How Do We Locate Disease- Causing Mutations?

Combinatorial Pattern Matching

Phillip Compeau and Pavel Pevzner.

Bioinformatics Algorithms: an Active Learning Approach

©2018 by Compeau and Pevzner. All rights reserved

Outline

- **Why do we map reads?**
- Using the Trie
- From a Trie to a Suffix Tree
- String Compression and the Burrows-Wheeler Transform
- Inverting Burrows-Wheeler
- Using Burrows-Wheeler for Pattern Matching
- Finding the Matched Patterns
- Setting Up Checkpoints
- Inexact Matching
- Further Applications of Read Mapping

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

Sequencing Costs Plummet

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

From Species to Personal Genomes

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

From Species to Personal Genomes

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

From Species to Personal Genomes

```
CTGATGATGGACTACGCTA
CTACTGCTAGCTGTATTAC
GATCAGCTACCACATCGTA
GCTACGATGCATTAGCAAG
CTATCGATCGATCGATCGA
TTATCTACGATCGATCGAT
CGATCACTATACGAGCTAC
TACGTACGTACGATCGCGG
GACTATTATCGACTACAGA
TAAAACATGCTAGTACAAC
AGTATACATAGCTGCGGGA
TACGATTAGCTAATAGCTG
ACGATATCCGAT
```

```
CTGATGATGGACTACGCTA
CTACTGCTAGCTGTATTAC
GATCAGCTACAACATCGTA
GCTACGATGCATTAGCAAG
CTATCGATCGATCGATCGA
TTATCTACGATCGATCGAT
CGATCACTATACGAGCTAC
TACGTACGTACGATCGCGT
GACTATTATCGACTACAGA
TGAAACATGCTAGTACAAC
AGTATACATAGCTGCGGGA
TACGATTAGCTAATAGCTG
ACGATATCCGAT
```

```
CTGATGATGGACTACGCTA
CTACTGCTAGCTGTATTAC
GATCAGCTACTACATCGTA
GCTACGATGCATTAGCAAG
CTATCGATCGATCGATCGA
TTATCTACGATCGATCGAT
CGATCACTATACGAGCTAC
TACGTACGTACGATCGCGA
GACTATTATCGACTACAGA
TCAAACATGCTAGTACAAC
AGTATACATAGCTGCGGGA
TACGATTAGCTAATAGCTG
ACGATATCCGAT
```

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

Why Personal Genomics?

- Nicholas Volker: first person to have life saved by genome sequencing (2010).

- Personal genomics can help us understand the genetic basis of diseases.

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

Genomes Meet the Crowd

- Personal Genome Project UK (2013): 100,000 human genomes

- Ethical issues?

- US: Genetic Information Nondiscrimination Act (2008)

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

Toward a Computational Problem

- **Reference genome**: database genome used for comparison.

- **Question:** How can we assemble individual genomes efficiently using the reference?

CTGATGATGGACTACGCTACTACTGCTAGCTGTAT Individual

CTGAGGATGGACTACGCTACTACTGATAGCTGTTT Reference

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

Why Not Use Assembly?

Multiple copies of a genome

Shatter the genome into reads

Sequence the reads

Assemble the genome with overlapping reads

AGAATATCA

GAGAATATC

TGAGAATAT

...TGAGAATATCA...

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

Why Not Use Assembly?

- Constructing a de Bruijn graph takes a lot of memory.

- Hope: a machine in a clinic that would collect and map reads in 10 minutes.

- Idea: use existing structure of reference genome to help us sequence a patient's genome.

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

Read Mapping

- **Read mapping**: determine where each read has high similarity to the reference genome.

```
CTGAGGATGGACTACGCTACTACTGATAGCTGTTT   Reference
  GAGGA      CCACG          TGA-A     Reads
```

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

Why Not Use Alignment?

- **Fitting alignment:** align each read *Pattern* to the best substring of *Genome*.

- Has runtime O(|*Pattern*| * |*Genome*|) for each *Pattern*.

- Has runtime O(|*Patterns*| * |*Genome*|) for a collection of *Patterns*.

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

Exact Pattern Matching

- Focus on a simple question: where do the reads match the reference genome *exactly*?

- **Single Pattern Matching Problem:**
  - **Input:** A string *Pattern* and a string *Genome.*
  - **Output:** All positions in *Genome* where *Pattern* appears as a substring.

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

Exact Pattern Matching

- Focus on a simple question: where do the reads match the reference genome *exactly*?

- **Multiple Pattern Matching Problem:**
  - **Input:** A collection of strings *Patterns* and a string *Genome.*
  - **Output:** All positions in *Genome* where a string from *Patterns* appears as a substring.

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

A Brute Force Approach

- We can simply iterate a brute force approach method, sliding each *Pattern* down *Genome*.

