lexicographically
- Keep only the last column L and the index of the original string

TAGAGA

[Burrows and Wheeler, 1994]

Burrows-Wheeler Transformation (BWT)

How does it work?

- Start with the full genome string (TAGAGA)
- Generate all rotations of the genome string
- Sort the matrix lexicographically
- Keep only the last column L and the index of the original string

1 **T**AGAGA
2 A**T**AGAG
3 GA**T**AGA
4 AGA**T**AG
5 GAGA**T**A
6 AGAGA**T**

[Burrows and Wheeler, 1994]

Burrows-Wheeler Transformation (BWT)

How does it work?

- Start with the full genome string (TAGAGA)
- Generate all rotations of the genome string
- Sort the matrix lexicographically
- Keep only the last column L and the index of the original string

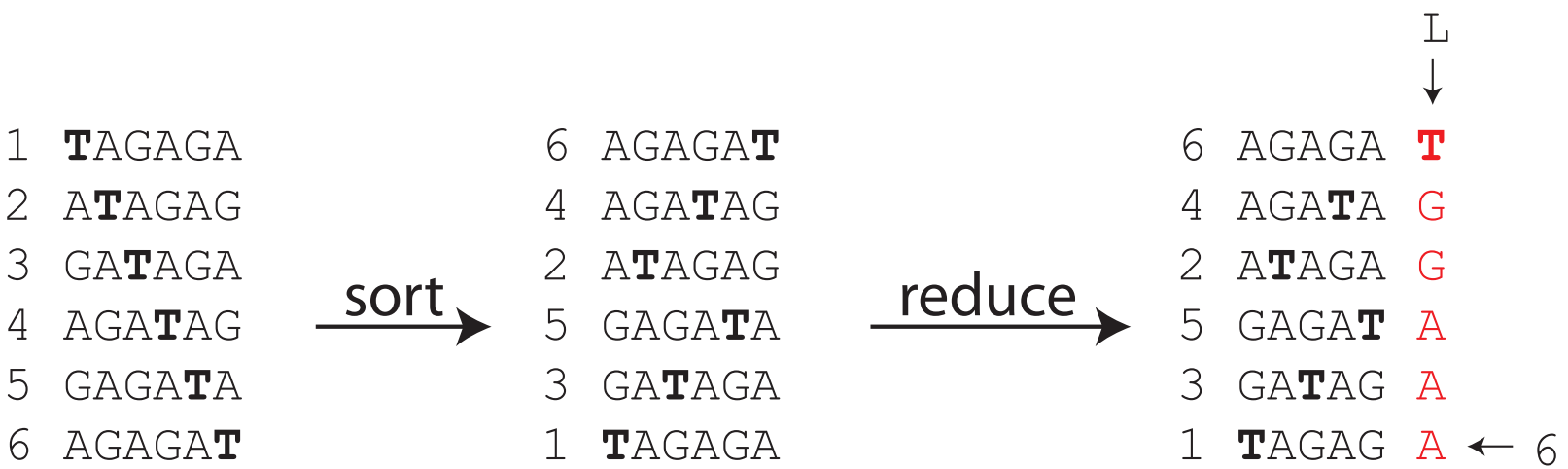
1	**T**AGAGA		6 AGAGA**T**
2	A**T**AGAG		4 AGA**T**AG
3	GA**T**AGA	sort →	2 A**T**AGAG
4	AGA**T**AG		5 GAGA**T**A
5	GAGA**T**A		3 GA**T**AGA
6	AGAGA**T**		1 **T**AGAGA

[Burrows and Wheeler, 1994]

Burrows-Wheeler Transformation (BWT)

How does it work?

- Start with the full genome string (TAGAGA)
- Generate all rotations of the genome string
- Sort the matrix lexicographically
- Keep only the last column L and the index of the original string



[Burrows and Wheeler, 1994]

Burrows-Wheeler Transformation (BWT)

Useful Properties

- String L shows long stretches of the same symbol
- Allows for very efficient compression (compressed genomes)
- String L and index of original string are sufficient to reconstruct the original string TAGAGA!
- After sorting, indices are equivalent to suffix array

6 AGAGA**T**
4 AGA**T**AG
2 A**T**AGAG
5 GAGA**T**A
3 GA**T**AGA
1 **T**AGAGA

[Burrows and Wheeler, 1994]

Burrows-Wheeler Transformation (BWT)

Useful Properties

- String L shows long stretches of the same symbol
- Allows for very efficient compression (compressed genomes)
- String L and index of original string are sufficient to reconstruct the original string TAGAGA!
- After sorting, indices are equivalent to suffix array

6	AGAGA**T**
4	AGA**T**AG
2	A**T**AGAG
5	GAGA**T**A
3	GA**T**AGA
1	**T**AGAGA

[Burrows and Wheeler, 1994]

Full-Text Minute-Space (FM) Index

Motivation

- Combination of the Suffix Array for efficient search and Burrows-Wheeler Transform for genome compression
- Allows for efficient search in a compressed genome representation
- Proposed by Ferragina and Manzini

[Ferragina and Manzini, 2001]

Full-Text Minute-Space (FM) Index

Construction

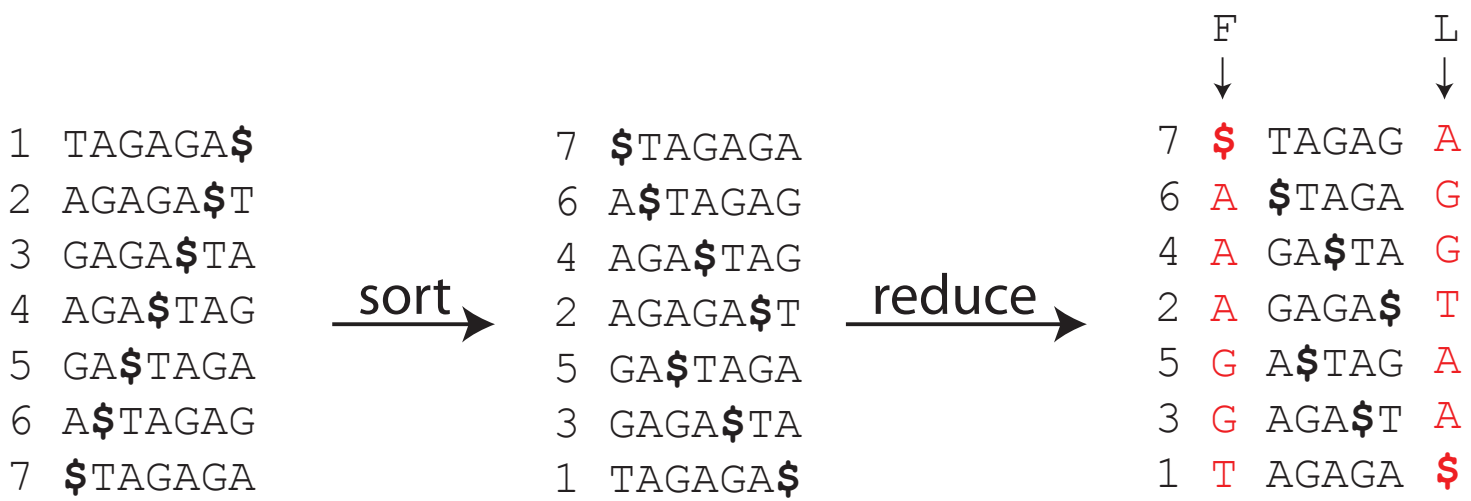
- Add $-symbol to the genome end ($ is lexicographically smaller than any other symbol, so no suffix can be the prefix of another suffix)
- Apply Burrows-Wheeler Transformation to string (compressible genome L)

TAGAGA**$**

Full-Text Minute-Space (FM) Index

Construction

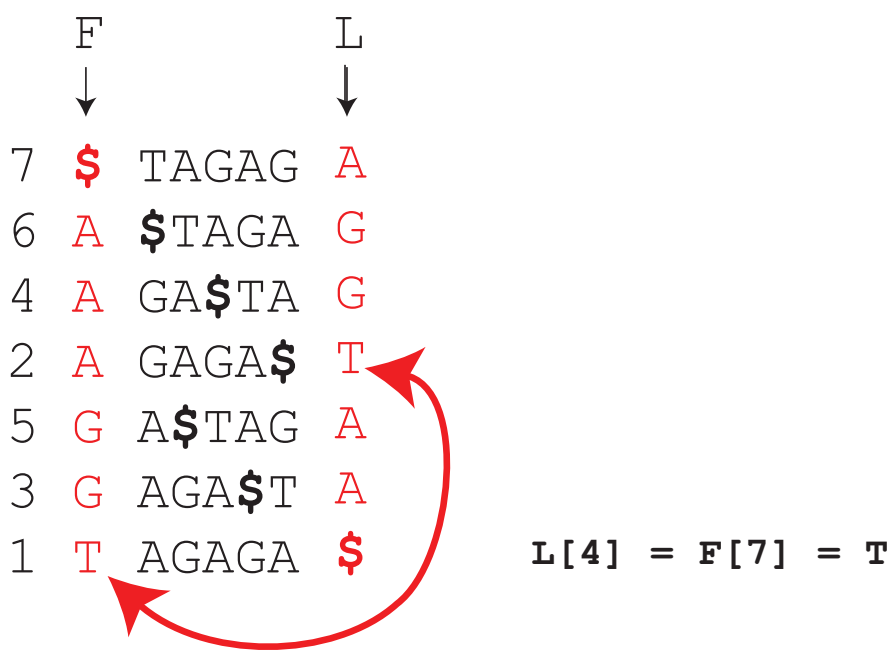
- Add $-symbol to the genome end ($ is lexicographically smaller than any other symbol, so no suffix can be the prefix of another suffix)
- Apply Burrows-Wheeler Transformation to string (compressible genome L)



Full-Text Minute-Space (FM) Index

Last-Columns-to-First-Column Mapping (LF)

- Important property that enables efficient search
- The k-th occurence of character c in L corresponds to the k-th occurence of character c in F



Full-Text Minute-Space (FM) Index

Helpers to compute LF

- C: $C[k]$ = total number of occurrences of the characters $< c$
- $Occ(c, k)$ = number of times c occurs in $L[1, k]$

	F		L
7	**$**	TAGAG	A
6	A	**$**TAGA	G
4	A	GA**$**TA	G
2	A	GAGA**$**	T
5	G	A**$**TAG	A
3	G	AGA**$**T	A
1	T	AGAGA	**$**

Array C

$	A	G	T
0	1	4	6

Full-Text Minute-Space (FM) Index

Helpers to compute LF

- C: $C[k]$ = total number of occurrences of the characters $< c$
- $Occ(c, k)$ = number of times c occurs in $L[1, k]$

	F		L
7	$	TAGAG	A
6	A	$TAGA	G
4	A	GA$TA	G
2	A	GAGA$	T
5	G	A$TAG	A
3	G	AGA$T	A
1	T	AGAGA	$

Array C

$	A	G	T
0	1	4	6

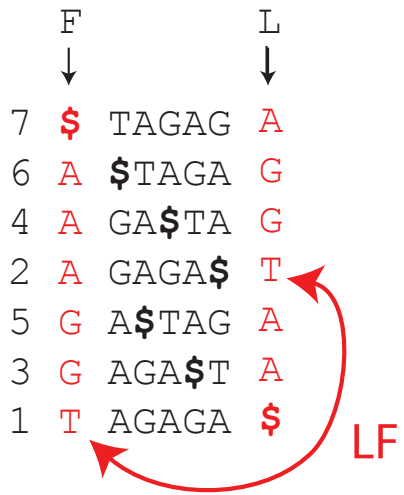
Matrix Occ

	1	2	3	4	5	6	7
$	0	0	0	0	0	0	1
A	1	1	1	1	2	3	3
G	0	1	2	2	2	2	2
T	0	0	0	1	1	1	1

Full-Text Minute-Space (FM) Index

Compute Last-to-First Mapping LF

- Provable correspondence between last col. L and first col. F
- $LF(i) = C[L(i)] + Occ(L[i], i])$



Array C

$	A	G	T
0	1	4	6

Matrix Occ

	1	2	3	4	5	6	7
$	0	0	0	0	0	0	1
A	1	1	1	1	2	3	3
G	0	1	2	2	2	2	2
T	0	0	0	1	1	1	1

```
LF(4) = C[L[4]] + Occ(L[4], 4)
LF(4) = C[T] + Occ(T, 4)
LF(4) = 6 + 1
LF(4) = 7
```

[Ferragina and Manzini, 2001]

FM Index Demo

http://blog.avadis-ngs.com/2012/04/elegant-exact-string-match-using-bwt-2/

Full-Text Minute-Space (FM) Index*

Locate Pattern Range:

$i \leftarrow p, \quad c \leftarrow P[p]$
$sp \leftarrow C[c] + 1$
$ep \leftarrow C[c+1]$
while $(sp \leq ep)$ **and** $(i \geq 2)$ **do**
    $c \leftarrow P[i-1]$
    $sp \leftarrow C[c] + Occ(c, sp-1) + 1$
    $ep \leftarrow C[c] + Occ(c, ep)$
    $i \leftarrow i - 1$
end while

Full-Text Minute-Space (FM) Index*

Locate Pattern Range:

$i \leftarrow p, \quad c \leftarrow P[p]$
$sp \leftarrow C[c] + 1$
$ep \leftarrow C[c+1]$
while $(sp \leq ep)$ **and** $(i \geq 2)$ **do**
  $c \leftarrow P[i-1]$
  $sp \leftarrow C[c] + Occ(c, sp-1) + 1$
  $ep \leftarrow C[c] + Occ(c, ep)$
  $i \leftarrow i - 1$
end while

Array C

$	A	G	T
0	1	4	6

Matrix Occ

	1	2	3	4	5	6	7
$	0	0	0	0	0	0	1
A	1	1	1	1	2	3	3
G	0	1	2	2	2	2	2
T	0	0	0	1	1	1	1

Example

$P = \mathbf{AGA}$, $p = |P| = 3$

Init:

$c \leftarrow P[3] = \mathbf{A}$
$sp \leftarrow C[\mathbf{A}] + 1 = 2$
$ep \leftarrow C[\mathbf{A}+1] = C[\mathbf{G}] = 4$

Iteration 1:

$c \leftarrow \mathbf{G}$
$sp \leftarrow C[\mathbf{G}] + Occ[\mathbf{G}, 1] + 1 = 5$
$ep \leftarrow C[\mathbf{G}] + Occ[\mathbf{G}, 4] = 6$

Iteration 2:

$c \leftarrow \mathbf{A}$
$sp \leftarrow C[\mathbf{A}] + Occ[\mathbf{A}, 5] + 1 = 3$
$ep \leftarrow C[\mathbf{A}] + Occ[\mathbf{A}, 6] = 4$

Full-Text Minute-Space (FM) Index*

Locate Pattern Range:

$i \leftarrow p, \quad c \leftarrow P[p]$
$sp \leftarrow C[c] + 1$
$ep \leftarrow C[c+1]$
while $(sp \leq ep)$ **and** $(i \geq 2)$ **do**
  $c \leftarrow P[i-1]$
  $sp \leftarrow C[c] + Occ(c, sp-1) + 1$
  $ep \leftarrow C[c] + Occ(c, ep)$
  $i \leftarrow i - 1$
end while

