Next Generation Sequencing and Short Read Alignment

Abrita Chakravarty

CPS160

Fall 2010

Outline

- Next generation sequencing technologies
- Types of analyses possible
- Computational problems: Mapping and De-novo assembly
- Short read alignment:
  - ✓ What is the problem?
  - ✓ One way to solve it
  - ✓ Popular approaches to solve it
  - ✓ Examples of short read aligners

Next Generation Sequencing Technologies

- Rapid, inexpensive sequencing of billions of bases
- Roche/454:
  - ✓ 1x10^6 reads, 450-500 bp, in 8-hour run
- Illumina/Solexa platform:
  - ✓ 50x10^6 reads, 35 bp, in 2 days.
- ABI Solid sequencing

NGS characteristics

- High throughput - parallelize the sequencing process
- Less time - millions of sequences at once
- Low cost - materials and methods

Generated data:

- Short reads (E.g. Illumina/solexa 35-70bp)
- Many reads (billions)

What can we do with this data?

Genomic analyses using short reads

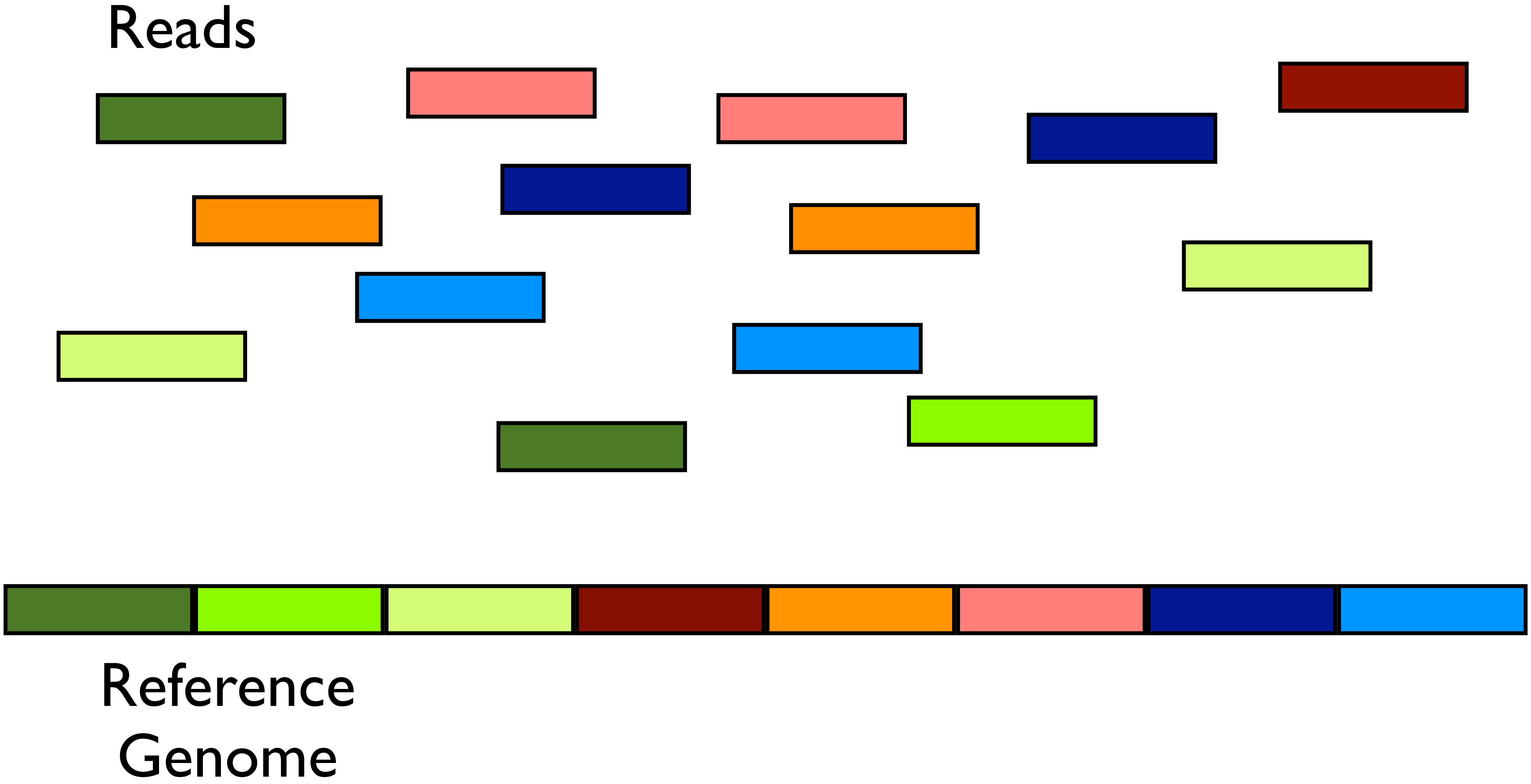
- Whole genome sequencing
- Genome resequencing
- Sequencing-based assays
  - ChIP-Seq: sequences immunoprecipitated DNA fragments
  - RNA-Seq: sequences mRNA converted to cDNA

Sequencing-based assays

- ChIP-Seq
  - ✓ Gene interaction with transcription factors and other proteins
  - ✓ Genomic/Epigenomic annotations
- RNA-Seq
  - ✓ Gene expression
  - ✓ Alternative splicing
  - ✓ Identification of previously unknown genes

How are the reads used?