`panamabananas` *Genome*

Pattern

- **Note**: we use words instead of DNA strings for convenience.

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

A Brute Force Approach

- We can simply iterate a brute force approach method, sliding each *Pattern* down *Genome*.

`panamabananas`	*Genome*
`nana`	*Pattern*

- **Note**: we use words instead of DNA strings for convenience.

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

A Brute Force Approach

- We can simply iterate a brute force approach method, sliding each *Pattern* down *Genome*.

panamabananas	*Genome*
nana	*Pattern*

- **Note**: we use words instead of DNA strings for convenience.

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

A Brute Force Approach

- We can simply iterate a brute force approach method, sliding each *Pattern* down *Genome*.

```
panamabananas        Genome
 nana                Pattern
```

- **Note**: we use words instead of DNA strings for convenience.

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

A Brute Force Approach

- We can simply iterate a brute force approach method, sliding each *Pattern* down *Genome*.

`panamabananas`	*Genome*
`nana`	*Pattern*

- **Note**: we use words instead of DNA strings for convenience.

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

A Brute Force Approach

- We can simply iterate a brute force approach method, sliding each *Pattern* down *Genome*.

```
panamabananas        Genome
  nana               Pattern
```

- **Note**: we use words instead of DNA strings for convenience.

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

A Brute Force Approach

- We can simply iterate a brute force approach method, sliding each *Pattern* down *Genome*.

panamabananas	*Genome*
nana	*Pattern*

- **Note**: we use words instead of DNA strings for convenience.

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

A Brute Force Approach

- We can simply iterate a brute force approach method, sliding each *Pattern* down *Genome*.

panamabananas *Genome*

nana *Pattern*

- **Note**: we use words instead of DNA strings for convenience.

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

A Brute Force Approach

- We can simply iterate a brute force approach method, sliding each *Pattern* down *Genome*.

`panamabananas`	*Genome*
`nana`	*Pattern*

- **Note**: we use words instead of DNA strings for convenience.

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

A Brute Force Approach

- We can simply iterate a brute force approach method, sliding each *Pattern* down *Genome*.

panamabananas	*Genome*
nana	*Pattern*

- **Note**: we use words instead of DNA strings for convenience.

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

A Brute Force Approach

- We can simply iterate a brute force approach method, sliding each *Pattern* down *Genome*.

`panamabananas`	*Genome*
`nana`	*Pattern*

- **Note**: we use words instead of DNA strings for convenience.

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

A Brute Force Approach

- We can simply iterate a brute force approach method, sliding each *Pattern* down *Genome*.

panamabananas	*Genome*
nana	*Pattern*

- **Note**: we use words instead of DNA strings for convenience.

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

A Brute Force Approach

- We can simply iterate a brute force approach method, sliding each *Pattern* down *Genome.*

```
panamabananas        Genome
             nana    Pattern
```

- **Note**: we use words instead of DNA strings for convenience.

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

A Brute Force Approach

- We can simply iterate a brute force approach method, sliding each *Pattern* down *Genome.*

panamabananas	*Genome*
nana	*Pattern*

- **Note**: we use words instead of DNA strings for convenience.

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

A Brute Force Approach

- We can simply iterate a brute force approach method, sliding each *Pattern* down *Genome.*

panamaba**nan**as	*Genome*
nana	*Pattern*

- **Note**: we use words instead of DNA strings for convenience.

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

A Brute Force Approach

- We can simply iterate a brute force approach method, sliding each *Pattern* down *Genome.*

`panamabanana`s	*Genome*
`nana`	*Pattern*

- **Note**: we use words instead of DNA strings for convenience.

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

A Brute Force Approach

- We can simply iterate a brute force approach method, sliding each *Pattern* down *Genome*.

`panamabananas`	*Genome*
`nana`	*Pattern*

- **Note**: we use words instead of DNA strings for convenience.

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

That Was Easy!

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

Outline

- Why do we map reads?
- **Using the Trie**
- From a Trie to a Suffix Tree
- String Compression and the Burrows-Wheeler Transform
- Inverting Burrows-Wheeler
- Using Burrows-Wheeler for Pattern Matching
- Finding the Matched Patterns
- Setting Up Checkpoints
- Inexact Matching
- Further Applications of Read Mapping

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

Brute Force Is Too Slow

- The runtime of the brute force approach is too high!
  - Single *Pattern*: O(|*Genome*| * |*Pattern*|)
  - Multiple *Patterns*: O(|*Genome*| * |*Patterns*|)
  - |*Patterns*| = combined length of *Patterns*

In the case of the human genome (**3 GB**), the total length of all reads may exceed **1 TB**.