Array C

$	A	G	T
0	1	4	6

Matrix Occ

	1	2	3	4	5	6	7
$	0	0	0	0	0	0	1
A	1	1	1	1	2	3	3
G	0	1	2	2	2	2	2
T	0	0	0	1	1	1	1

Example

$P = \mathbf{AGA}$, $p = |P| = 3$

Init:
  $c \leftarrow P[3] = \mathbf{A}$
  $sp \leftarrow C[\mathbf{A}] + 1 = 2$
  $ep \leftarrow C[\mathbf{A}{+}1] = C[\mathbf{G}] = 4$

Iteration 1:
  $c \leftarrow \mathbf{G}$
  $sp \leftarrow C[\mathbf{G}] + Occ[\mathbf{G}, 1] + 1 = 5$
  $ep \leftarrow C[\,\mathbf{G}] + Occ[\mathbf{G}, 4] = 6$

Iteration 2:
  $c \leftarrow \mathbf{A}$
  $sp \leftarrow C[\mathbf{A}] + Occ[\mathbf{A}, 5] + 1 = 3$
  $ep \leftarrow C[\,\mathbf{A}] + Occ[\mathbf{A}, 6] = 4$

Full-Text Minute-Space (FM) Index*

Locate Pattern Range:

$i \leftarrow p, \quad c \leftarrow P[p]$
$sp \leftarrow C[c] + 1$
$ep \leftarrow C[c+1]$
while $(sp \leq ep)$ **and** $(i \geq 2)$ **do**
  $c \leftarrow P[i-1]$
  $sp \leftarrow C[c] + Occ(c, sp-1) + 1$
  $ep \leftarrow C[c] + Occ(c, ep)$
  $i \leftarrow i - 1$
end while

Array C

$	A	G	T
0	1	4	6

Matrix Occ

	1	2	3	4	5	6	7
$	0	0	0	0	0	0	1
A	1	1	1	1	2	3	3
G	0	1	2	2	2	2	2
T	0	0	0	1	1	1	1

Example

$P = \mathbf{AGA}$, $p = |P| = 3$

Init:
  $c \leftarrow P[3] = \mathbf{A}$
  $sp \leftarrow C[\mathbf{A}] + 1 = 2$
  $ep \leftarrow C[\mathbf{A}{+}1] = C[\mathbf{G}] = 4$

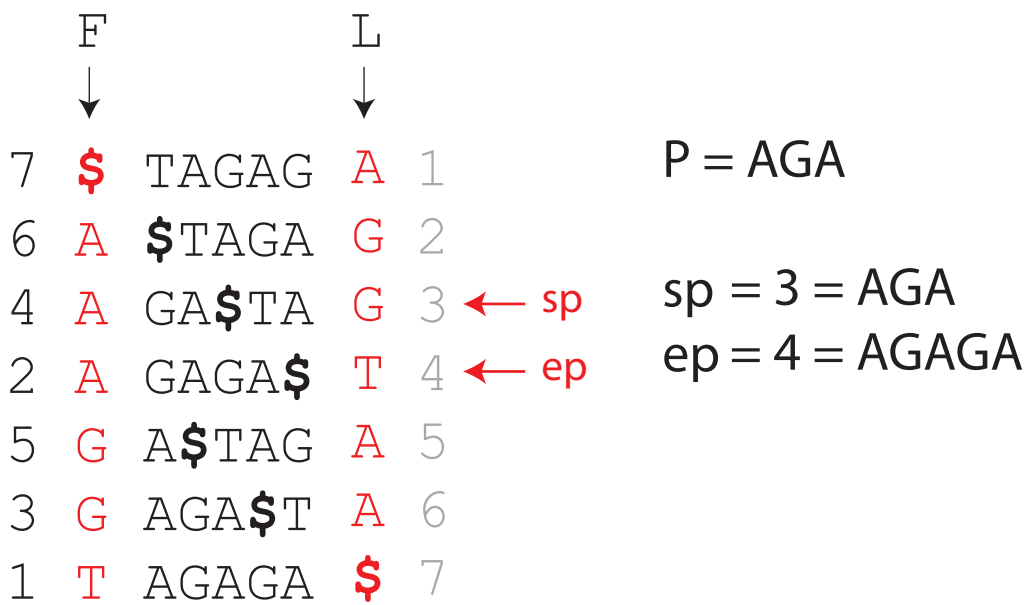
Iteration 1:
  $c \leftarrow \mathbf{G}$
  $sp \leftarrow C[\mathbf{G}] + Occ[\mathbf{G}, 1] + 1 = 5$
  $ep \leftarrow C[\,\mathbf{G}] + Occ[\mathbf{G}, 4] = 6$

Iteration 2:
  $c \leftarrow \mathbf{A}$
  $sp \leftarrow C[\mathbf{A}] + Occ[\mathbf{A}, 5] + 1 = 3$
  $ep \leftarrow C[\,\mathbf{A}] + Occ[\mathbf{A}, 6] = 4$

Full-Text Minute-Space (FM) Index*

LF-Mapping for Search

- Search algorithm provides hit-range for pattern P in BWT-Matrix index
- Translation into original Coordinates for S necessary

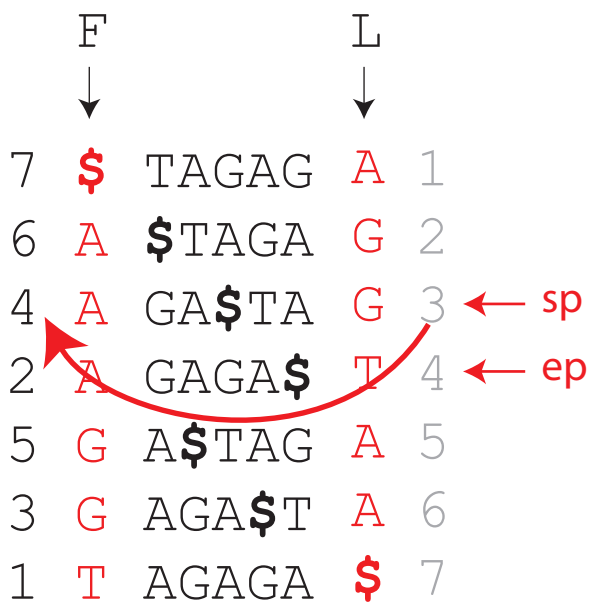


[Ferragina and Manzini, 2001]

Full-Text Minute-Space (FM) Index*

LF-Mapping for Search

- Search algorithm provides hit-range for pattern P in BWT-Matrix index
- Translation into original Coordinates for S necessary



P = AGA

sp = 3 = AGA
ep = 4 = AGAGA

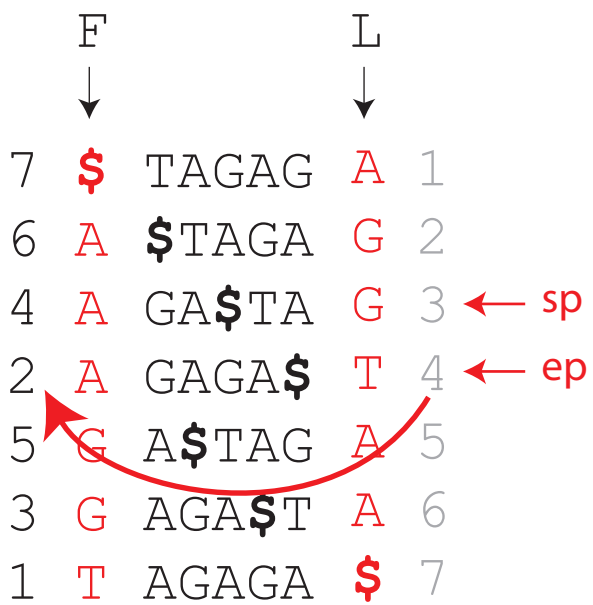
```
   123456
S = TAGAGA
      AGA
```

[Ferragina and Manzini, 2001]

Full-Text Minute-Space (FM) Index*

LF-Mapping for Search

- Search algorithm provides hit-range for pattern P in BWT-Matrix index
- Translation into original Coordinates for S necessary



P = AGA

sp = 3 = AGA
ep = 4 = AGAGA

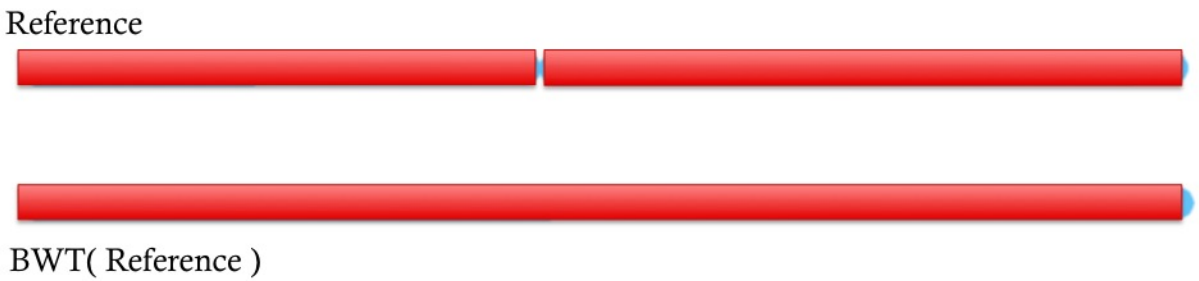
123456
S = TAGAGA
AGAGA

[Ferragina and Manzini, 2001]

Alignments with Bowtie

Details

- Bowtie uses FM-Index on a compressed BWT of the genome
- Search read pattern P in genome $S \rightarrow$ list of start positions
- Longer patterns decrease the number of hit positions



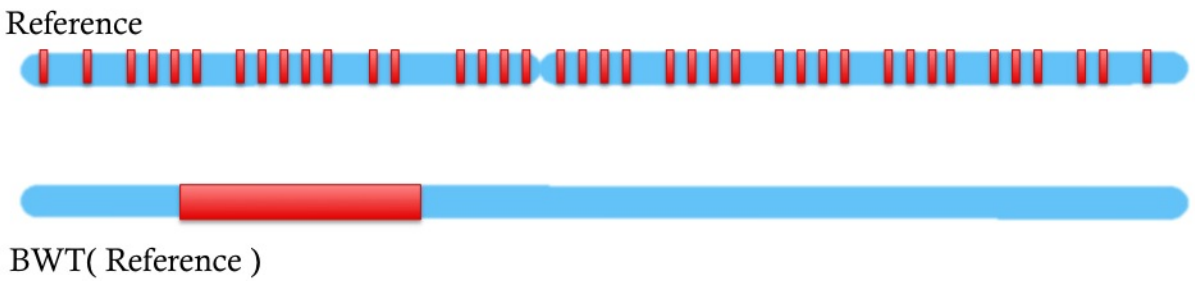
@HWI-EAS412_4:1:1:1376:380
AATGATACGGCGACCACCGAGATCTA

(Slides provided by Cole Trapnell, U Maryland & UC Berkeley)

Alignments with Bowtie

Details

- Bowtie uses FM-Index on a compressed BWT of the genome
- Search read pattern P in genome $S \rightarrow$ list of start positions
- Longer patterns decrease the number of hit positions



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

(Slides provided by Cole Trapnell, U Maryland & UC Berkeley)

Alignments with Bowtie

Details

- Bowtie uses FM-Index on a compressed BWT of the genome
- Search read pattern P in genome $S \rightarrow$ list of start positions
- Longer patterns decrease the number of hit positions



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

(Slides provided by Cole Trapnell, U Maryland & UC Berkeley)

Mapping Tools

Common Tools

- bowtie [Langmead et al., 2009]
- bwa [Li and Durbin, 2009, 2010]
- GenomeMapper [Schneeberger et al., 2009a]
- *Shrimp* [Rumble et al., 2009]
- *SOAP(2)* [Li et al., 2009]
- *VMATCH*
- MAQ [Li et al., 2008]
- ELAND
- segemehl [Hoffmann et al., 2009]
- . . . (≈50 more)

Main Issues

- Accuracy
- Speed
- Memory Consumption

Mapping Tools

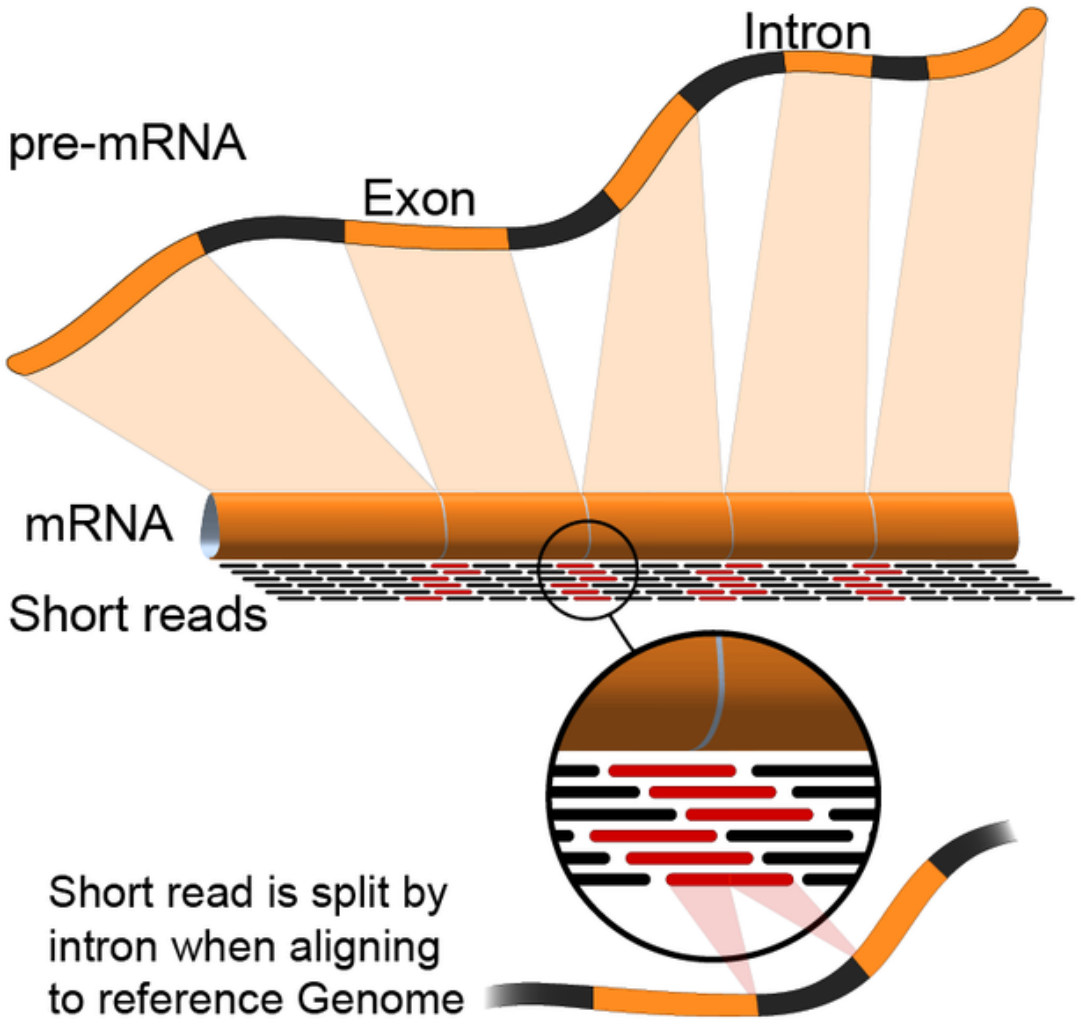
Common Tools

- bowtie [Langmead et al., 2009]
- bwa [Li and Durbin, 2009, 2010]
- GenomeMapper [Schneeberger et al., 2009a]
- *Shrimp* [Rumble et al., 2009]
- *SOAP(2)* [Li et al., 2009]
- *VMATCH*
- MAQ [Li et al., 2008]
- ELAND
- segemehl [Hoffmann et al., 2009]
- . . . (≈50 more)