Mapping: short read alignment

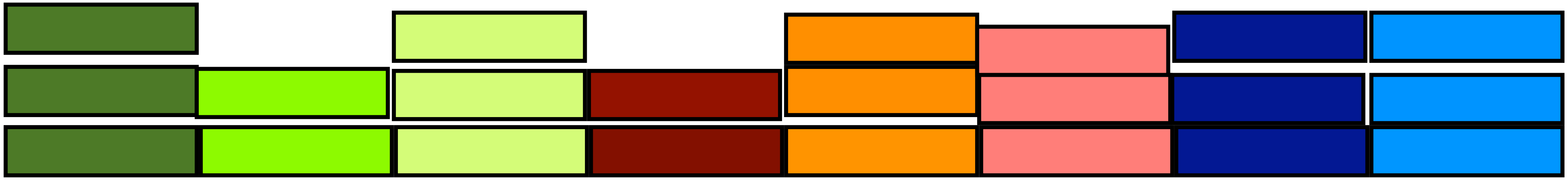


Reads

Reference Genome

Mapping: short read alignment

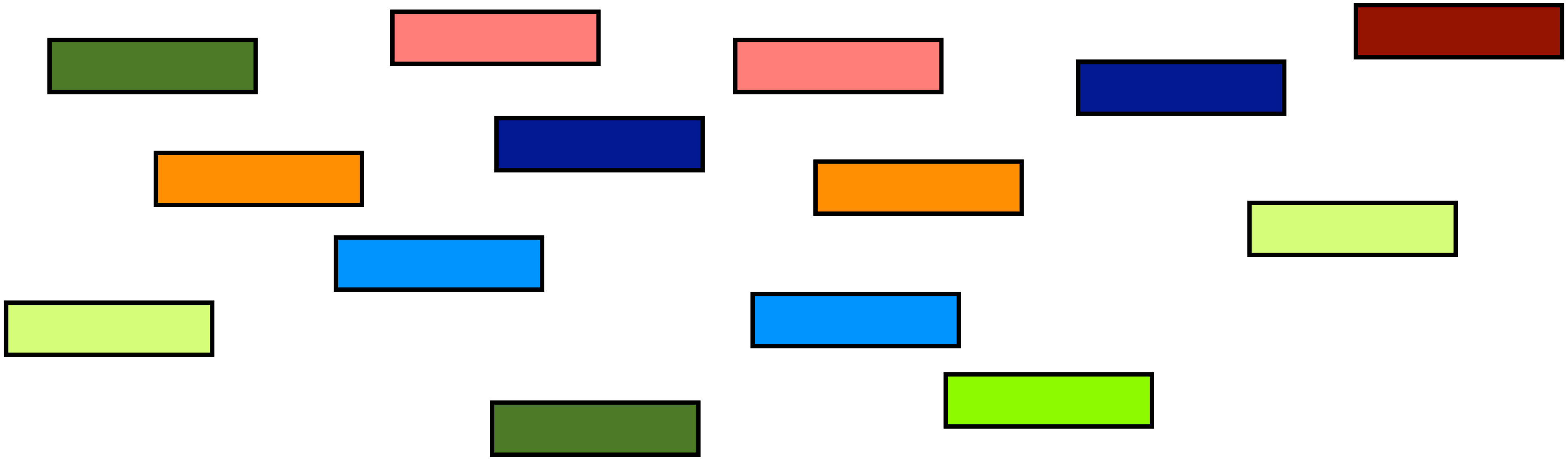
Reads



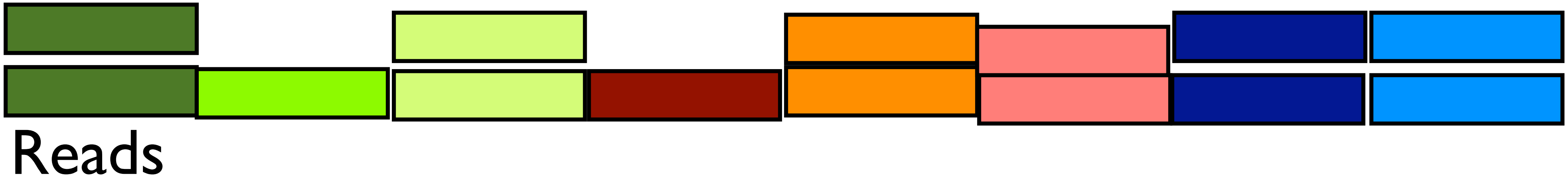
Reference Genome

De-novo assembly

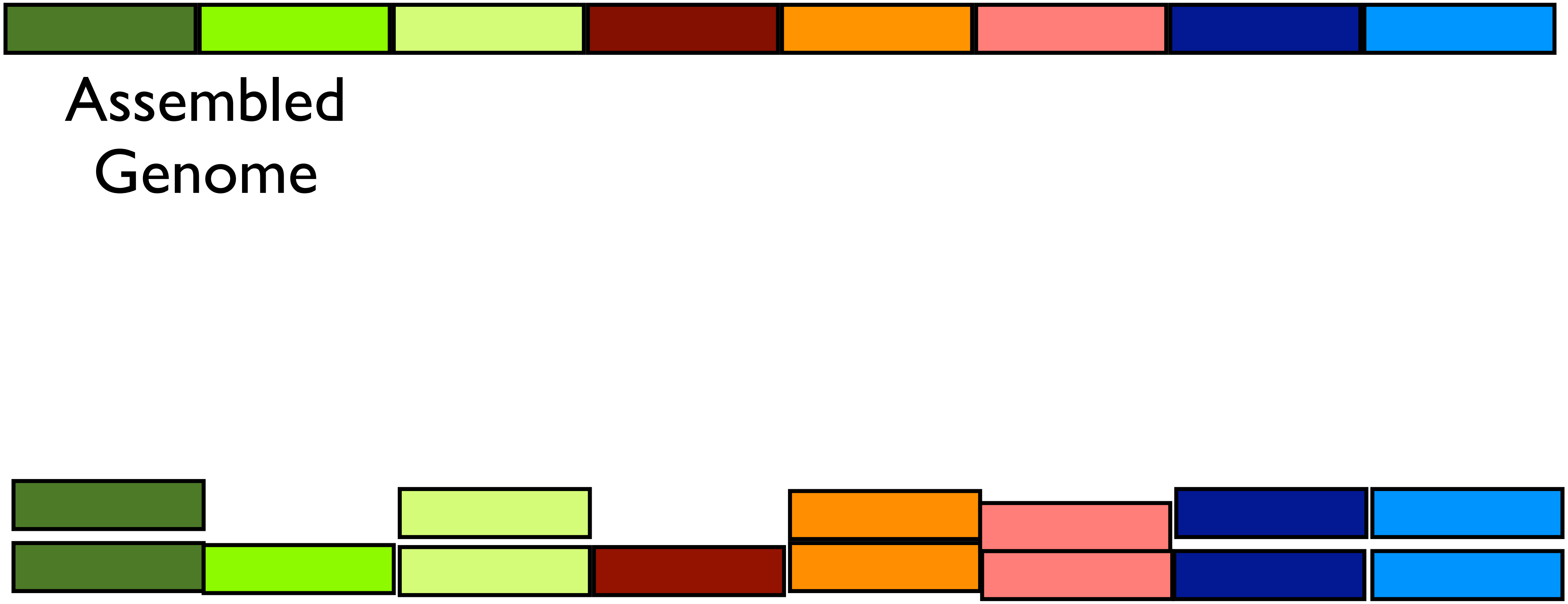
Reads



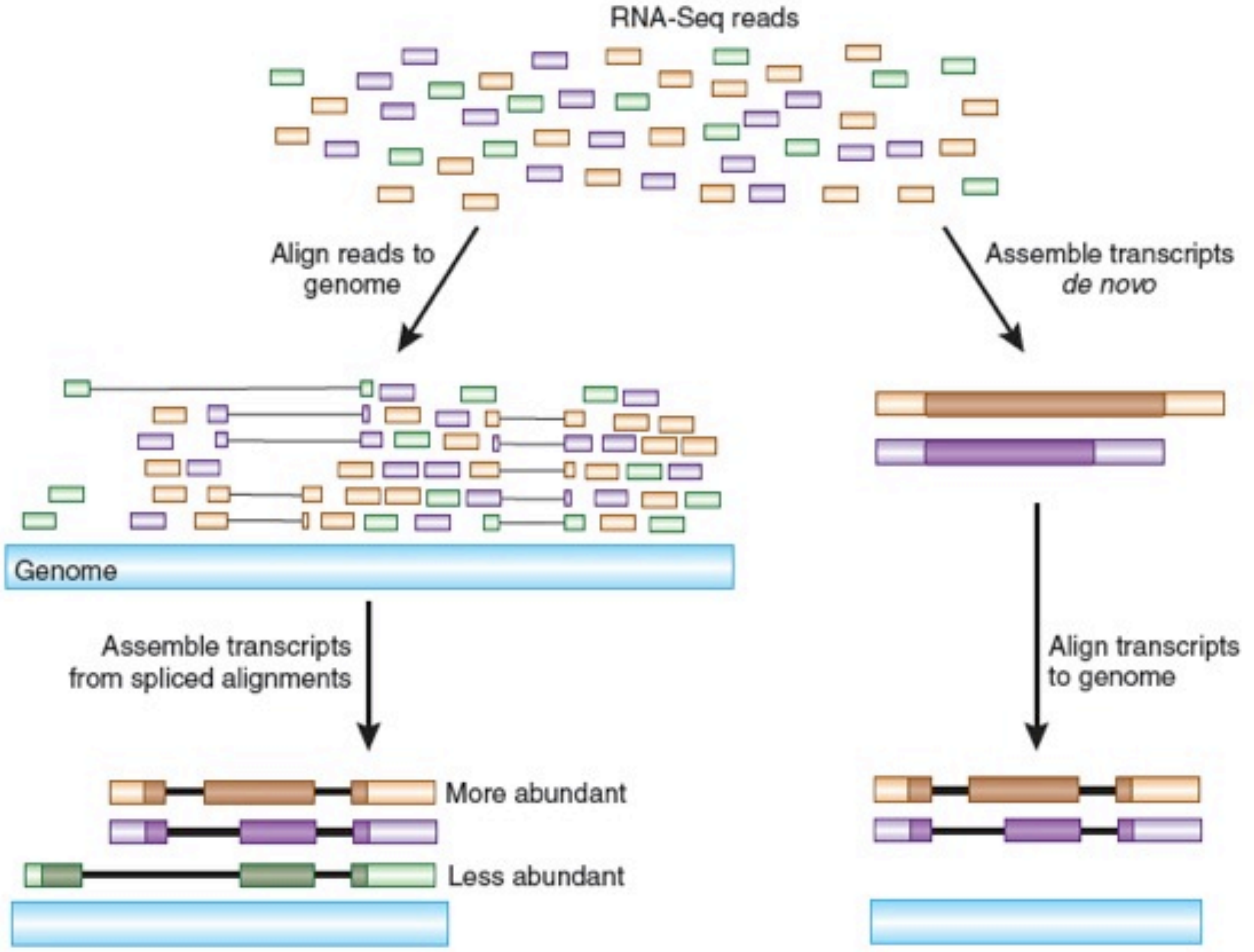
De-novo assembly



De-novo assembly



RNA-seq analysis


RNA-Seq reads
Align reads to genome
Assemble transcripts *de novo*
Genome
Assemble transcripts from spliced alignments
Align transcripts to genome
More abundant
Less abundant

http://www.nature.com/nbt/journal/v28/n5/full/nbt0510-421.html

What are the computational challenges?