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

Patterns Travel One at a Time

Genome

CTGATGATGGACTACGCTACTACTGCTAGCTGTATTACGATCAGCTAC**C**ACATCGTAGCTAC

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

Packing Patterns onto a Bus

Genome

CTGATGATGGACTACGCTACTACTGCTAGCTGTATTACGATCAGCTAC**C**ACATCGTAGCTAC

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

Processing Patterns into a Trie

- Idea: combine reads into a graph. Each substring of the genome can match at most one read. So each read will correspond to a unique path through this graph.

- The resulting graph is called a **trie**.

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

Root

Patterns

banana
pan
nab
antenna
bandana
ananas
nana

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

Patterns

banana

pan

nab

antenna

bandana

ananas

nana

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

Patterns

banana

pan

nab

antenna

bandana

ananas

nana

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

Patterns

banana

pan

nab

antenna

bandana

ananas

nana

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

Patterns

banana

pan

nab

antenna

bandana

ananas

nana

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

Patterns

banana

pan

nab

antenna

bandana

ananas

nana

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

Patterns

banana

pan

nab

antenna

bandana

ananas

nana

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

Patterns

banana

pan

nab

antenna

bandana

ananas

nana

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

Patterns

banana

pan

and

nab

antenna

bandana

ananas

nana

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

Patterns

banana

pan

and

nab

antenna

bandana

ananas

nana

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

Patterns

banana

pan

and

nab

antenna

bandana

ananas

nana

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

Patterns

banana

pan

and

nab

antenna

bandana

ananas

nana

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

Patterns

banana

pan

and

nab

antenna

bandana

ananas

nana

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

Patterns

banana

pan

and

nab

antenna

bandana

ananas

nana

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

Patterns

banana

pan

and

nab

antenna

bandana

ananas

nana

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

Patterns

banana

pan

and

nab

antenna

bandana

ananas

nana

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

Patterns

banana

pan

and

nab

antenna

bandana

ananas

nana

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

Patterns

banana

pan

and

nab

antenna

bandana

ananas

nana

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

Patterns

banana

pan

and

nab

antenna

bandana

ananas

nana

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

Patterns

banana

pan

and

nab

antenna

bandana

ananas

nana

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

Patterns

banana

pan

and

nab

antenna

bandana

ananas

nana

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

Patterns

banana

pan

and

nab

antenna

bandana

ananas

nana

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

Patterns

banana

pan

and

nab

antenna

bandana

ananas

nana

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

Patterns

banana

pan

and

nab

antenna

bandana

ananas

nana

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

Patterns

banana

pan

and

nab

antenna

bandana

ananas

nana

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

Patterns

banana

pan

and

nab

antenna

bandana

ananas

nana

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

Patterns

banana

pan

and

nab

antenna

bandana

ananas

nana

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

Patterns

banana

pan

and

nab

antenna

bandana

ananas

nana

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

Patterns

banana

pan

and

nab

antenna

bandana

ananas

nana

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

Patterns

banana

pan

and

nab

antenna

bandana

ananas

nana

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

Patterns

banana

pan

and

nab

antenna

bandana

ananas

nana

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

Patterns

banana

pan

and

nab

antenna

bandana

ananas

nana

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

Patterns

banana

pan

and

nab

antenna

bandana

ananas

nana

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

Patterns

banana

pan

and

nab

antenna

bandana

ananas

nana

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

Patterns

banana

pan

and

nab

antenna

bandana

ananas

nana

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

Patterns

banana

pan

and

nab

antenna

bandana

ananas

nana

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

Patterns

banana

pan

and

nab

antenna

bandana

ananas

nana

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

Patterns

banana
pan
and
nab
antenna
bandana
ananas
nana

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

Patterns

banana
pan
and
nab
antenna
bandana
ananas
nana

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

Patterns

banana
pan
and
nab
antenna
bandana
ananas
nana

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

Patterns

banana
pan
and
nab
antenna
bandana
ananas
nana

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

Using the Trie for Pattern Matching

- **TrieMatching**: Slide the trie down the genome.

- At each position, walk down the trie and see if we can reach a leaf by matching symbols.

- Analogy: bus stops

Bioinformatics Algorithms: An Active Learning Approach
Copyright 2018 Compeau and Pevzner.

panamabananas

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

panamabananas

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

panamabananas

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

panamabananas

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

panamabananas

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

panamabananas

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

panamabananas

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

panamabananas

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

panamabananas

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

panamabananas

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

panamabananas

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

panamabananas

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

panamabananas

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

panamabananas

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

panamabananas

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

panamabananas

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

panamabananas

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

panamabananas

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

panamabananas

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

panamabananas

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

panama**banana**s

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

panama**banan**as

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

panama**banana**s

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

panamabananas

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

panamabananas

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

panamabananas

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

panamabananas

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

panamab**anana**s

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

panamab**ananas**

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

panamabananas

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

panamabananas

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

panamabanana s

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

panamabanana s

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

panamabanaanas

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

panamabanan as

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

panamabanana**s**

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

panamabanana**s**

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

panamabanan**a**s

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

panamabanan**as**

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

panamabanan**as**

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

panamabananas

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

panamabananas

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

panamabananas

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

Success!