Main Issues

- Accuracy
- Speed
- Memory Consumption

Spliced vs. Unspliced Alignments

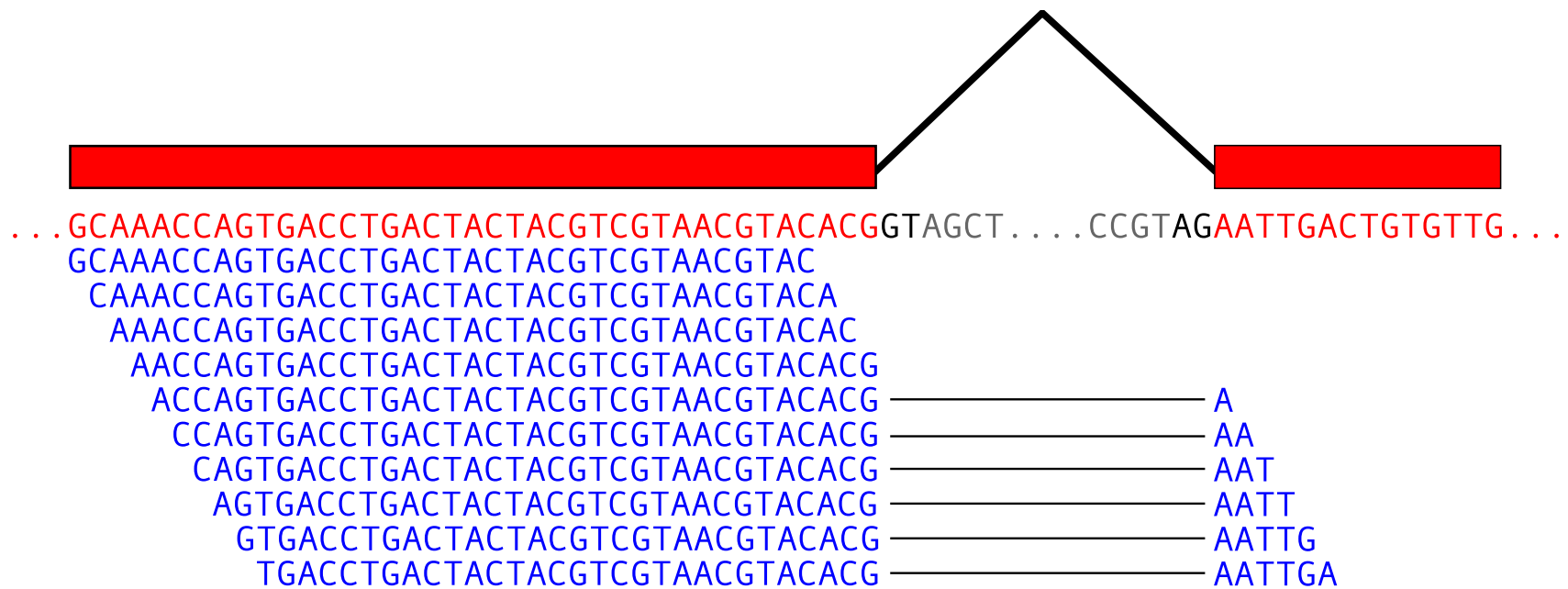


[Wikipedia]

Spliced vs. Unspliced Alignments

Details

- Find split-read mapping in genome with a few mismatches
- Efficient data structures for mapping many reads
- Initially, most mapping techniques were limited to unspliced reads



Tools for Spliced Read Alignments

Spliced Alignment of cDNA

- *BLAST* [Altschul et al., 1990]
- *spliced alignments* [Gelfand et al., 1996]
- *SIM4* [Florea et al., 1998]
- *GeneSeqer* [Usuka et al., 2000]
- *Spidey* [Wheelan SJ, 2001]
- *BLAT* [Kent, 2002]
- *EXALIN* [Zhang and Gish, 2006]
- *PALMA* [Schulze et al., 2007]

⇒Too slow for RNA-seq read alignment

Tools for Spliced Read Alignments

Spliced Alignment of NGS Reads

- Erange [Mortazavi et al., 2008]
- GEM [Marco-Sola et al., 2012]
- MapNext [Bao et al., 2009]
- MapSplice [Wang et al., 2010]
- **PALMapper** [De Bona et al., 2008, Rätsch et al., 2010]
- PASS [Campagna et al., 2009]
- STAR [Dobin et al., 2013]
- **TopHat** [Trapnell et al., 2009], TopHat2 [Kim et al., 2013]
- . . .

Main Issues

- Assumptions on splice consensus
- Accuracy of intron predictions
- Speed (often higher than for unspliced alignments)
- Memory consumption (similar to unspliced mappers)

Tools for Spliced Read Alignments

Spliced Alignment of NGS Reads

- Erange [Mortazavi et al., 2008]
- GEM [Marco-Sola et al., 2012]
- MapNext [Bao et al., 2009]
- MapSplice [Wang et al., 2010]
- **PALMapper** [De Bona et al., 2008, Rätsch et al., 2010]
- PASS [Campagna et al., 2009]
- STAR [Dobin et al., 2013]
- **TopHat** [Trapnell et al., 2009], TopHat2 [Kim et al., 2013]
- . . .

Main Issues

- Assumptions on splice consensus
- Accuracy of intron predictions
- Speed (often higher than for unspliced alignments)
- Memory consumption (similar to unspliced mappers)

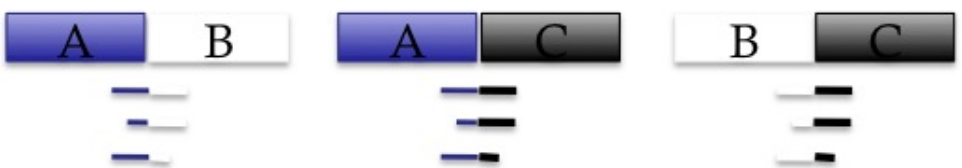
Aligning reads with TopHat



identify candidate exons via genomic mapping

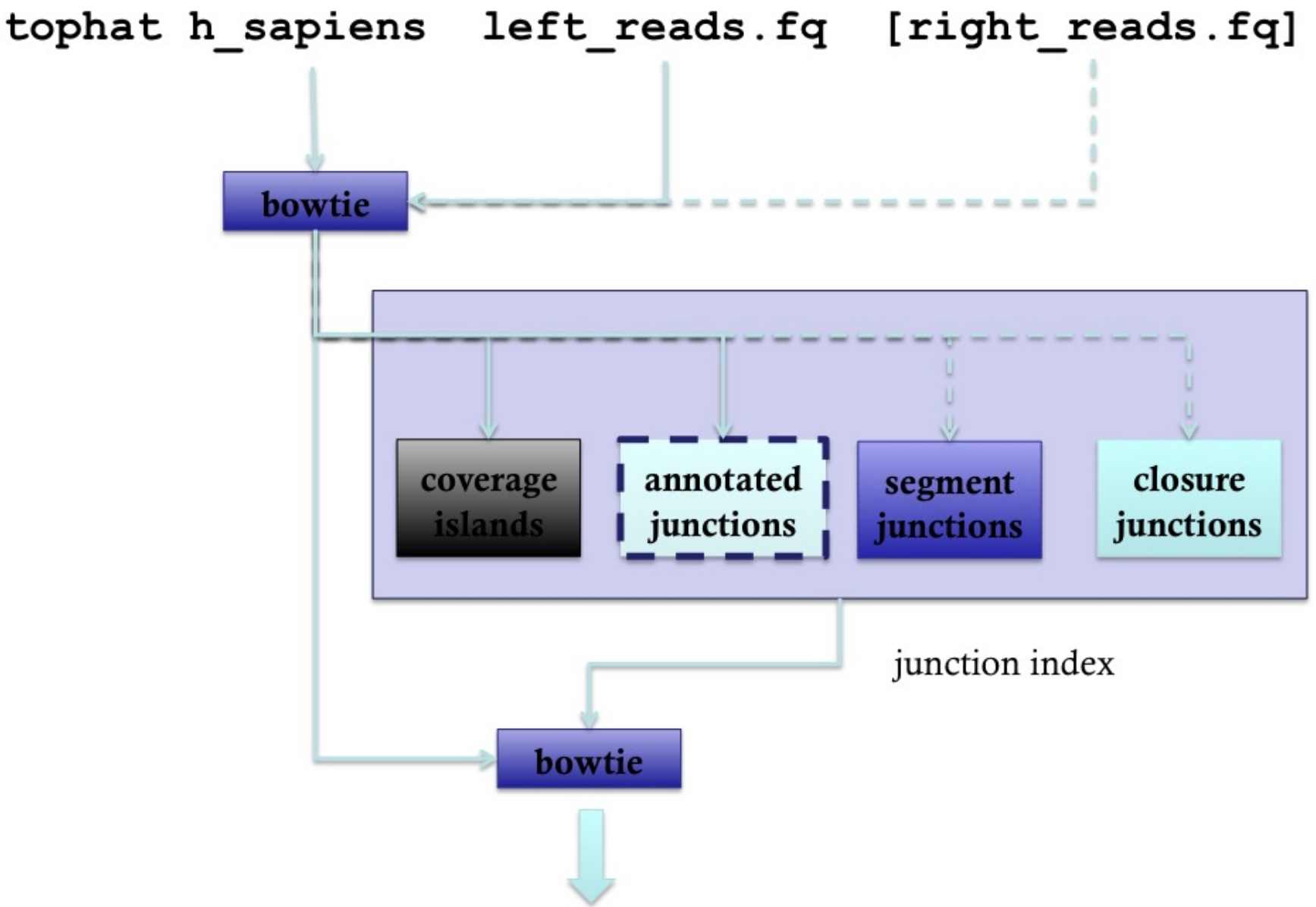


Generate possible pairings of exons



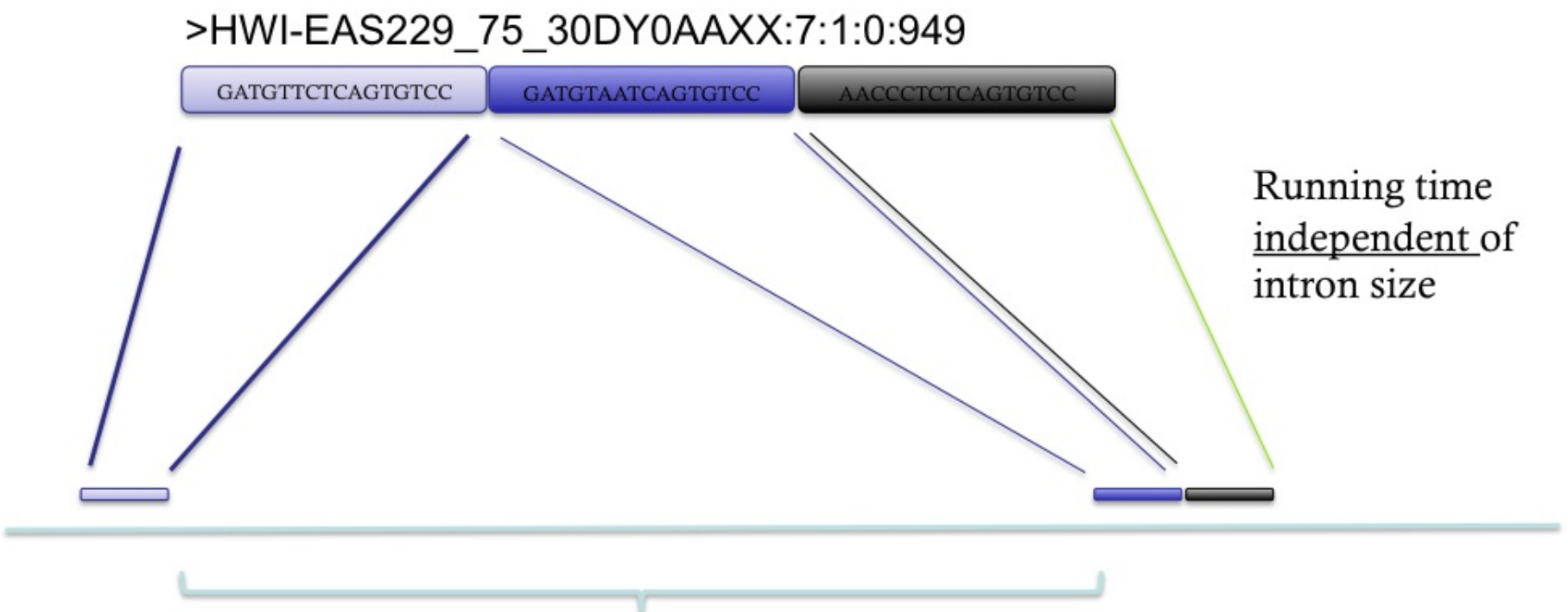
Align reads to possible junctions

(Trapnell, 2010)


tophat h_sapiens left_reads.fq [right_reads.fq]
bowtie
coverage islands
annotated junctions
segment junctions
closure junctions
junction index
bowtie
Spliced alignments
(Trapnell, 2010)