The *read mapping* problem

- Sequenced reads are short (35-70 bp)
- Must be mapped to unique positions in reference genome (billions of bp)
- Reads have sequencing errors
- Reference genome has repetitive elements
- Orientation of read relative to reference genome not known
- Genome from which reads are generated may have diverged from reference genome

Short read alignment

- How can we align the reads to the reference genome
  - Efficiently in terms of time and memory
  - Account for inexact pattern matching and ambiguous locations to map to

Exact pattern matching

- Given: a long piece of text, and a much smaller pattern (in the same alphabet)
- Find the locations in the text where the pattern occurs

Exercise

- Find **AGG** in

```
CTCGAGGGGCCTAGACATTGCCCTCCAGAGAGAG
CACCCAACACCCTCCAGGCTTGACCGGCCAGGGT
GTCCCCTTCCTACCTTGGAGAGAGCAGCCCCAGG
GCATCCTGCAGGGGGTGCTGGGACACCAGCTGGC
CTTCAAGGTCTCTGCCTCCCTCCAGCCACCCCAC
TACACGCTGCTGGGATCCTGGATCTCAGCTCCCT
GGCCGACAACACTGGCAAACTCCTACTCATCCAC
GAAGGCCCTCCTGGGCATGGTGGTCCTTCCCAGC
CTGGCAGTCTGTTCCTCACACACCTTGTTAGTGC
CCAGCCCCTGAGGTTGCAGCTGGGGGTGTCTCTG
```

Exercise

- Find **AGG** in

CTCG**AGG**GGCCTAGACATTGCCCTCCAGAGAGAG
CACCCAACACCCTCC**AGG**CTTGACCGGCC**AGG**GT
GTCCCCTTCCTACCTTGGAGAGAGCAGCCCC**AGG**
GCATCCTGC**AGG**GGGTGCTGGGACACCAGCTGGC
CTTCA**AGG**TCTCTGCCTCCCTCCAGCCACCCCAC
TACACGCTGCTGGGATCCTGGATCTCAGCTCCCT
GGCCGACAACACTGGCAAACTCCTACTCATCCAC
GA**AGG**CCCTCCTGGGCATGGTGGTCCTTCCCAGC
CTGGCAGTCTGTTCCTCACACACCTTGTTAGTGC
CCAGCCCCTG**AGG**TTGCAGCTGGGGGTGTCTCTG

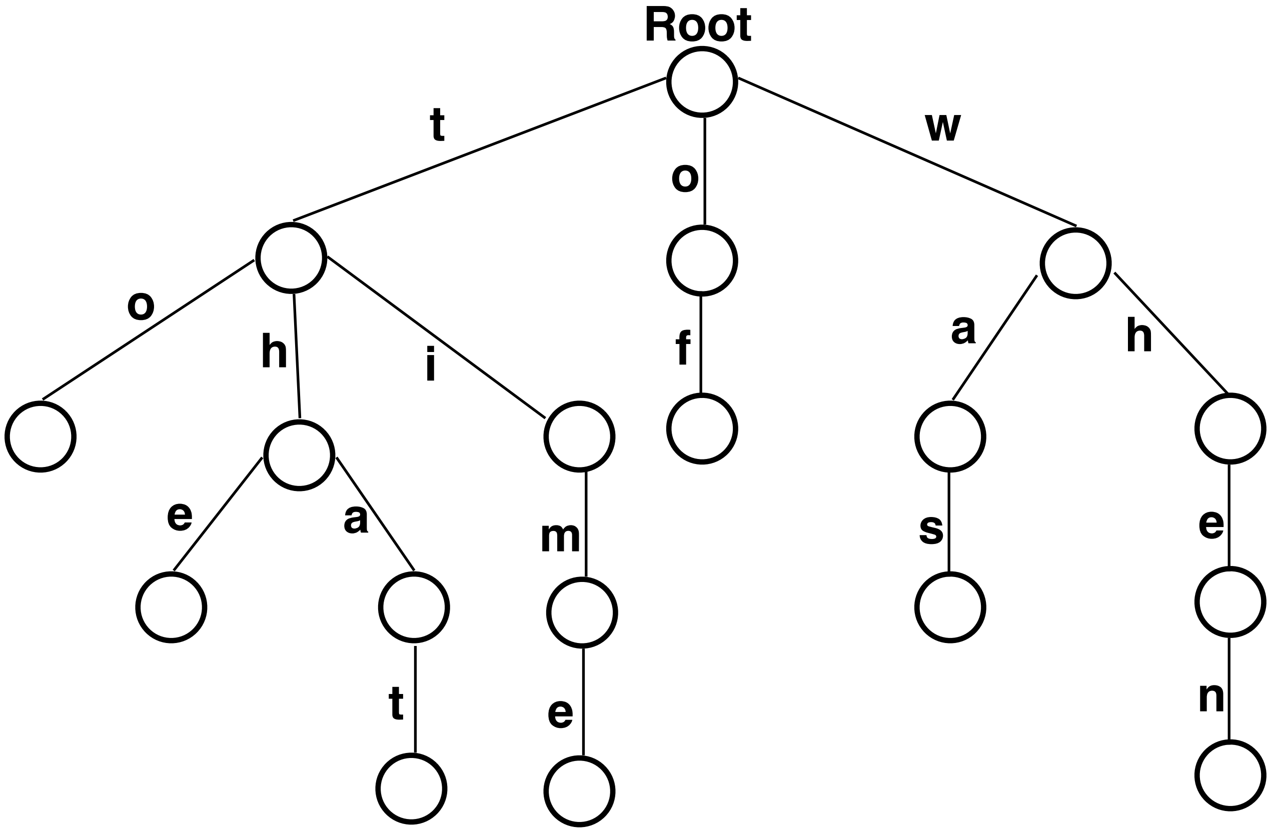
How to do this efficiently?