- Runtime of Brute Force:
  - Total: O(|*Genome*|*|*Patterns*|)

- Runtime of Trie Matching:
  - Trie Construction: O(|*Patterns*|)
  - Pattern Matching: O(|*Genome*| * |*LongestPattern*|)

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

Outline

- Why do we map reads?
- Using the Trie
- **From a Trie to a Suffix Tree**
- String Compression and the Burrows-Wheeler Transform
- Inverting Burrows-Wheeler
- Using Burrows-Wheeler for Pattern Matching
- Finding the Matched Patterns
- Setting Up Checkpoints
- Inexact Matching
- Further Applications of Read Mapping

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

Memory Analysis of TrieMatching

- We completely forgot about memory!
- Our trie: 30 edges, |*Patterns*| = 39
- Worst case: # edges = O(|*Patterns*|)

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

Memory Analysis of TrieMatching

- We completely forgot about memory!
- Our trie: 30 edges, $|Patterns| = 39$
- Worst case: # edges = **$O(|Patterns|)$**

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

Preprocessing the Genome

- What if instead we create a data structure from the *genome itself*?

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

Preprocessing the Genome

- Split *Genome* into all its suffixes. (Show matching “banana” by finding the suffix “bananas”.)
- How can we combine these suffixes into a data structure?
- Let’s use a trie!

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

panamabananas$

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

panamabananas$

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

panamabananas$

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

panamabananas$

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

panamabananas$

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

panam**abananas$**

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

panama**bananas$**

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

panamab**ananas$**

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

panamaba**nanas$**

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

panamaban**anas$**

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

panamabana**nas$**

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

panamabanan**as$**

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

The Suffix Trie and Pattern Matching

- For each *Pattern*, see if *Pattern* can be spelled out from the root downward in the suffix trie.

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

banana

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

banana

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

banana

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

banana

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

banana

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

banana

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

nab

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

antenna

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

nana

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

nana

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

nana

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

nana

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

Where Are the Matches?

- To find where the pattern matches are, we need to add a little more information to the suffix trie.

- At each leaf ($), we add the starting position in *Genome* of the suffix ending at that leaf.

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

panamabananas$

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

panamabananas$

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

panamabananas$

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

panamabananas$

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

panamabananas$

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

panam**abananas$**

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

panama**bananas$**

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

panamab**ananas$**

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

panamabanas$

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

panamaban**anas$**

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

panamabana**nas$**

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

Where Are the Matches?

- Once we find a match, we “walk down” to the leaf in order to find the starting position of the match.

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

banana

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

banana

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

banana

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

panama**banana**s$

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

panamabananas$

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

Memory Trouble Once Again

- Worst case: the suffix trie holds O(|*Suffixes*|) nodes.

- For a *Genome* of length *n*, |*Suffixes*| = $n(n-1)/2 = O(n^2)$

Suffixes

panamabananas$
anamabananas$
namabananas$
amabananas$
mabananas$
abananas$
bananas$
ananas$
nanas$
anas$
nas$
as$
s$
$

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

Compressing the Trie

- This doesn't mean that our idea was bad!
- To reduce memory, we can compress each "nonbranching path" of the tree into an edge.

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

- This data structure is called a **suffix tree**.

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

- This data structure is called a **suffix tree**.

- For any *Genome*, # nodes < 2|*Genome*| = O(|*Genome*|).

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

- This data structure is called a **suffix tree**.

- For any *Genome*, # nodes < 2|*Genome*|.
  - **# leaves = |*Genome*|**;

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

- This data structure is called a **suffix tree**.

- For any *Genome*, # nodes < 2|*Genome*|.
  - **# leaves** = |*Genome*|;
  - **# internal nodes** < |*Genome*| – 1

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

Runtime and Memory Analysis

- Runtime:
  - $O(|Genome|^2)$ to construct the suffix tree.
  - $O(|Genome| + |Patterns|)$ to find pattern matches.

- Memory:
  - $O(|Genome|^2)$ to construct the suffix tree.
  - $O(|Genome|)$ to store the suffix tree.

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

Runtime and Memory Analysis

- Runtime:
  - O(|*Genome*|) to construct the suffix tree ***directly.***
  - O(|*Genome*| + |*Patterns*|) to find pattern matches.

- Memory:
  - O(|*Genome*|) to construct the suffix tree ***directly.***
  - O(|*Genome*|) to store the suffix tree.