Longer reads

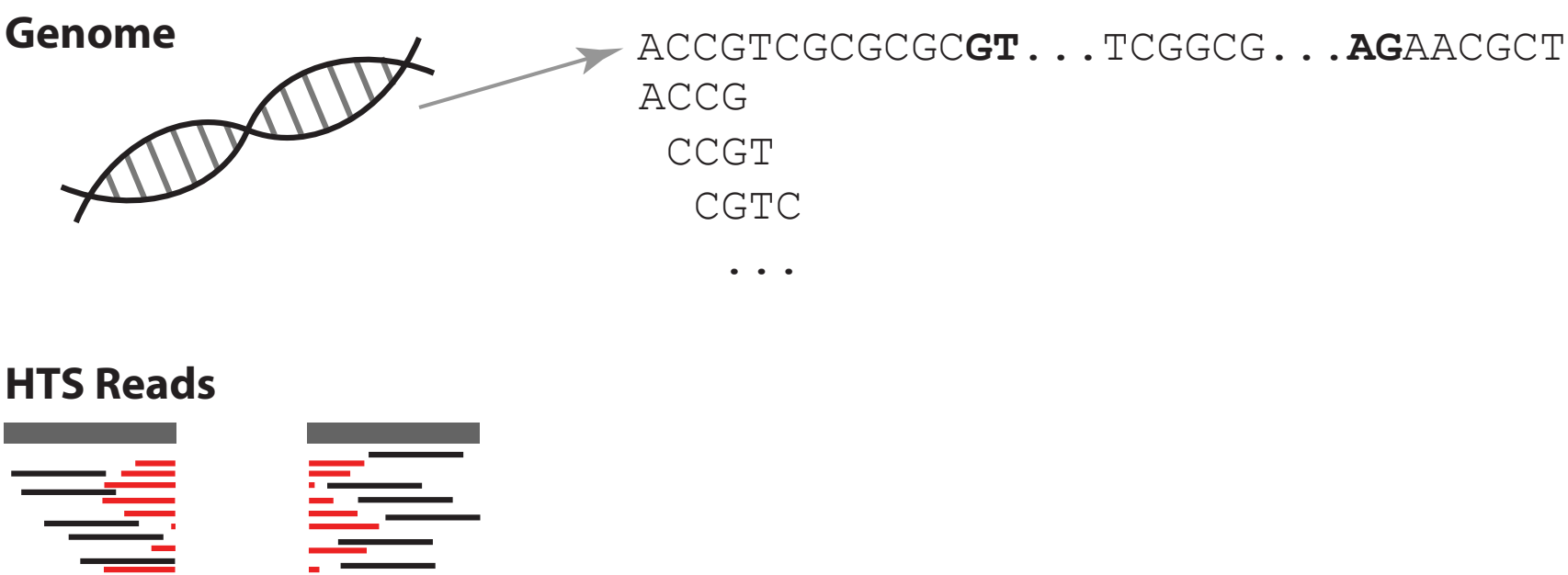
By segmenting the long reads, and mapping the segments independently, we can look harder for junctions we might have missed with shorter reads



RNA-seq Read Alignments with PALMapper

Index and Seeding

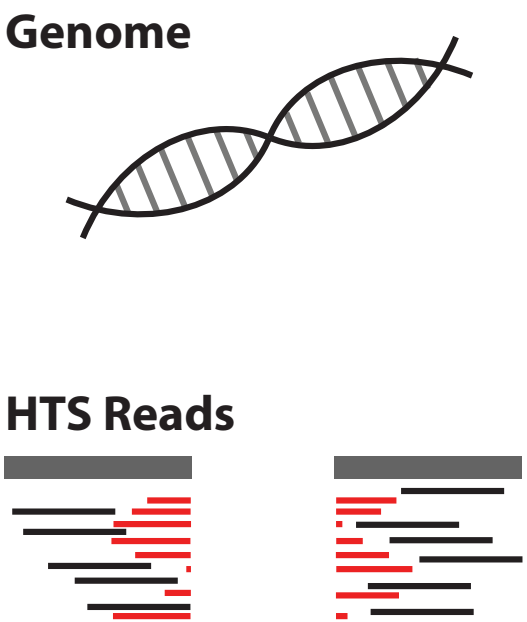
- Indexing genome with *k*-mer based index or bwt-based index
- Lookup read *k*-mers in the genome index
- Generate long and short seed regions for further alignment

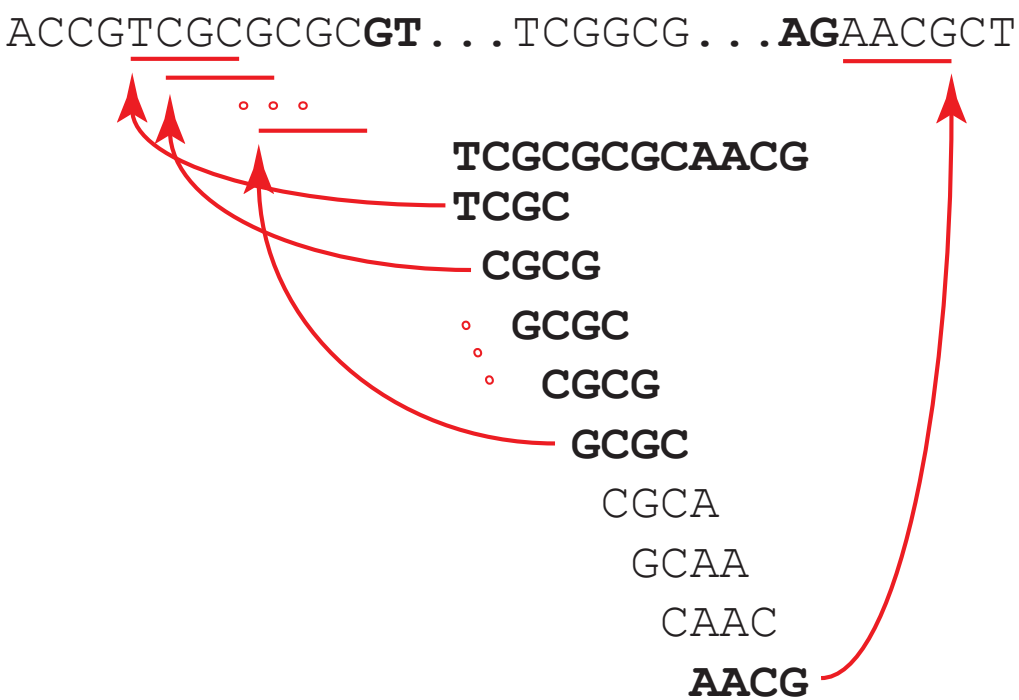


RNA-seq Read Alignments with PALMapper

Index and Seeding

- Indexing genome with *k*-mer based index or bwt-based index
- Lookup read *k*-mers in the genome index
- Generate long and short seed regions for further alignment



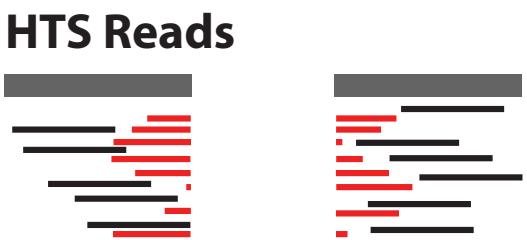


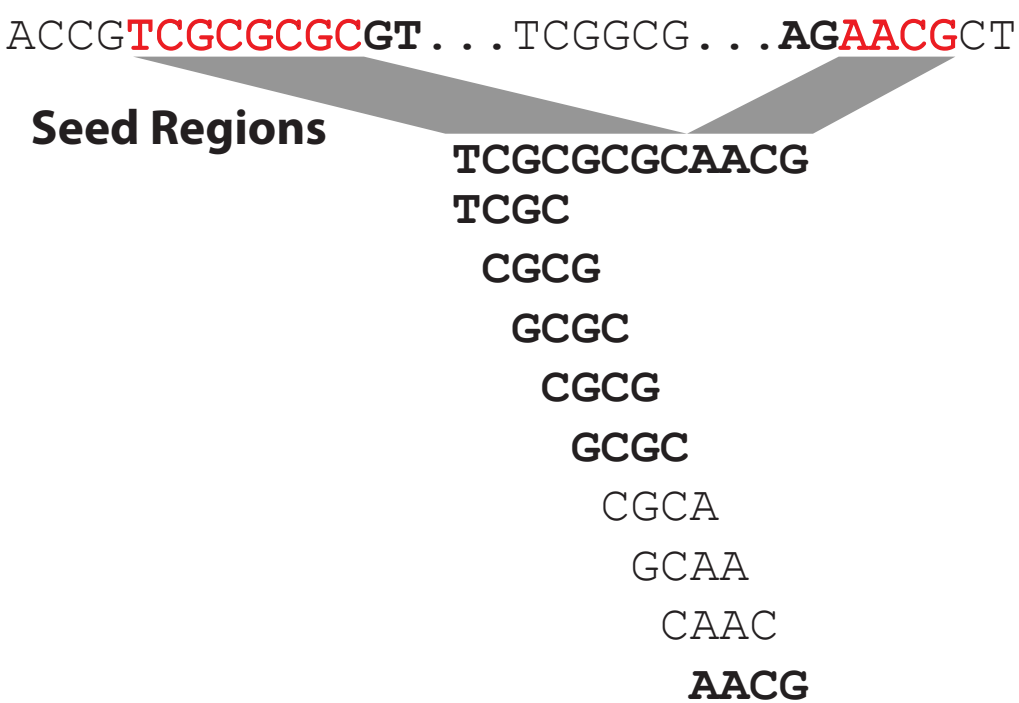
RNA-seq Read Alignments with PALMapper

Index and Seeding

- Indexing genome with *k*-mer based index or bwt-based index
- Lookup read *k*-mers in the genome index
- Generate long and short seed regions for further alignment



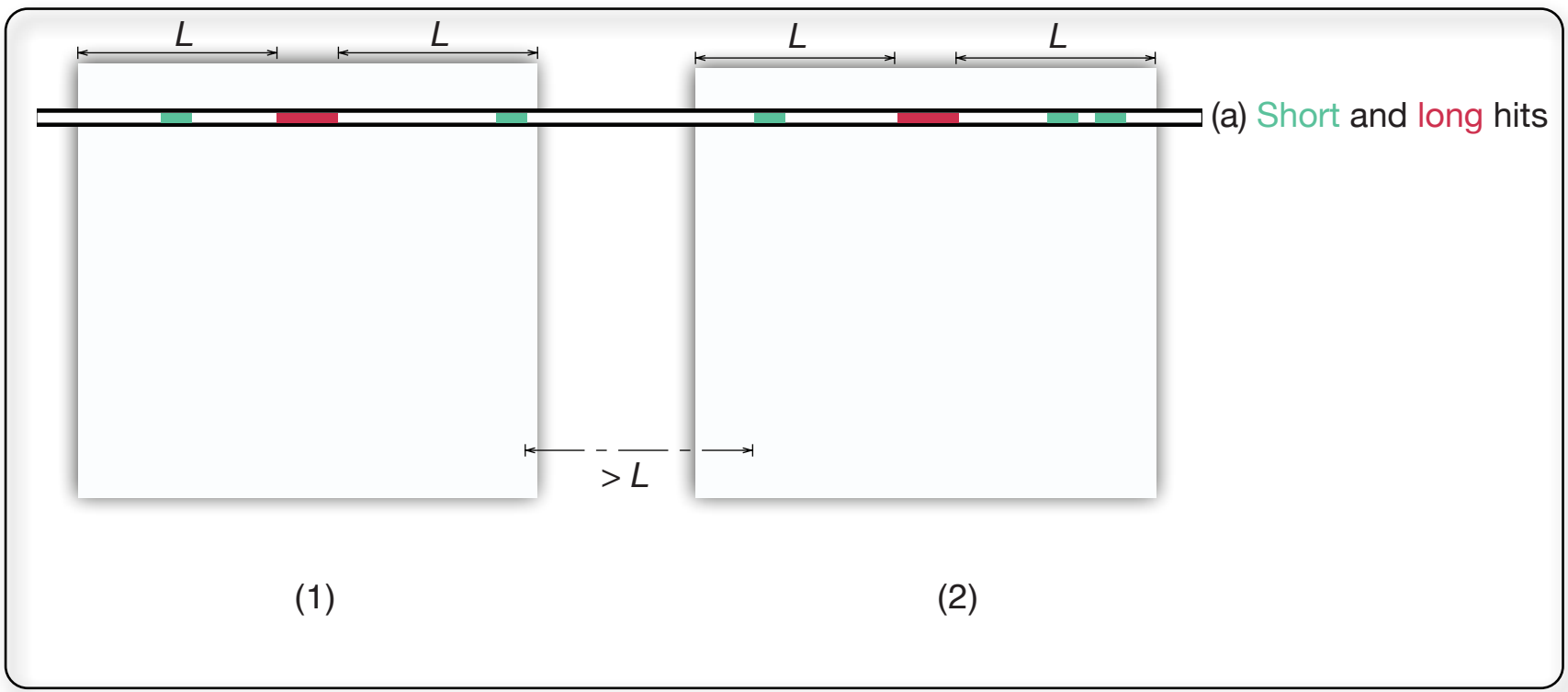




Genomic Sequence Reduction in PALMapper

Projection to Pseudo Chromosomes

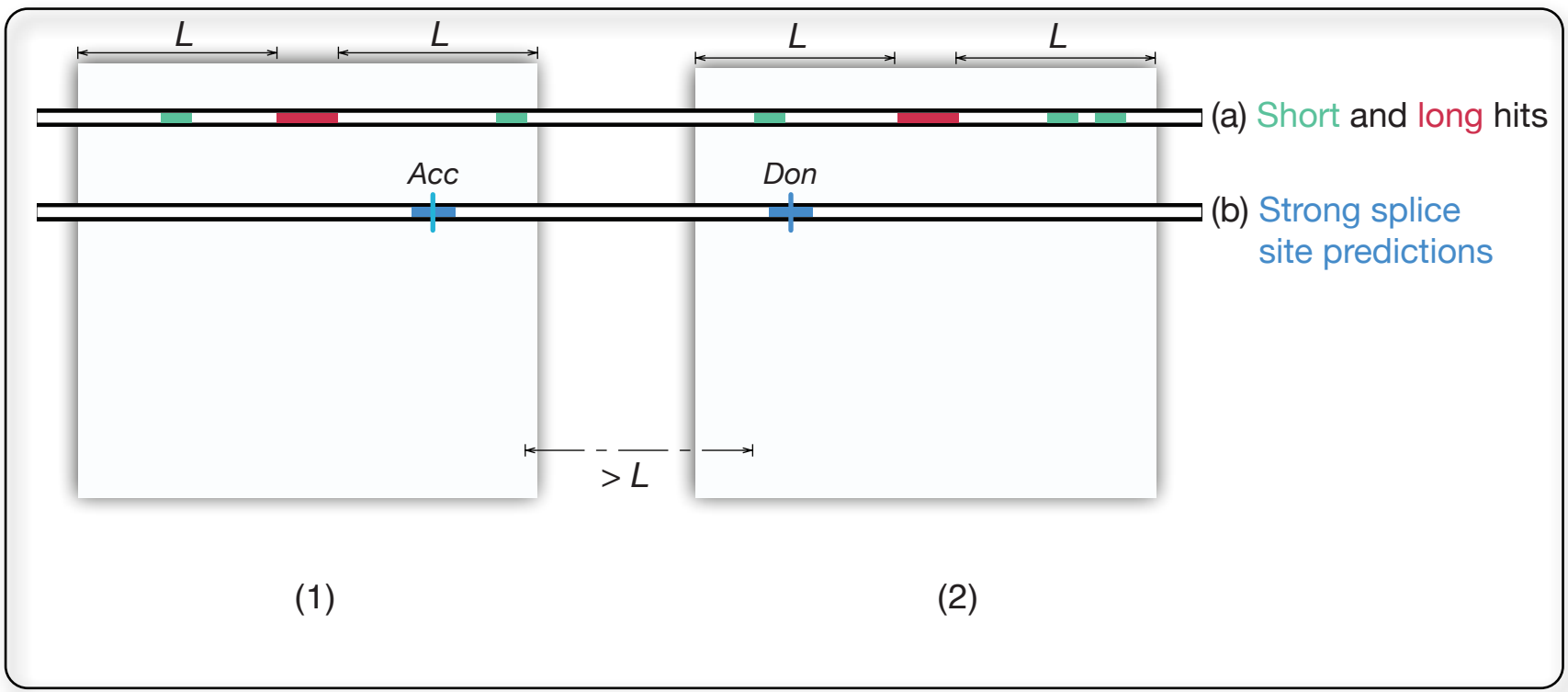
- Rank and combine seed regions from previous step
- Integrate with splice site prediction and coverage information
- Project to genome and form *Pseudo Chromosome Sequences*