- Index the text
- Use efficient data structure to store the index
- Optimize time: search for matches quickly
- Optimize space: must fit in the memory

The keyword tree

Text: ***It was the best of times***.

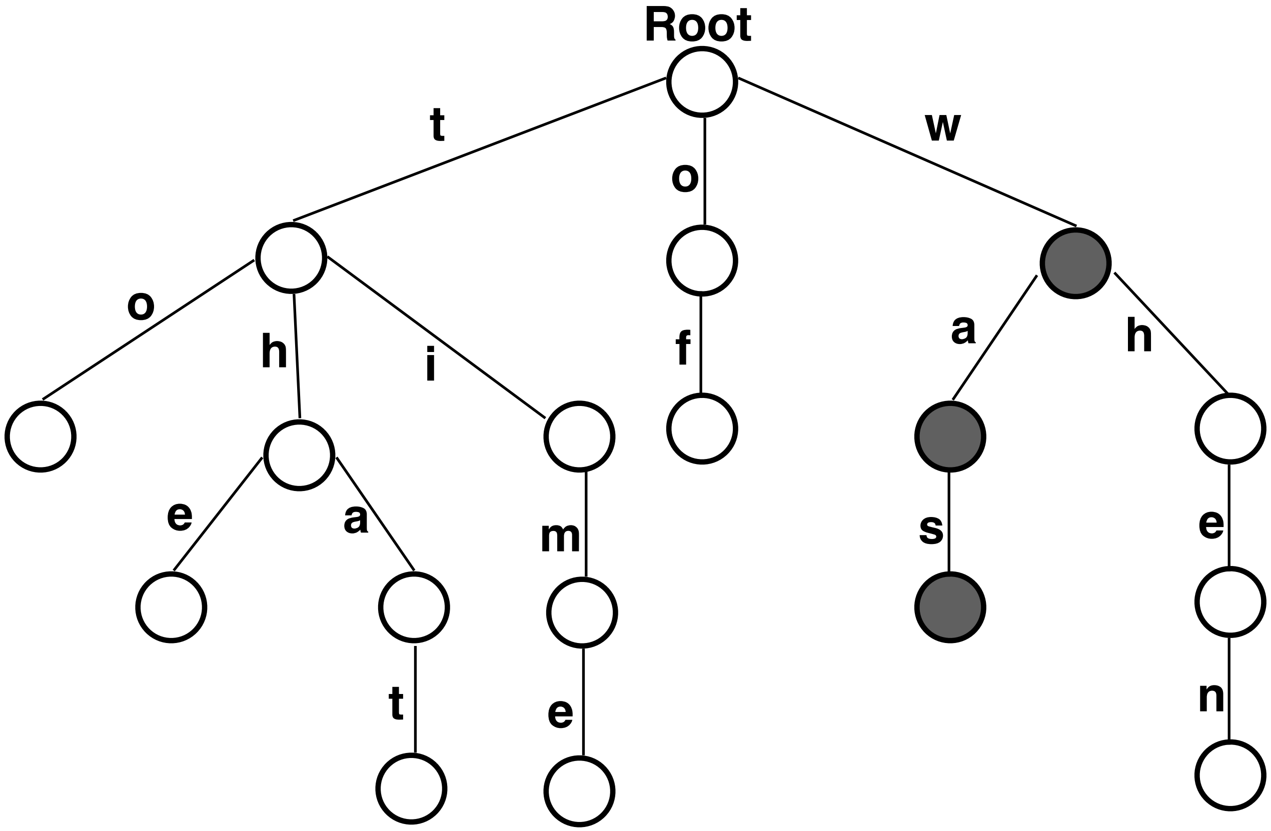
Patterns: to, the, that, time, of, was, when



The keyword tree

Text: ***It was the best of times***.