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

Runtime and Memory Analysis

- Runtime:
  - O(|*Genome*|) to construct the suffix tree ***directly.***
  - O(|*Genome*| + |*Patterns*|) to find pattern matches.
  - **Total: O(|*Genome*| + |*Patterns*|)**

- Memory:
  - O(|*Genome*|) to construct the suffix tree ***directly.***
  - O(|*Genome*|) to store the suffix tree.
  - **Total: O(|*Genome*| + |*Patterns*|)**

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

Outline

- Why do we map reads?
- Using the Trie
- From a Trie to a Suffix Tree
- **String Compression and the Burrows-Wheeler Transform**
- Inverting Burrows-Wheeler
- Using Burrows-Wheeler for Pattern Matching
- Finding the Matched Patterns
- Setting Up Checkpoints
- Inexact Matching
- Further Applications of Read Mapping

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

We are Not Finished Yet

- We are happy with the suffix tree, but we are not completely satisfied.
  - Runtime: O(|*Genome*| + |*Patterns*|)
  - Memory: O(|*Genome*|)
- However, **big-O** notation ignores constants!
  - The best known suffix tree implementations require **~ 20 times** the length of |*Genome*|. (It would require **60 GB** of RAM for **3 GB** human genome)
  - Can we reduce this constant factor?

Suffix Arrays

In 1993, **Udi Manber** and **Gene Myers** introduced **suffix arrays** as a memory-efficient alternative to suffix trees.

To construct `SUFFIXARRAY(Text)`, we first **sort** all suffixes of Text **lexicographically**, assuming that "**$**" comes **first** in the alphabet.

Suffix Arrays

Sorted Suffixes	Starting Positions
$	13
abananas$	5
amabananas$	3
anamabananas$	1
ananas$	7
anas$	9
as$	11
bananas$	6
mabananas$	4
namabananas$	2
nanas$	8
nas$	10
panamabananas$	0
s$	12

SUFFIXARRAY("panamabananas$") = [13, 5, 3, 1, 7, 9, 11, 6, 4, 2, 8, 10, 0, 12]

```
PatternMatchingWithSuffixArray(Text, Pattern, SuffixArray)
    minIndex ← 0
    maxIndex ← |Text|
    while minIndex < maxIndex
        midIndex ← (minIndex + maxIndex) / 2
        if Pattern > suffix of Text starting at position SuffixArray(midIndex)
            minIndex ← midIndex + 1
        else
            maxIndex ← midIndex
    first ← minIndex
    maxIndex ← |Text|
    while minIndex < maxIndex
        midIndex ← (minIndex + maxIndex) / 2
        if Pattern < suffix of Text starting at position SuffixArray(midIndex)
            maxIndex ← midIndex
        else
            minIndex ← midIndex + 1
    last ← maxIndex
    if first > last
        return "Pattern does not appear in Text"
    else
        return (first, last)
```

Is it Possible to Improve More?

Suffix Array:

- **sorting** all suffixes of Text
- O(**|Text| · log (|Text|**))
- There exists algorithm using a fifth as much memory as suffix trees (**60GB** to **12GB** not bad)

Can we be so **ambitious** as to look for a data **structure** that would **encode** Text using **memory** approximately **equal** to the **length** of Text while still enabling **fast** pattern matching?

Genome Compression

- Idea: decrease the amount of memory required to hold *Genome*.

- This indicates that we need methods of **compressing** a large genome, which is seemingly a separate problem.

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

Idea #1: Run-Length Encoding

- **Run-length encoding:** compresses a run of *n* identical symbols.

Genome

GGGGGGGGGGCCCCCCCCCCCAAAAAAATTTTTTTTTTTTTTTTCCCCCG

↓

10G11C7A15T5C1G

Run-length encoding

- Problem: Genomes don't have lots of runs…

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

Converting Repeats to Runs

- …but they do have lots of repeats!

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

Converting Repeats to Runs

- …but they do have lots of repeats!

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

The **B**urrows-**W**heeler **T**ransform

panamabananas$

Form all cyclic rotations of
"panamabananas$"

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

The Burrows-Wheeler Transform

panamabananas$
$panamabananas

Form all cyclic rotations of
"panamabananas$"

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

The Burrows-Wheeler Transform

panamabananas$
$panamabananas
s$panamabanana

Form all cyclic rotations of
"panamabananas$"

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

The Burrows-Wheeler Transform

```
panamabananas$
$panamabananas
s$panamabanana
as$panamabanan
```

Form all cyclic rotations of
"panamabananas$"

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

The Burrows-Wheeler Transform

```
panamabananas$
$panamabananas
s$panamabanana
as$panamabanan
nas$panamabana
```

Form all cyclic rotations of
"panamabananas$"

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

The Burrows-Wheeler Transform

```
panamabananas$
$panamabananas
s$panamabanana
as$panamabanan
nas$panamabana
anas$panamaban
```

Form all cyclic rotations of
"panamabananas$"

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

The Burrows-Wheeler Transform

```
panamabananas$
$panamabananas
s$panamabanana
as$panamabanan
nas$panamabana
anas$panamaban
nanas$panamaba
```