Genomic Sequence Reduction in PALMapper

Projection to Pseudo Chromosomes

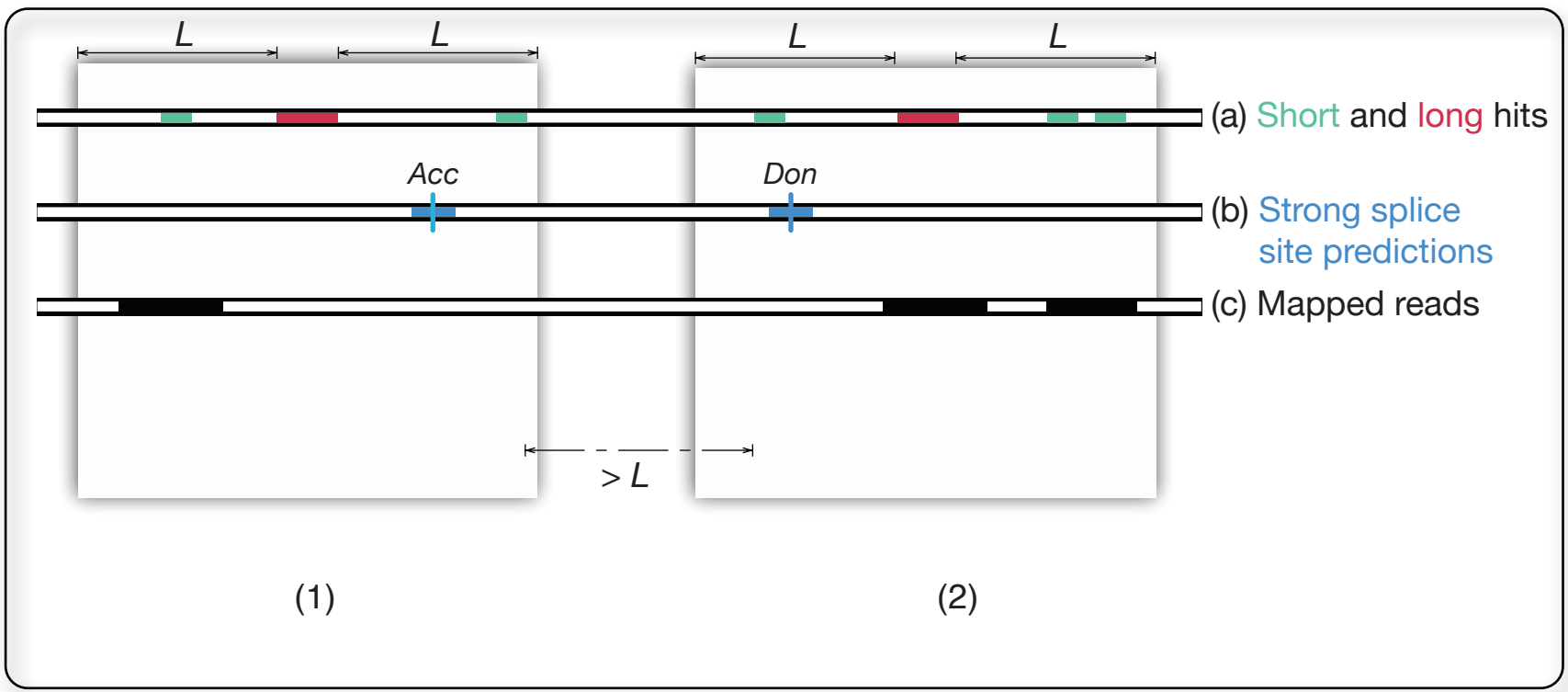
- Rank and combine seed regions from previous step
- Integrate with splice site prediction and coverage information
- Project to genome and form *Pseudo Chromosome Sequences*



Genomic Sequence Reduction in PALMapper

Projection to Pseudo Chromosomes

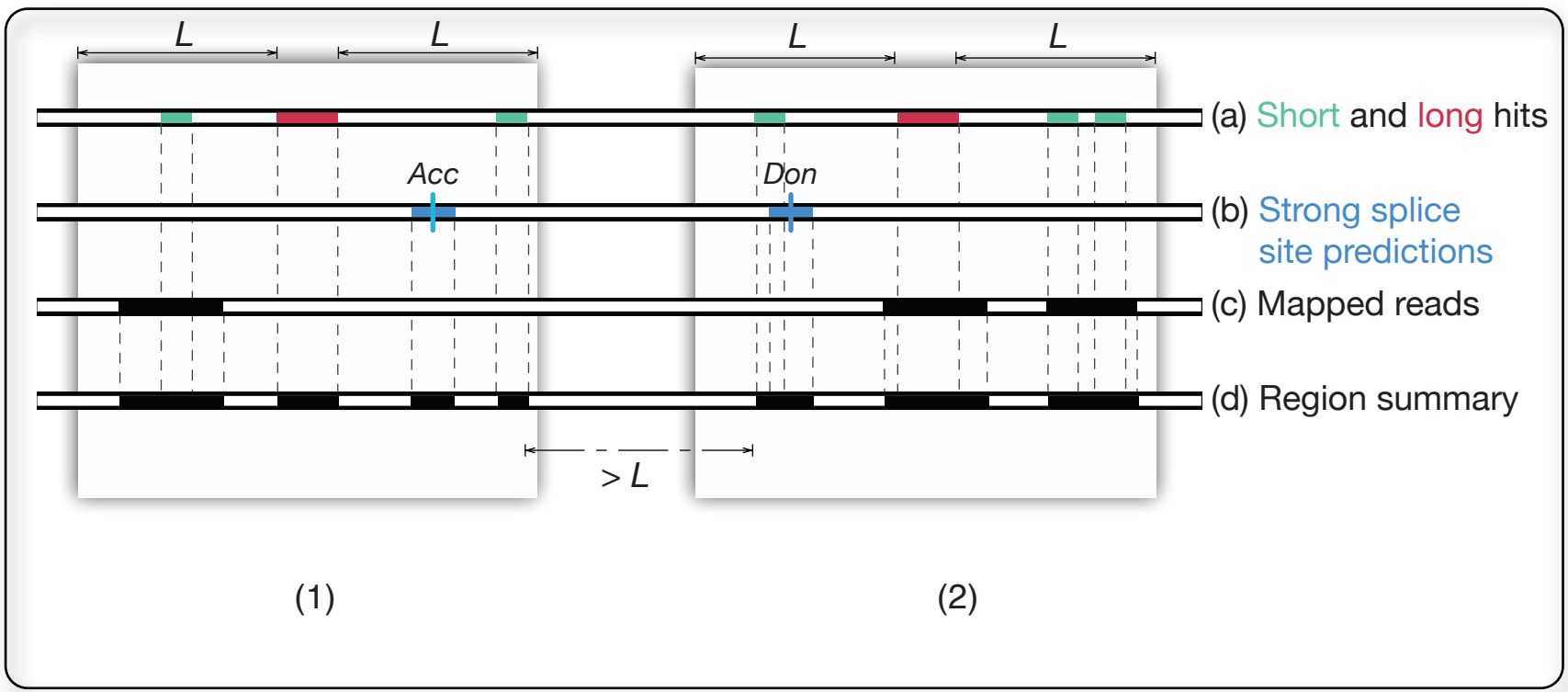
- Rank and combine seed regions from previous step
- Integrate with splice site prediction and coverage information
- Project to genome and form *Pseudo Chromosome Sequences*



Genomic Sequence Reduction in PALMapper

Projection to Pseudo Chromosomes

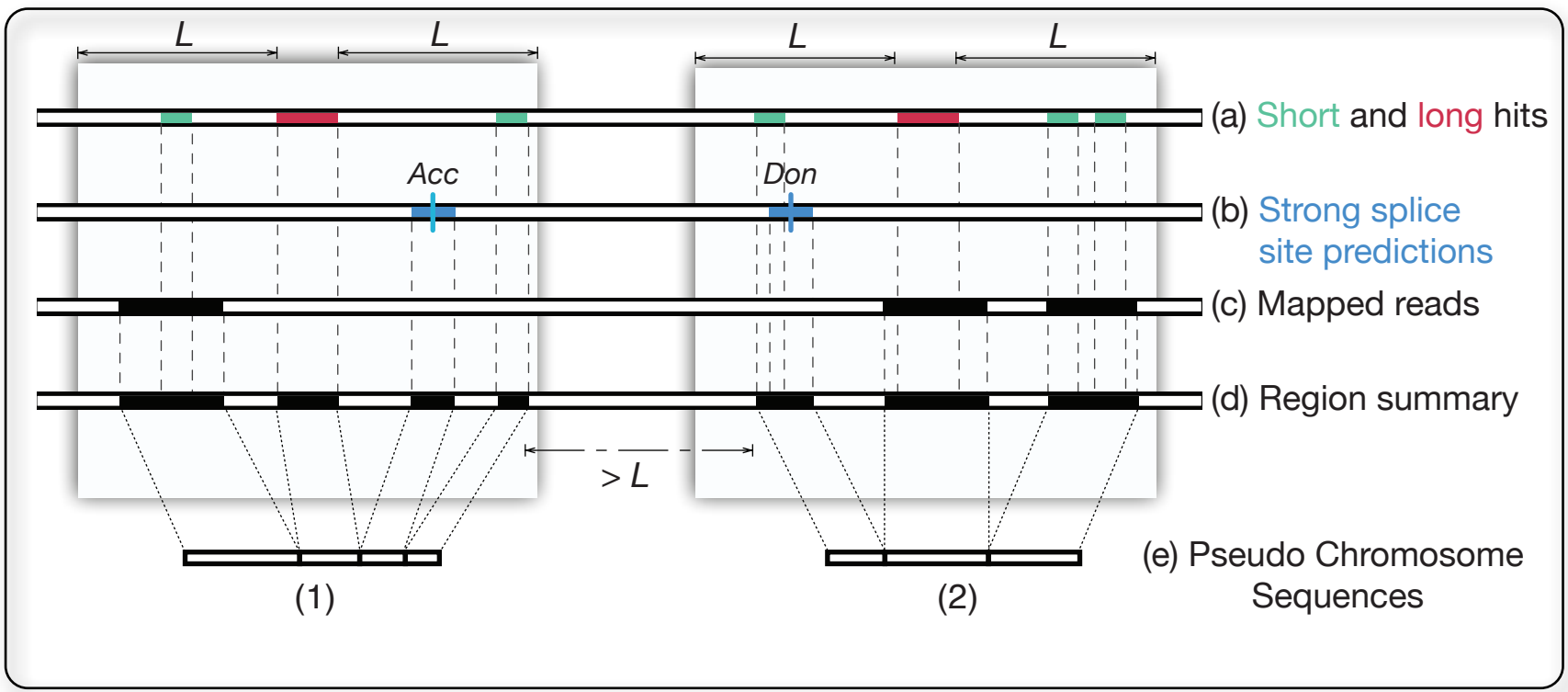
- Rank and combine seed regions from previous step
- Integrate with splice site prediction and coverage information
- Project to genome and form *Pseudo Chromosome Sequences*



Genomic Sequence Reduction in PALMapper

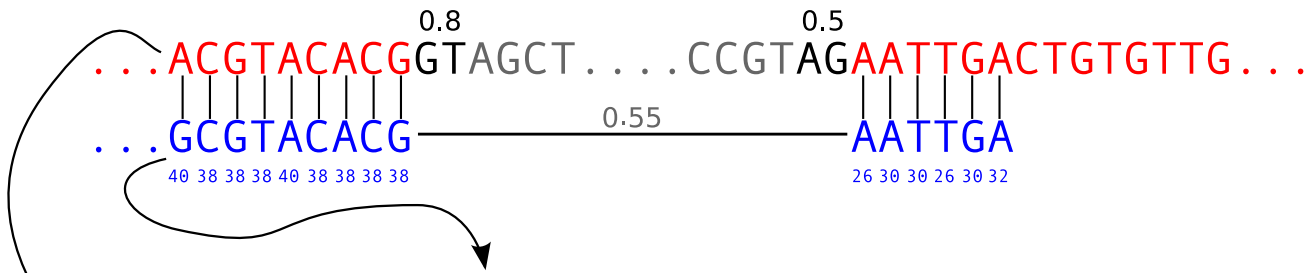
Projection to Pseudo Chromosomes

- Rank and combine seed regions from previous step
- Integrate with splice site prediction and coverage information
- Project to genome and form *Pseudo Chromosome Sequences*



Dynamic Program (DP) in PALMapper

The inferred *seed region* guides a Smith-Waterman alignment:



	gap	A	C	G	T	N
gap	0.33	$f_{-A}(\cdot)$	$f_{-C}(\cdot)$	$f_{-G}(\cdot)$	$f_{-T}(\cdot)$	$f_{-N}(\cdot)$
A	0.31	$f_{AA}(\cdot)$	$f_{AC}(\cdot)$	$f_{AG}(\cdot)$	$f_{AT}(\cdot)$	$f_{AN}(\cdot)$
C	0.44	$f_{CA}(\cdot)$	$f_{CC}(\cdot)$	$f_{CG}(\cdot)$	$f_{CT}(\cdot)$	$f_{CN}(\cdot)$
G	0.13	$f_{GA}(\cdot)$	$f_{GC}(\cdot)$	$f_{GG}(\cdot)$	$f_{GT}(\cdot)$	$f_{GN}(\cdot)$
T	0.55	$f_{TA}(\cdot)$	$f_{TC}(\cdot)$	$f_{TG}(\cdot)$	$f_{TT}(\cdot)$	$f_{TN}(\cdot)$
N	0.12	$f_{NA}(\cdot)$	$f_{NC}(\cdot)$	$f_{NG}(\cdot)$	$f_{NT}(\cdot)$	$f_{NN}(\cdot)$

QPALMA Scores are functions of:

- splice site strengths
- intron length
- quality information
- error profile

Semi-Global

Align whole read to part of pseudo chromosome

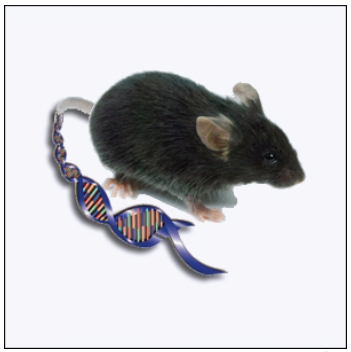
Banded

Limit number of gaps from perfect alignment

Spliced

Allow long gaps corresponding to introns

Need for Variation-aware Alignments



Mouse Genome Project

source: sanger.ac.uk



source: genome.sfu.ca/mmp/

Arabidopsis 1001 Genomes



source: 1001genomes.org



Panzea

source: alpha.maizegdb.org



1000 Genome Project

source: nature.com

Resource Name	Organism	Strains/Individuals	SNPs/SNVs	InDels	Reference
Mouse Genome Project	*M. musculus*	17 strains	129,260,574	21,683,297	Keane (2011)
19 Genomes Project	*A. thaliana*	19 strains	3,070,000	1,200,000	Gan (2011)
Panzea	*T. mays*	103 strains	55,061,920	3,200,000	Chia (2012)
Million Mutation Project	*C. elegans*	2,047 strains	800,000	220,000	Thompson (2013)
1000 Genomes Project	*H. sapians*	1,092 individuals	36,700,000	1,380,000	The 1000 Genomes Project Consortium (2012)