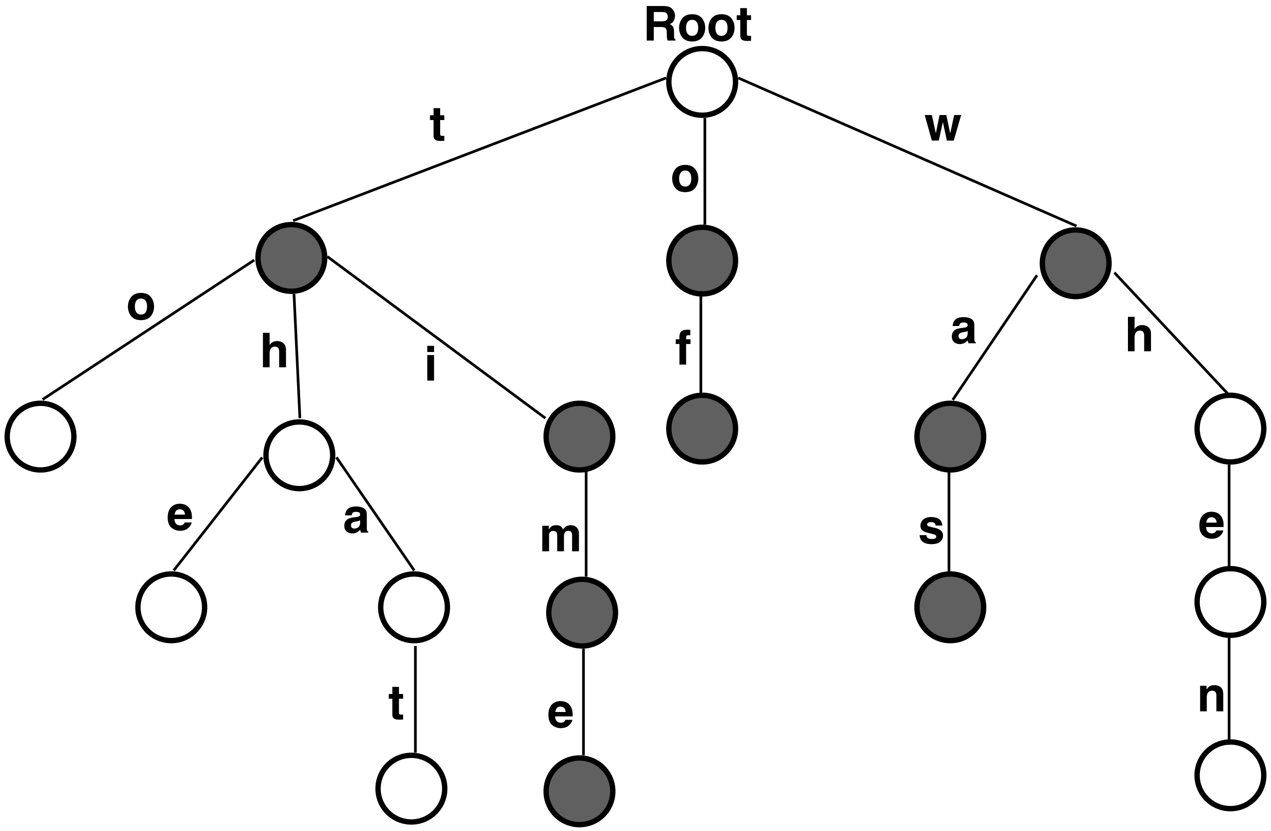
Patterns: to, the, that, time, of, was, when



The keyword tree

Text: ***It was the best of times***.

Patterns: to, the, that, time, of, was, when



The suffix tree

Text: ATCTAATG

Suffixes:

1)ATCTAATG
2)TCTAATG
3)CTAATG
4)TAATG
5)AATG
6)ATG
7)TG
8)G

Length of text = m
Number of suffixes = m
Total length of suffixes
$= m + m\text{-}1 + \ldots + 3 + 2 + 1$
$= m(m+1)/2$
$= O(m^2)$

Building the suffix tree

Text: ATCTAATG$

Suffixes:

1)ATCTAATG$

2)TCTAATG$

3)CTAATG$

4)TAATG$

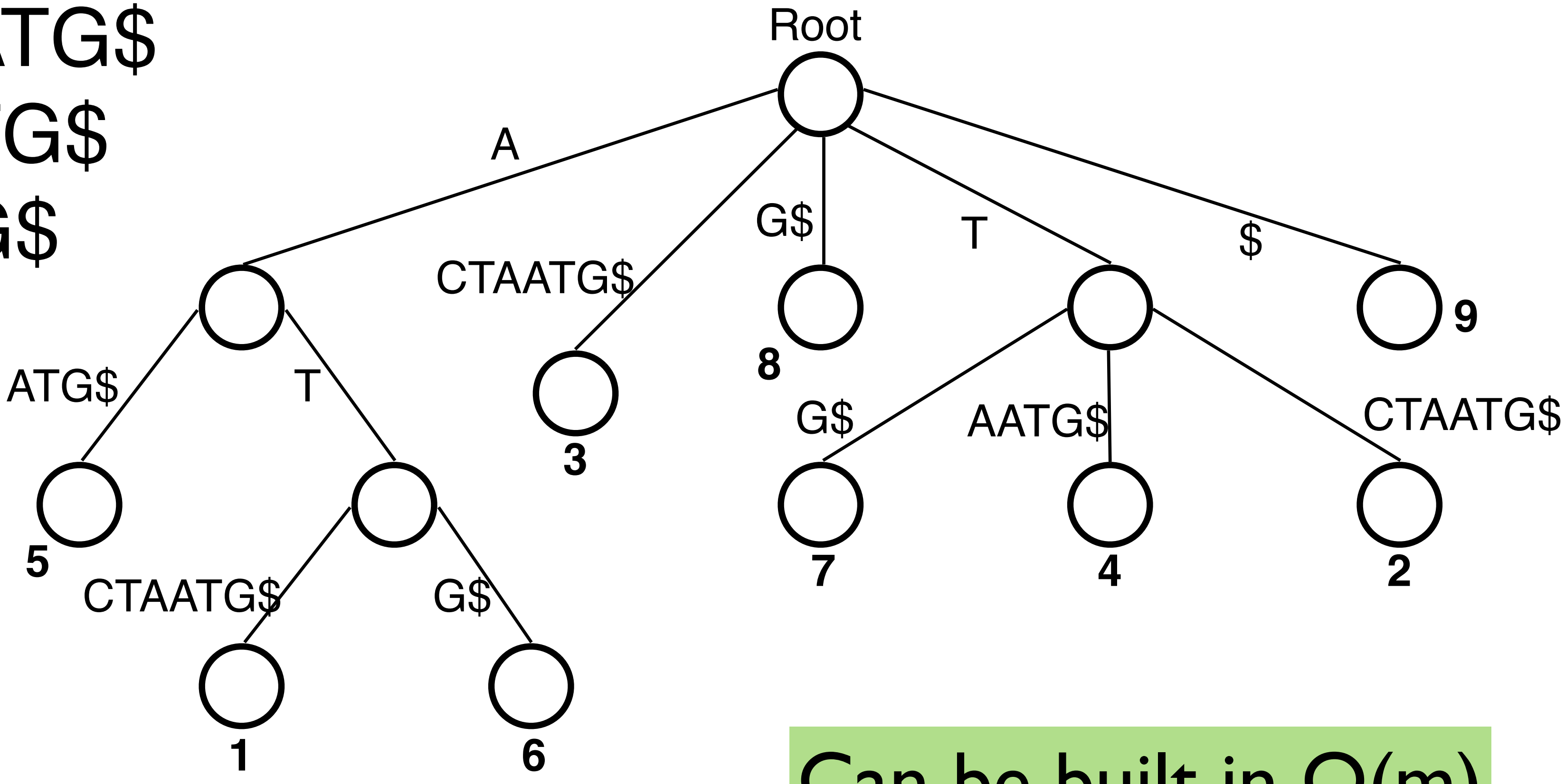
5)AATG$

6)ATG$

7)TG$

8)G$

9)$



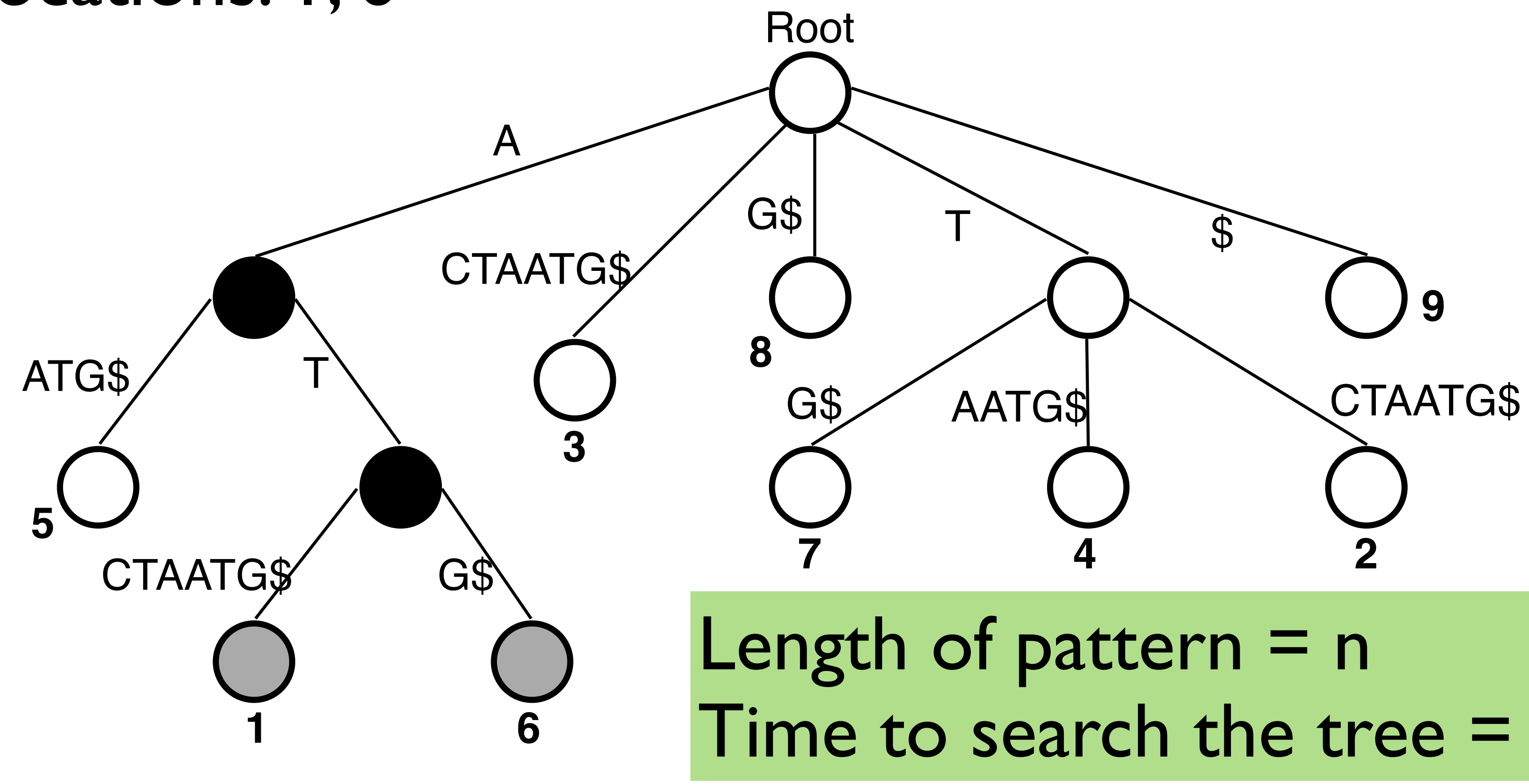
Can be built in O(m)

Threading the suffix tree

Sequence: ATCTAATG

Read: AT

Locations: 1, 6



Length of pattern = n

Time to search the tree = O(n)