Form all cyclic rotations of
"panamabananas$"

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

The Burrows-Wheeler Transform

```
panamabananas$
$panamabananas
s$panamabanana
as$panamabanan
nas$panamabana
anas$panamaban
nanas$panamaba
ananas$panamab
```

Form all cyclic rotations of
"panamabananas$"

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

The Burrows-Wheeler Transform

```
panamabananas$
$panamabananas
s$panamabanana
as$panamabanan
nas$panamabana
anas$panamaban
nanas$panamaba
ananas$panamab
bananas$panama
```

Form all cyclic rotations of
"panamabananas$"

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

The Burrows-Wheeler Transform

```
panamabananas$
$panamabananas
s$panamabanana
as$panamabanan
nas$panamabana
anas$panamaban
nanas$panamaba
ananas$panamab
bananas$panama
abananas$panam
```

Form all cyclic rotations of
"panamabananas$"

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

The Burrows-Wheeler Transform

```
panamabananas$
$panamabananas
s$panamabanana
as$panamabanan
nas$panamabana
anas$panamaban
nanas$panamaba
ananas$panamab
bananas$panama
abananas$panam
mabananas$pana
```

Form all cyclic rotations of
"panamabananas$"

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

The Burrows-Wheeler Transform

```
panamabananas$
$panamabananas
s$panamabanana
as$panamabanan
nas$panamabana
anas$panamaban
nanas$panamaba
ananas$panamab
bananas$panama
abananas$panam
mabananas$pana
amabananas$pan
```

Form all cyclic rotations of
"panamabananas$"

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

The Burrows-Wheeler Transform

```
panamabananas$
$panamabananas
s$panamabanana
as$panamabanan
nas$panamabana
anas$panamaban
nanas$panamaba
ananas$panamab
bananas$panama
abananas$panam
mabananas$pana
amabananas$pan
namabananas$pa
```

Form all cyclic rotations of
"panamabananas$"

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

The Burrows-Wheeler Transform

```
panamabananas$
$panamabananas
s$panamabanana
as$panamabanan
nas$panamabana
anas$panamaban
nanas$panamaba
ananas$panamab
bananas$panama
abananas$panam
mabananas$pana
amabananas$pan
namabananas$pa
anamabananas$p
```

Form all cyclic rotations of
"panamabananas$"

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

The Burrows-Wheeler Transform

```
panamabananas$
$panamabananas
s$panamabanana
as$panamabanan
nas$panamabana
anas$panamaban
nanas$panamaba
ananas$panamab
bananas$panama
abananas$panam
mabananas$pana
amabananas$pan
namabananas$pa
anamabananas$p
```

Form all cyclic rotations of
"panamabananas$"

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

The Burrows-Wheeler Transform

```
panamabananas$
$panamabananas
s$panamabanana
as$panamabanan
nas$panamabana
anas$panamaban
nanas$panamaba
ananas$panamab
bananas$panama
abananas$panam
mabananas$pana
amabananas$pan
namabananas$pa
anamabananas$p
```

Form all cyclic rotations of "panamabananas$"

→

```
$panamabananas
abananas$panam
amabananas$pan
anamabananas$p
ananas$panamab
anas$panamaban
as$panamabanan
bananas$panama
mabananas$pana
namabananas$pa
nanas$panamaba
nas$panamabana
panamabananas$
s$panamabanana
```

Sort the strings lexicographically ($ comes first)

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

The Burrows-Wheeler Transform

```
panamabananas$
$panamabananas
s$panamabanana
as$panamabanan
nas$panamabana
anas$panamaban
nanas$panamaba
ananas$panamab
bananas$panama
abananas$panam
mabananas$pana
amabananas$pan
namabananas$pa
anamabananas$p
```

Form all cyclic rotations of "panamabananas$"

→

```
$panamabananas
abananas$panam
amabananas$pan
anamabananas$p
ananas$panamab
anas$panamaban
as$panamabanan
bananas$panama
mabananas$pana
namabananas$pa
nanas$panamaba
nas$panamabana
panamabananas$
s$panamabanana
```

Sort the strings lexicographically ($ comes first)

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

The Burrows-Wheeler Transform

```
panamabananas$
$panamabananas
s$panamabanana
as$panamabanan
nas$panamabana
anas$panamaban
nanas$panamaba
ananas$panamab
bananas$panama
abananas$panam
mabananas$pana
amabananas$pan
namabananas$pa
anamabananas$p
```

→

```
$panamabananas
abananas$panam
amabananas$pan
anamabananas$p
ananas$panamab
anas$panamaban
as$panamabanan
bananas$panama
mabananas$pana
namabananas$pa
nanas$panamaba
nas$panamabana
panamabananas$
s$panamabanana
```

Form all cyclic rotations of "panamabananas$"