Variation-Aware Alignment with PALMapper

k-mer Indexing (variant case)

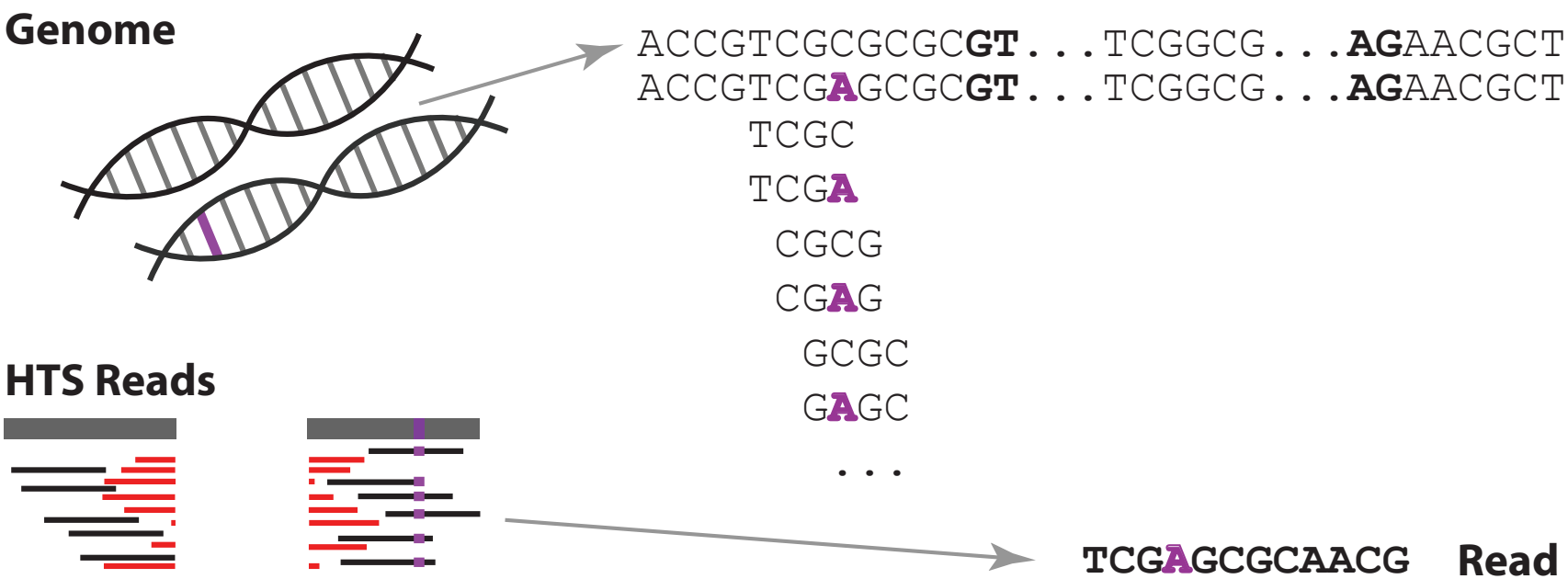
- Include available variation into index (dbSNP, COSMIC, ENCODE, 1000 Genomes . . .)
- Principle for querying the index stays the same



Variation-Aware Alignment with PALMapper

k-mer Indexing (variant case)

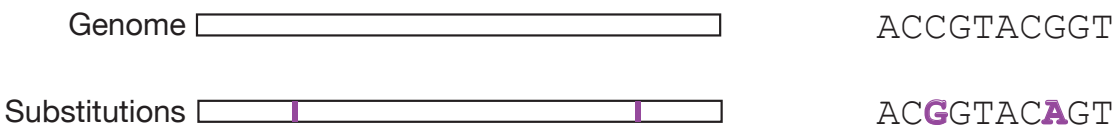
- Include available variation into index (dbSNP, COSMIC, ENCODE, 1000 Genomes . . .)
- **Principle for querying the index stays the same**



Variation-Aware Alignment with PALMapper

Dynamic Program Alignment Against Graph

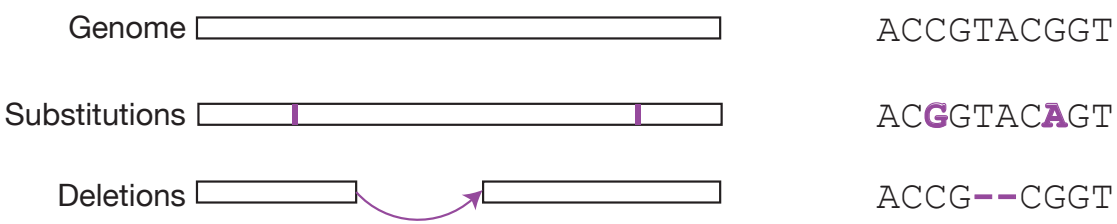
- Integrate substitutions as IUPAC codes for ambiguous bases
- Integrate deletions as jumps
- Integrate insertions as jumps
- Collapse all variations into graph-like structure
- Align against variation graph



Variation-Aware Alignment with PALMapper

Dynamic Program Alignment Against Graph

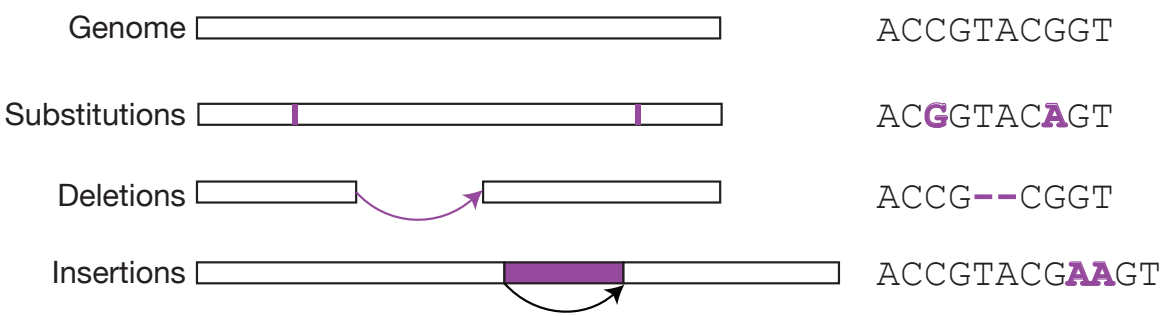
- Integrate substitutions as IUPAC codes for ambiguous bases
- Integrate deletions as jumps
- Integrate insertions as jumps
- Collapse all variations into graph-like structure
- Align against variation graph



Variation-Aware Alignment with PALMapper

Dynamic Program Alignment Against Graph

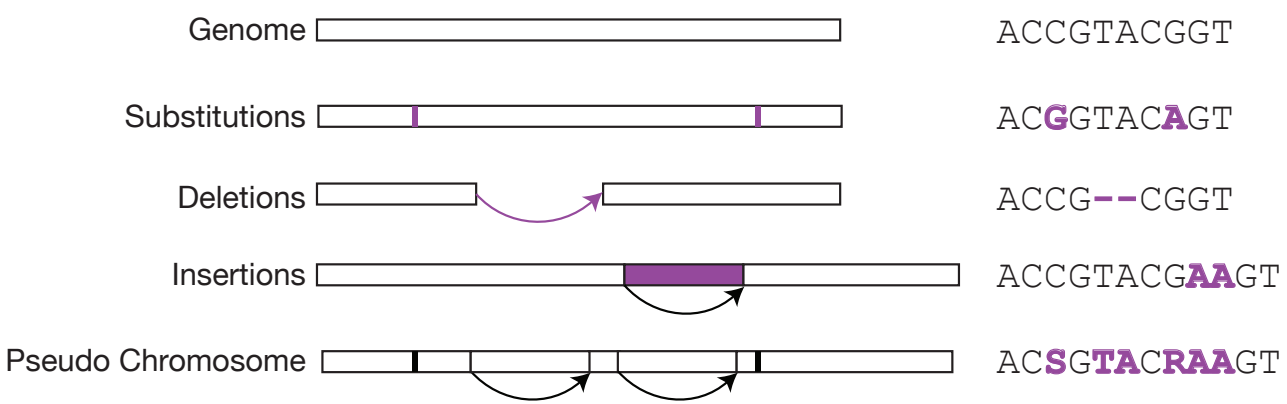
- Integrate substitutions as IUPAC codes for ambiguous bases
- Integrate deletions as jumps
- Integrate insertions as jumps
- Collapse all variations into graph-like structure
- Align against variation graph



Variation-Aware Alignment with PALMapper

Dynamic Program Alignment Against Graph

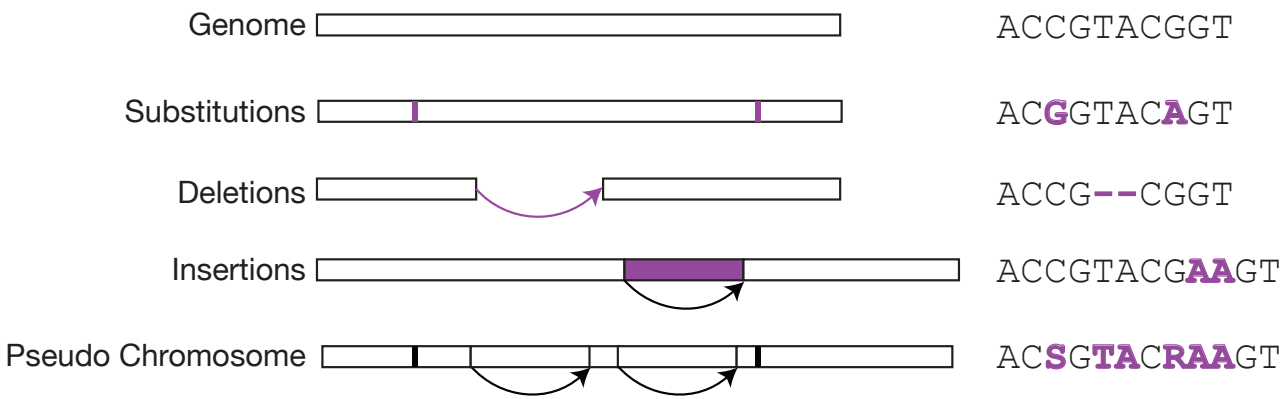
- Integrate substitutions as IUPAC codes for ambiguous bases
- Integrate deletions as jumps
- Integrate insertions as jumps
- Collapse all variations into graph-like structure
- Align against variation graph

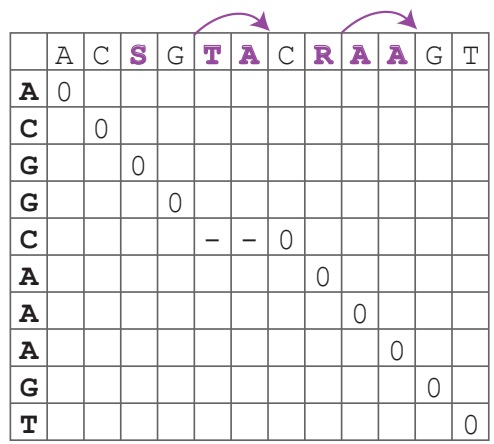


Variation-Aware Alignment with PALMapper

Dynamic Program Alignment Against Graph

- Integrate substitutions as IUPAC codes for ambiguous bases
- Integrate deletions as jumps
- Integrate insertions as jumps
- Collapse all variations into graph-like structure
- Align against variation graph





Read Alignment Summary

Ungapped mapping strategies:

1. find k-mers and extend alignment
   - many hits are possible, heuristics necessary
2. find the whole read in suffix tree/array or BWT/FM index
   - Exact matches found very fast
   - mismatches/indels require backtracking $\Rightarrow$ slower

Gapped Mapping strategies:

1. Build index containing junctions
2. Split reads, align separately, extend to merge
3. Collect regions, perform gapped/banded alignment
4. ...

Great differences between mapping tools: speed (100x), accuracy, completeness, etc.

Read Alignment Summary

Ungapped mapping strategies:

1. find k-mers and extend alignment
   - many hits are possible, heuristics necessary
2. find the whole read in suffix tree/array or BWT/FM index
   - Exact matches found very fast
   - mismatches/indels require backtracking $\Rightarrow$ slower

Gapped Mapping strategies:

1. Build index containing junctions
2. Split reads, align separately, extend to merge
3. Collect regions, perform gapped/banded alignment
4. ...

Great differences between mapping tools: speed (100x), accuracy, completeness, etc.