Inexact pattern matching

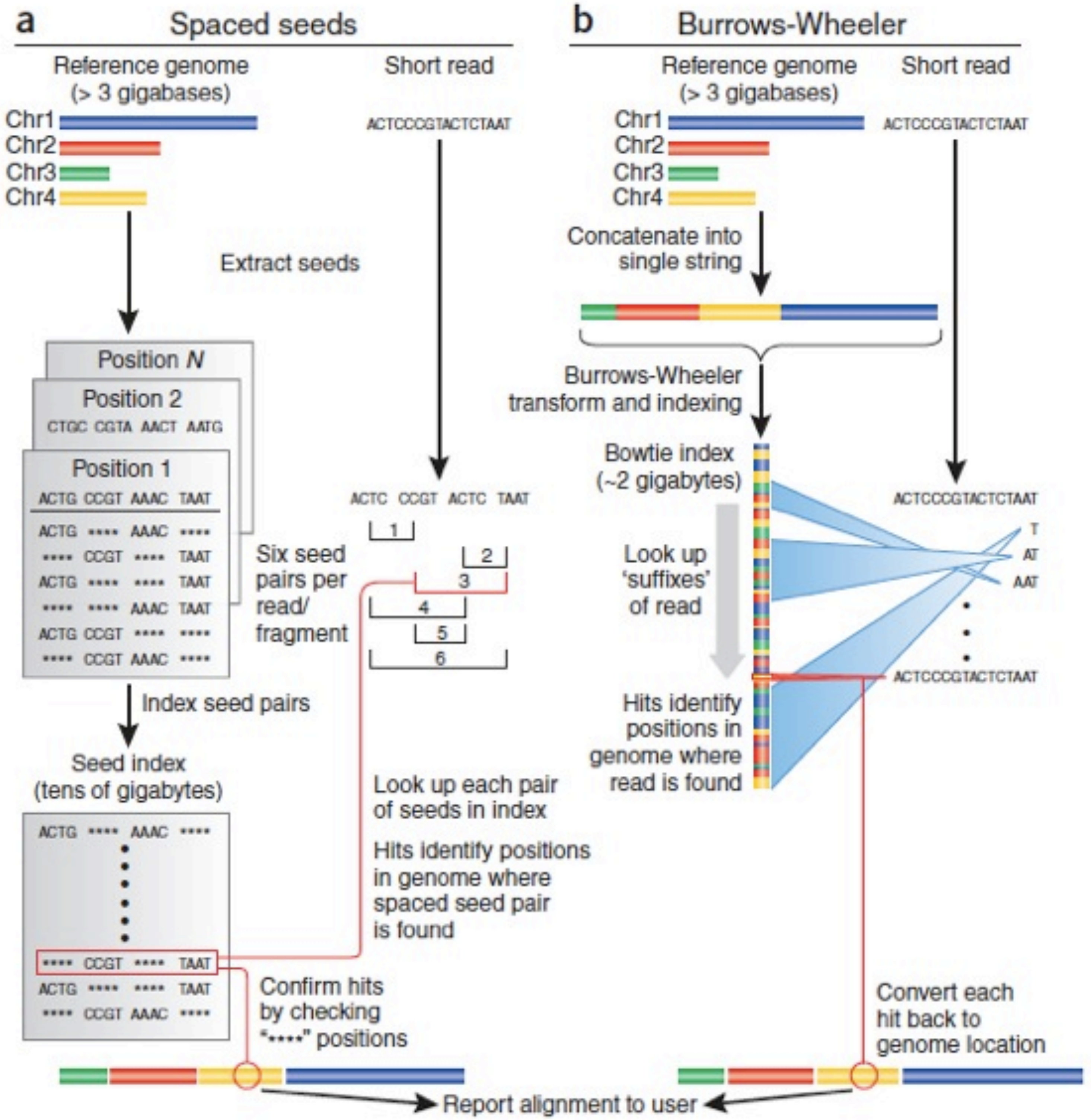
- Given: a long piece of text, and a much smaller pattern (in the same alphabet)
- Find the location in the text where the pattern occurs
- Allow for a predetermined number of mismatches.

Spaced seed indexing - MAQ

- Read/fragment divided into 4 equal segments - seeds
  - ✓ If entire read aligns, all seeds align perfectly
  - ✓ If there is one mismatch, one seed (with the mismatch) will not align, other three will
  - ✓ If there are two mismatches, two seeds will align
- For two mismatches - 6 possible pairs of "aligned" seeds
- Create spaced seed index to search against
- Narrow search to hits for spaced seed pairs

Burrows-Wheeler transform

- Used in Bowtie, SOAP2
- Transform helps to index entire human genome in less than 2 gb memory
- Aligner matches suffixes of reads against the index (increasing one character at a time)
- If perfect alignment not found, goes back; substitutes a character in the read; resumes



http://www.nature.com/nbt/journal/v27/n5/full/nbt0509-455.html

Short read aligners

- MAQ: *Mapping and Assembly with Quality*
- ELAND: Proprietary program from Illumina
- SOAP: *Short Oligonucleotide Alignment Program*
- Bowtie: using Burrows-Wheeler Transform
- SHREC: *Short Read Error Correction* (uses suffix tree)
- http://en.wikipedia.org/wiki/List_of_sequence_alignment_software#Short-Read_Sequence_Alignment

De-novo assembly

- Problems:
  - ✓ Hard to distinguish correct assembly from repetitive sequence overlap
  - ✓ Difficult to record in memory all the sequence overlap information
- Strategies:
  - ✓ De Bruijn graphs
  - ✓ Overlap and extension
- Available programs: EULER, Velvet, ALLPATHS, SSAKE

Challenges

- Even with few sequencing errors and quality filters for reads, 70-75% reads successfully mapped
- Reads increasing in length > 100bp
- Spliced read mapping across exon-intron junction
- How best to use quality scores to handle sequencing errors?
- How to account for insertions and deletions in reads?