Burrows-Wheeler Transform:
Last column = smnpbnnaaaaa$a

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

Why Should We Care?

```
nd Corey (1).  They kindly made their manuscript availa ...... a
nd criticism, especially on interatomic distances.  We  ...... a
nd cytosine.  The sequence of bases on a single chain d ...... a
nd experimentally (3,4) that the ratio of the amounts o ...... u
nd for this reason we shall not comment on it.  We wish ...... a
nd guanine (purine) with cytosine (pyrimidine).  In oth ...... a
nd ideas of Dr.  M. H. F. Wilkins, Dr.  R. E. Franklin  ...... a
nd its water content is rather high.  At lower water co ...... a
nd pyrimidine bases.  The planes of the bases are perpe ...... a
nd stereochemical arguments.  It has not escaped our no ...... a
nd that only specific pairs of bases can bond together  ...... u
nd the atoms near it is close to Furberg's 'standard co ...... a
nd the bases on the inside, linked together by hydrogen ...... a
nd the bases on the outside.  In our opinion, this stru ...... a
nd the other a pyrimidine for bonding to occur.  The hy ...... a
nd the phosphates on the outside.  The configuration of ...... a
nd the ration of guanine to cytosine, are always very c ...... a
nd the same axis (see diagram).  We have made the usual ...... u
nd their co-workers at King's College, London.  One of  ...... a
```

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

BWT: Converting Repeats to Runs

Genome

Burrows-Wheeler Transform!

↓ Convert repeats to runs

BWT(*Genome*)

↓ Run-length encoding

Compression(*BWT*(*Genome*))

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

Outline

- Why do we map reads?
- Using the Trie
- From a Trie to a Suffix Tree
- String Compression and the Burrows-Wheeler Transform
- **Inverting Burrows-Wheeler**
- Using Burrows-Wheeler for Pattern Matching
- Finding the Matched Patterns
- Setting Up Checkpoints
- Inexact Matching
- Further Applications of Read Mapping

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

How Can We Decompress?

Genome

↓ Burrows-Wheeler Transform

BWT(*Genome*)

↓ Run-length encoding

Compression(*BWT*(*Genome*))

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

How Can We Decompress?

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

Reconstructing banana$

banana$

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

Reconstructing `banana$`

```
banana$
$banana
a$banan
na$bana
ana$ban
nana$ba
anana$b
```

Form all cyclic rotations of
"banana$"

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

Reconstructing banana$

banana$	$banana
$banana	a$banan
a$banan	ana$ban
na$bana →	anana$b
ana$ban	banana$
nana$ba	na$bana
anana$b	nana$ba

Form all cyclic rotations of "banana$"

Sort the strings lexicographically

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

Reconstructing banana

```
$banana
a$banan
ana$ban
anana$b
banana$
na$bana
nana$ba
```

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

Reconstructing `banana`

```
$banana
a$banan
ana$ban
anana$b
banana$
na$bana
nana$ba
```

- Sorting all elements of “annb$aa” gives first column of matrix.

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

Reconstructing `banana`

$banan**a**		a$
a$banan**n**		na
ana$ban		na
anana$**b**	→	ba
banana**$**	2-mers	$b
na$bana		an
nana$b**a**		an

- We now know 2-mer composition of the circular string `banana$`

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

Reconstructing banana

$banan**a**		a$		$b
a$bana**n**		na		a$
ana$ba**n**		na		an
anana$**b**	→ 2-mers	ba	→ Sort	an
banana**$**		$b		ba
na$ban**a**		an		na
nana$b**a**		an		na

- We now know 2-mer composition of the circular string banana$
- Sorting gives us the first 2 columns of the matrix.

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

Reconstructing `banana`

$banana		a$		$b
a$banan		na		a$
ana$ban		na		an
anana$b	→ 2-mers	ba	→ Sort	an
banana$		$b		ba
na$bana		an		na
nana$ba		an		na

- We now know 2-mer composition of the circular string `banana$`
- Sorting gives us the first 2 columns of the matrix.

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

Reconstructing banana

```
$banana
a$banan
ana$ban
anana$b
banana$
na$bana
nana$ba
```

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

Reconstructing `banana`

$banan**a**		a$b
a$banan**n**		na$
ana$ba**n**		nan
anana$**b**	→	ban
banana**$**	3-mers	$ba
na$ban**a**		ana
nana$b**a**		ana

- We now know 3-mer composition of the circular string `banana$`

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

Reconstructing `banana`

$banan**a**		a$b		$ba
a$bana**n**		na$		a$b
ana$ba**n**		nan		ana
anana$**b**	→	ban	→	ana
banana**$**	3-mers	$ba	Sort	ban
na$ban**a**		ana		na$
nana$b**a**		ana		nan

- We now know 3-mer composition of the circular string `banana$`
- Sorting gives us the first 3 columns of the matrix.