Read Alignment Summary

Ungapped mapping strategies:

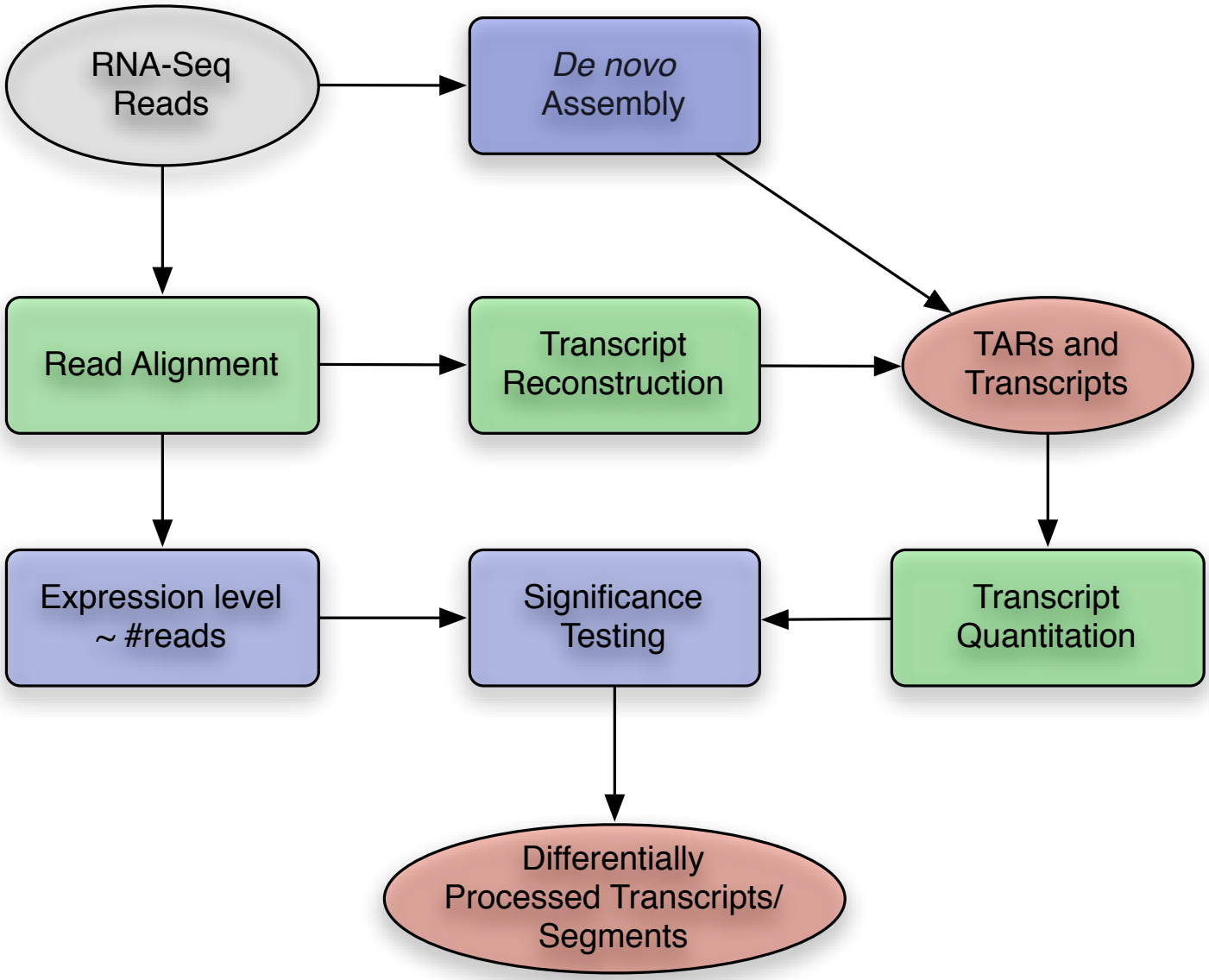
1. find k-mers and extend alignment
   - many hits are possible, heuristics necessary
2. find the whole read in suffix tree/array or BWT/FM index
   - Exact matches found very fast
   - mismatches/indels require backtracking $\Rightarrow$ slower

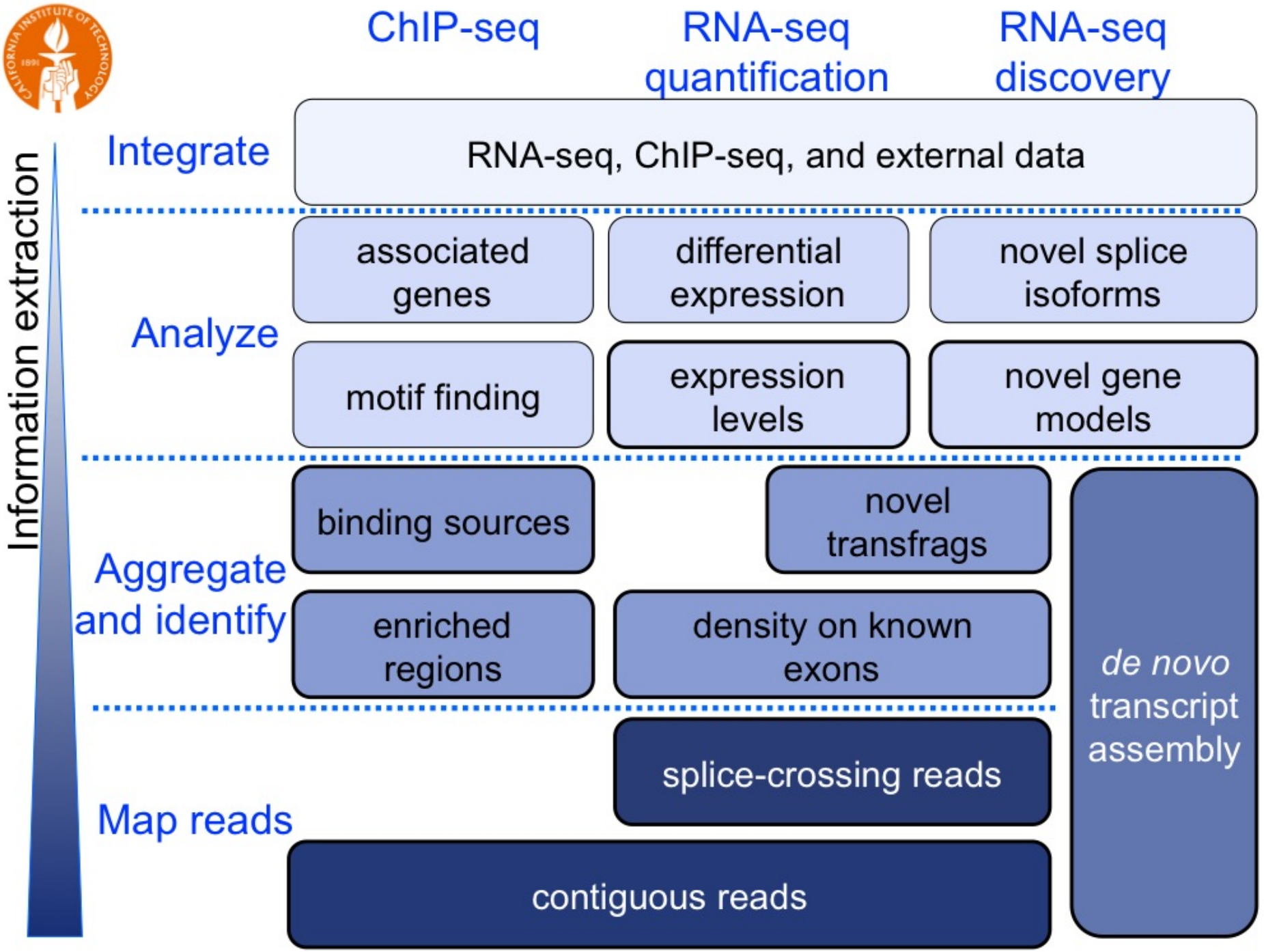
Gapped Mapping strategies:

1. Build index containing junctions
2. Split reads, align separately, extend to merge
3. Collect regions, perform gapped/banded alignment
4. ...

Great differences between mapping tools: speed (100x), accuracy, completeness, etc.

Common RNA-Seq Analysis Steps




ChIP-seq
RNA-seq quantification
RNA-seq discovery
Information extraction
Integrate
RNA-seq, ChIP-seq, and external data
Analyze
associated genes
differential expression
novel splice isoforms
motif finding
expression levels
novel gene models
Aggregate and identify
binding sources
novel transfrags
enriched regions
density on known exons
de novo transcript assembly
Map reads
splice-crossing reads
contiguous reads

RGASP Submission Evaluation

1. *C. elegans*

TAlioto	17G	*126,161,293* algnmts.
LPachter	18G	83,882,605 algnmts.
TWu	16G	112,614,991 algnmts.
CIseli	15G	76,782,527 algnmts.
MStanke	12G	56,175,582 algnmts.
GRaetsch	16G	73,180,783 algnmts.

2. *D. melanogaster* (not shown)

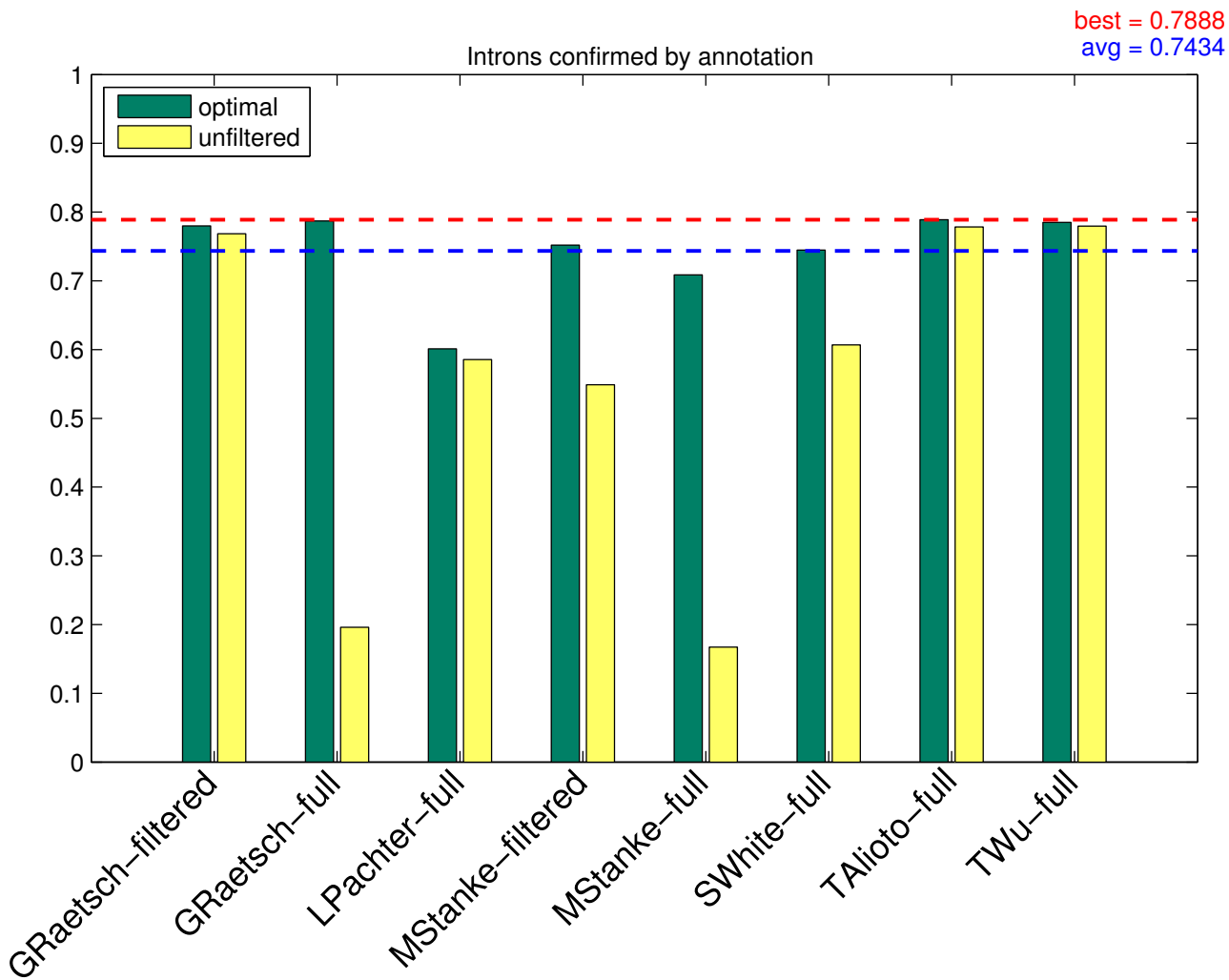
3. Human (chr.1)

TAlioto	3.2G	*23,787,964* algnmts.
LPachter	1.9G	8,689,830 algnmts.
TWu	1.6G	10,925,662 algnmts.
CIseli	1.3G	6,512,751 algnmts.
Ger	1.6G	7,494,284 algnmts.
SWhite	0.4G	6,245,126 algnmts.
GRaetsch	1.6G	6,898,023 algnmts.

Legend:

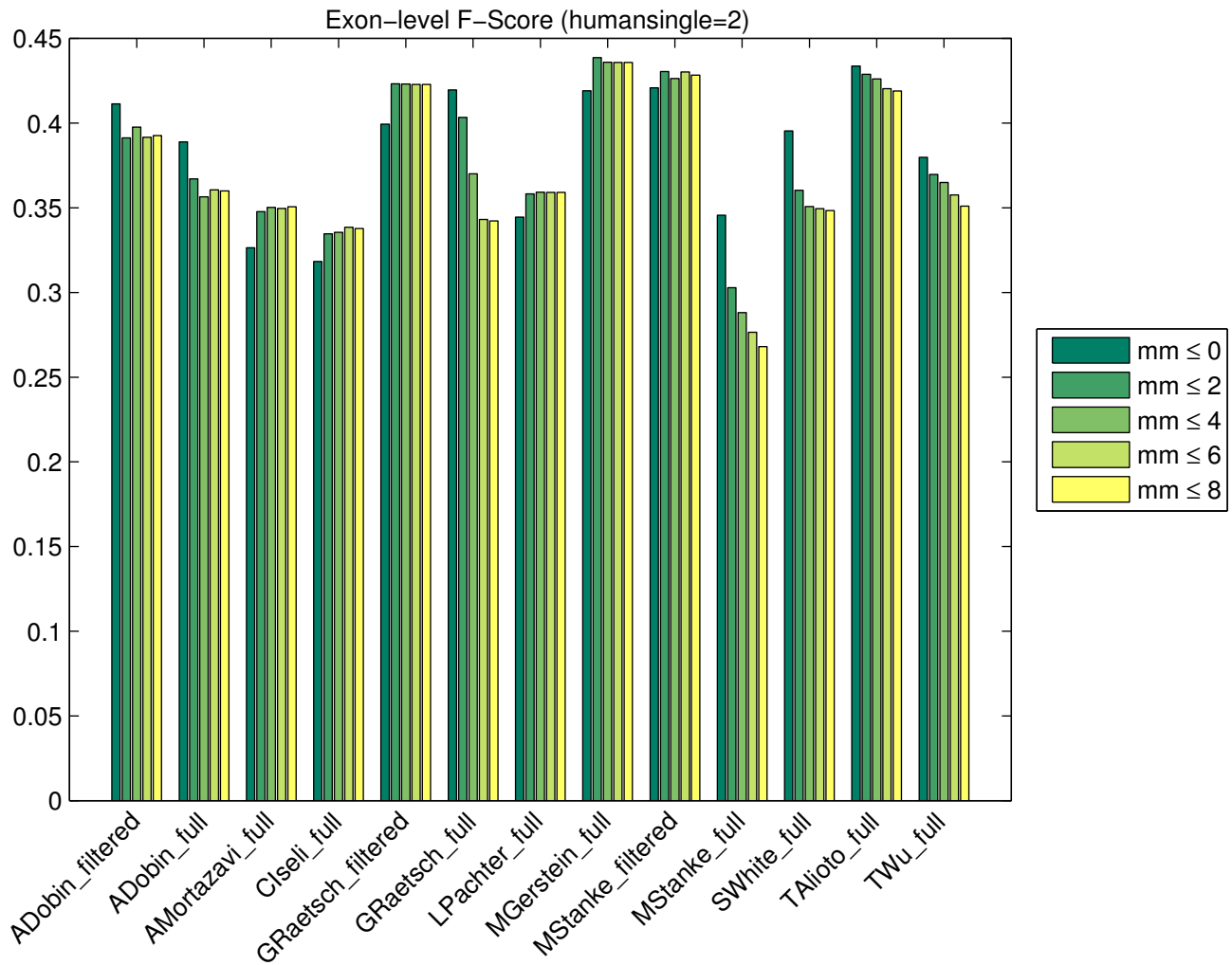
CIseli	Christian Iseli et al., CH
LPachter	Lior Pachter et al., USA (TopHat)
TAlioto	Tyler Alioto et al., ES
SWhite	Simon White et al., UK (Ensembl)
TWu	Thomas Wu, USA (GSnap)
MStanke	Mario Stanke, DE (Blat)
GRaetsch	Gunnar Rätsch et al., DE (Palmapper)

Gapped Alignment Accuracy
(Mapping: Human - Junction F-score)



Filter: combination of edit ops, segment length, and splice support

Effects on Downstream Analysis
(Cufflinks: Human - Exon F-score)



Filter: by max edit ops (0 – 8); prediction F-Score (exon level)

RGASP Summary

Method	Introns	Completeness	Cufflinks	Scripture	Nanostring	# ≤ 3
C. elegans						
GRaetsch	2	3	1	1		4
LPachter		2			3	2
MStanke			3	3		2
SWhite					1	1
TAlioto	1	1	2	2	2	**5**
TWu	3					1
D. melanogaster						
GRaetsch		2	2	2		**3**
LPachter	3		1			2
MStanke	2	1	3			**3**
TAlioto		3		1		2
TWu	1			3		2
Human						
ADobin	1					1
GRaetsch	3		1		2	**3**
LPachter				3		1
MGerstein		2	2	1		**3**
MStanke		1	3		1	**3**
TAlioto		3		2	3	**3**
TWu	2					*1*

Overview

Part I

- RNAs, RNA-seq & research questions
- Common analysis steps
- Read mapping
  - The problem
  - String data structures
  - Programs
  - Extensions
- Differences between read mappers

Part II

- Transcript quantification & biases
- Differential "expression"
- Transcript reconstruction by ...
  - alignment
  - assembly
- Genome annotation

Acknowledgments

We gratefully acknowledge help with the material:

- Jonas Behr, FML & Sloan-Kettering Institute
- Regina Bohnert, FML & Molecular Health
- Philipp Drewe, FML & Sloan-Kettering Institute
- Fabio De Bona, FML & Google Research
- **André Kahles**, FML & Sloan-Kettering Institute
- Stefan Henz, MPI for Developmental Biology
- Cole Trapnell, Broad Institute
- Ali Mortazavi, UC Irvine
- Moran Yassour, Hebrew University & Broad Institute
- Georg Zeller, FML & EMBL Heidelberg
- David Kuo, Sloan-Kettering Institute & Weill Cornell
- Katherine Redfield Chan, Sloan-Kettering Institute

The slides and additional material will be available online at

`http://raetschlab.org/lectures/ows2013`

References I

S.F. Altschul, W. Gish, W. Miller, E.W. Myers, and D.J. Lipman. Basic local alignment search tool. *Journal Molecular Biology*, 215(3):403–10, 1990.

H. Bao, Y. Xiong, H. Guo, R. Zhou, X. Lu, Z. Yang, Y. Zhong, and S. Shi. MapNext: a software tool for spliced and unspliced alignments and SNP detection of short sequence reads. *BMC genomics*, 10(Suppl 3):S13, 2009.

J. Behr, G. Schweikert, J. Cao, F. De Bona, G. Zeller, S. Laubinger, S. Ossowski, K. Schneeberger, D. Weigel, and G. Rätsch. Rna-seq and tiling arrays for improved gene finding. Oral presentation at the CSHL Genome Informatics Meeting, September 2008. URL `http://www.fml.tuebingen.mpg.de/raetsch/lectures/RaetschGenomeInformatics08.pdf`.

M. Bergkessel, G. Wilmes, and C. Guthrie. Snapshot: Formation of mrnps. *Cell*, 136, January 2009.

M Burrows and DJ Wheeler. A block-sorting lossless data compression algorithm. 1994. URL `http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.141.5254`.

D. Campagna, A. Albiero, A. Bilardi, E. Caniato, C. Forcato, S. Manavski, N. Vitulo, and G. Valle. PASS: a program to align short sequences. *Bioinformatics*, 25(7):967, 2009.

RM Clark, G Schweikert, C Toomajian, S Ossowski, G Zeller, P Shinn, N Warthmann, TT Hu, G Fu, DA Hinds, H Chen, KA Frazer, DH Huson, B Schölkopf, M Nordborg, G Rätsch, JR Ecker, and D Weigel. Common sequence polymorphisms shaping genetic diversity in arabidopsis thaliana. *Science*, 317(5836):338–342, 2007. ISSN 1095-9203 (Electronic). doi: 10.1126/science.1138632.

References II

F. De Bona, S. Ossowski, K. Schneeberger, and G. Rätsch. Qpalma: Optimal spliced alignments of short sequence reads. *Bioinformatics*, 24:i174–i180, 2008.

Alexander Dobin, Carrie a Davis, Felix Schlesinger, Jorg Drenkow, Chris Zaleski, Sonali Jha, Philippe Batut, Mark Chaisson, and Thomas R Gingeras. STAR: ultrafast universal RNA-seq aligner. *Bioinformatics (Oxford, England)*, 29(1):15–21, January 2013. ISSN 1367-4811. doi: 10.1093/bioinformatics/bts635. URL `http://www.ncbi.nlm.nih.gov/pubmed/23104886`.

Paolo Ferragina and Giovanni Manzini. An experimental study of a compressed index. *Information Sciences*, 135(1-2):13–28, June 2001. ISSN 00200255. doi: 10.1016/S0020-0255(01)00098-6. URL `http://linkinghub.elsevier.com/retrieve/pii/S0020025501000986`.

L. Florea, G. Hartzell, Z. Zhang, G.M. Rubin, and W. Miller. A computer program for aligning a cdna sequence with a genomic dna sequence. *Genome Research*, 8:967–974, 1998.

M.S. Gelfand, A.A. Mironov, and P.A. Pevzner. Gene recognition via spliced sequence alignment. *Proc. Natl. Acad. Sci.*, 93(17):9061–6, 1996.

Mitchell Guttman, Manuel Garber, Joshua Z Levin, Julie Donaghey, James Robinson, Xian Adiconis, Lin Fan, Magdalena J Koziol, Andreas Gnirke, Chad Nusbaum, John L Rinn, Eric S Lander, and Aviv Regev. Ab initio reconstruction of cell type-specific transcriptomes in mouse reveals the conserved multi-exonic structure of lincrnas. *Nat Biotechnol*, 28(5): 503–10, May 2010. doi: 10.1038/nbt.1633.

References III

Steve Hoffmann, Christian Otto, Stefan Kurtz, Cynthia M Sharma, Philipp Khaitovich, Jörg Vogel, Peter F Stadler, and Jörg Hackermüller. Fast mapping of short sequences with mismatches, insertions and deletions using index structures. *PLoS Comput Biol*, 5(9): e1000502, Sep 2009. doi: 10.1371/journal.pcbi.1000502.

Géraldine Jean, André Kahles, Vipin T Sreedharan, Fabio De Bona, and Gunnar Rätsch. *RNA-Seq read alignments with PALMapper.*, volume Chapter 11. December 2010. ISBN 0471250953. doi: 10.1002/0471250953.bi1106s32. URL `http://www.ncbi.nlm.nih.gov/pubmed/21154708`.

W.J. Kent. BLAT—the BLAST-like alignment tool. *Genome research*, 12(4):656, 2002.

Daehwan Kim, Geo Pertea, Cole Trapnell, Harold Pimentel, Ryan Kelley, and Steven L Salzberg. TopHat2: accurate alignment of transcriptomes in the presence of insertions, deletions and gene fusions. *Genome biology*, 14(4):R36, April 2013. ISSN 1465-6914. doi: 10.1186/gb-2013-14-4-r36. URL `http://www.ncbi.nlm.nih.gov/pubmed/23618408`.

Ben Langmead, Cole Trapnell, Mihai Pop, and Steven L Salzberg. Ultrafast and memory-efficient alignment of short dna sequences to the human genome. *Genome Biol*, 10 (3):R25, 2009. doi: 10.1186/gb-2009-10-3-r25.

Heng Li and Richard Durbin. Fast and accurate short read alignment with burrows-wheeler transform. *Bioinformatics*, 25(14):1754–60, Jul 2009. doi: 10.1093/bioinformatics/btp324.

Heng Li and Richard Durbin. Fast and accurate long-read alignment with burrows-wheeler transform. *Bioinformatics*, 26(5):589–95, Mar 2010. doi: 10.1093/bioinformatics/btp698.

References IV

Heng Li, Jue Ruan, and Richard Durbin. Mapping short dna sequencing reads and calling variants using mapping quality scores. *Genome Res*, 18(11):1851–8, Nov 2008. doi: 10.1101/gr.078212.108.

Ruiqiang Li, Chang Yu, Yingrui Li, Tak-Wah Lam, Siu-Ming Yiu, Karsten Kristiansen, and Jun Wang. Soap2: an improved ultrafast tool for short read alignment. *Bioinformatics*, 25(15): 1966–7, Aug 2009. doi: 10.1093/bioinformatics/btp336.

Udi Manber and Gene Myers. Suffix arrays: a new method for on-line string searches. *siam Journal on Computing*, 1993. URL `http://epubs.siam.org/doi/abs/10.1137/0222058`.

Santiago Marco-Sola, Michael Sammeth, Roderic Guigó, and Paolo Ribeca. The GEM mapper: fast, accurate and versatile alignment by filtration. *Nature methods*, (octobeR), October 2012. ISSN 1548-7105. doi: 10.1038/nmeth.2221. URL `http://www.ncbi.nlm.nih.gov/pubmed/23103880`.

Edward M McCreight. A Space-Economical Suffix Tree Construction Algorithm. *Journal of the Association of Computing Machinery*, 23(2):262–272, 1976.

A. Mortazavi, B.A. Williams, K. McCue, L. Schaeffer, and B. Wold. Mapping and quantifying mammalian transcriptomes by RNA-Seq. *Nature methods*, 5(7):621–628, 2008.

G. Rätsch and S. Sonnenburg. Accurate splice site detection for *Caenorhabditis elegans*. In K. Tsuda B. Schoelkopf and J.-P. Vert, editors, *Kernel Methods in Computational Biology*. MIT Press, 2004.

G. Rätsch, S. Sonnenburg, and B. Schölkopf. RASE: recognition of alternatively spliced exons in *C. elegans*. *Bioinformatics*, 21(Suppl. 1):i369–i377, June 2005.

References V

G Rätsch, G Jean, A Kahles, S Sonnenburg, F De Bona, K Schneeberger, J Hagmann, and D Weigel. PALMapper: Fast and accurate alignment of RNA-seq reads. in preparation, 2010.

Stephen M Rumble, Phil Lacroute, Adrian V Dalca, Marc Fiume, Arend Sidow, and Michael Brudno. Shrimp: accurate mapping of short color-space reads. *PLoS Comput Biol*, 5(5): e1000386, May 2009. doi: 10.1371/journal.pcbi.1000386.

Korbinian Schneeberger, Jörg Hagmann, Stephan Ossowski, Norman Warthmann, Sandra Gesing, Oliver Kohlbacher, and Detlef Weigel. Simultaneous alignment of short reads against multiple genomes. *Genome Biol*, 10(9):R98, 2009a. doi: 10.1186/gb-2009-10-9-r98.

Korbinian Schneeberger, Jörg Hagmann, Stephan Ossowski, Norman Warthmann, Sandra Gesing, Oliver Kohlbacher, and Detlef Weigel. Simultaneous alignment of short reads against multiple genomes. *Genome Biol*, 10(9):R98, Jan 2009b. doi: 10.1186/gb-2009-10-9-r98. URL `http://genomebiology.com/2009/10/9/R98`.

U. Schulze, B. Hepp, C.S. Ong, and G. Ratsch. PALMA: mRNA to genome alignments using large margin algorithms. *Bioinformatics*, 23(15):1892, 2007.

Gabriele Schweikert, Alexander Zien, Georg Zeller, Jonas Behr, Christoph Dieterich, Cheng Soon Ong, Petra Philips, Fabio De Bona, Lisa Hartmann, Anja Bohlen, Nina Krüger, Sören Sonnenburg, and Gunnar Rätsch. mgene: Accurate svm-based gene finding with an application to nematode genomes. *Genome Research*, 2009. URL `http://genome.cshlp.org/content/early/2009/06/29/gr.090597.108.full.pdf+html`. Advance access June 29, 2009.

References VI

S. Sonnenburg, G. Rätsch, A. Jagota, and K.-R. Müller. New methods for splice-site recognition. In *Proc. International Conference on Artificial Neural Networks*, 2002.

Sören Sonnenburg, Alexander Zien, and Gunnar Rätsch. ARTS: Accurate Recognition of Transcription Starts in Human. *Bioinformatics*, 22(14):e472–480, 2006.

I. Sutskever. Arachne: A whole genome shotgun assembler. oral presentation, 2008.

C. Trapnell, L. Pachter, and S.L. Salzberg. TopHat: discovering splice junctions with RNA-Seq. *Bioinformatics*, 25(9):1105, 2009.

Cole Trapnell, Brian A Williams, Geo Pertea, Ali Mortazavi, Gordon Kwan, Marijke J van Baren, Steven L Salzberg, Barbara J Wold, and Lior Pachter. Transcript assembly and quantification by rna-seq reveals unannotated transcripts and isoform switching during cell differentiation. *Nature Biotech*, advance online publication, May 2010. doi: 10.1038/nbt.1621. URL `http://dx.doi.org/10.1038/nbt.1621`.

E. Ukkonen. On-line construction of suffix trees. *Algorithmica*, 14(3):249–260, September 1995. ISSN 0178-4617. doi: 10.1007/BF01206331. URL `http://link.springer.com/10.1007/BF01206331`.

J. Usuka, W. Zhu, and V. Brendel. Optimal spliced alignment of homologous cdna to a genomic dna template. *Bioinformatics*, 16(3):203–211, 2000.

References VII

Kai Wang, Darshan Singh, Zheng Zeng, Stephen J Coleman, Yan Huang, Gleb L Savich, Xiaping He, Piotr Mieczkowski, Sara a Grimm, Charles M Perou, James N MacLeod, Derek Y Chiang, Jan F Prins, and Jinze Liu. MapSplice: accurate mapping of RNA-seq reads for splice junction discovery. *Nucleic acids research*, 38(18):e178, October 2010. ISSN 1362-4962. doi: 10.1093/nar/gkq622. URL `http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=2952873&tool=pmcentrez&rendertype=abstract`.

Peter Weiner. Linear Pattern Matching Algorithms. *Switching and Automata Theory, 1973. SWAT'08. ...*, 1973. URL `http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=4569722`.

Ostell JM. Wheelan SJ, Church DM. Spidey: a tool for mrna-to-genomic alignments. *Genome Research*, 11(11):1952–7, 2001.

G Zeller, RM Clark, K Schneeberger, A Bohlen, D Weigel, and G Ratsch. Detecting polymorphic regions in arabidopsis thaliana with resequencing microarrays. *Genome Res*, 18 (6):918–929, 2008. ISSN 1088-9051 (Print). doi: 10.1101/gr.070169.107.

M. Zhang and W. Gish. Improved spliced alignment from an information theoretic approach. *Bioinformatics*, 22(1):13–20, January 2006.

A. Zien, G. Rätsch, S. Mika, B. Schölkopf, T. Lengauer, and K.-R. Müller. Engineering Support Vector Machine Kernels That Recognize Translation Initiation Sites. *BioInformatics*, 16(9): 799–807, September 2000.