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

Reconstructing `banana`

$banan**a**		a$b		**$ba**
a$bana**n**		na$		**a$b**
ana$ba**n**		nan		**ana**
anana$**b**	→ 3-mers	ban	→ Sort	**ana**
banana**$**		$ba		**ban**
na$ban**a**		ana		**na$**
nana$b**a**		ana		**nan**

- We now know 3-mer composition of the circular string `banana$`
- Sorting gives us the first 3 columns of the matrix.

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

Reconstructing banana

```
$banana
a$banan
ana$ban
anana$b
banana$
na$bana
nana$ba
```

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

Reconstructing `banana`

$banan**a**		a$ba
a$banan**n**		na$b
ana$ban		nana
anana$**b**	→	bana
banana**$**	4-mers	$ban
na$ban**a**		ana$
nana$b**a**		anan

- We now know 4-mer composition of the circular string `banana$`

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

Reconstructing `banana`

$banan**a**		a$ba		$ban
a$banan**n**		na$b		a$bb
ana$ban		nana		anaa
anana$**b**	→	bana	→	anaa
banana**$**	4-mers	$ban	Sort	bann
na$ban**a**		ana$		na$b
nana$b**a**		anan		nana

- We now know 4-mer composition of the circular string `banana$`
- Sorting gives us the first 4 columns of the matrix.

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

Reconstructing `banana`

$banana		a$ba		**$ban**
a$banan		na$b		**a$bb**
ana$ban		nana		**anaa**
anana$b	→ 4-mers	bana	→ Sort	**anaa**
banana$		$ban		**bann**
na$bana		ana$		**na$b**
nana$ba		anan		**nana**

- We now know 4-mer composition of the circular string `banana$`
- Sorting gives us the first 4 columns of the matrix.

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

Reconstructing banana

```
$banana
a$banan
ana$ban
anana$b
banana$
na$bana
nana$ba
```

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

Reconstructing `banana`

$banan**a**		a$ban
a$banan**n**		na$ba
ana$ban		nana$
anana$**b**	→	banan
banana**$**	5-mers	$bana
na$ban**a**		ana$b
nana$b**a**		anana

- We now know 5-mer composition of the circular string `banana$`

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

Reconstructing `banana`

$banana		a$ban		$bana
a$banan		na$ba		a$bbn
ana$ban		nana$		anaab
anana$b	→	banan	→	anaaa
banana$	5-mers	$bana	Sort	bannn
na$bana		ana$b		na$ba
nana$ba		anana		nana$

- We now know 5-mer composition of the circular string `banana$`
- Sorting gives us the first 5 columns of the matrix.

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

Reconstructing banana

$banana		a$ban		$bana
a$banan		na$ba		a$bbn
ana$ban		nana$		anaab
anana$b	→ 5-mers	banan	→ Sort	anaaa
banana$		$bana		bannn
na$bana		ana$b		na$ba
nana$ba		anana		nana$

- We now know 5-mer composition of the circular string banana$
- Sorting gives us the first 5 columns of the matrix.

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

Reconstructing banana

```
$banana
a$banan
ana$ban
anana$b
banana$
na$bana
nana$ba
```

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

Reconstructing banana

$banana		a$bana
a$banan		na$ban
ana$ban		nana$b
anana$b	→	banana
banana$	6-mers	$banan
na$bana		ana$ba
nana$ba		anana$

- We now know 6-mer composition of the circular string banana$

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

Reconstructing `banana`

$banana		a$bana		$banan
a$banan		na$ban		a$bbna
ana$ban		nana$b		anaaba
anana$b	→	banana	→	anaaa$
banana$	6-mers	$banan	Sort	bannna
na$bana		ana$ba		na$ban
nana$ba		anana$		nana$b

- We now know 6-mer composition of the circular string `banana$`
- Sorting gives us the first 6 columns of the matrix.

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

Reconstructing `banana`

$banana		a$bana		**$banan**
a$banan		na$ban		**a$bbna**
ana$ban		nana$b		**anaaba**
anana$b	→	banana	→	**anaaa$**
banana$	6-mers	$banan	Sort	**bannna**
na$bana		ana$ba		**na$ban**
nana$ba		anana$		**nana$b**

- We now know 6-mer composition of the circular string `banana$`
- Sorting gives us the first 6 columns of the matrix.

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

Reconstructing `banana`

```
$banana
a$banan
ana$ban
anana$b
banana$
na$bana
nana$ba
```

- We now know the entire matrix!

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

Reconstructing `banana`

```
$banana
a$banan
ana$ban
anana$b
banana$
na$bana
nana$ba
```

- We now know the entire matrix!

- Taking all elements in the first row (after `$`) produces `banana`.

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

More Memory Issues

- Reconstructing *Genome* from *BWT*(*Genome*) required us to store |*Genome*| copies of |*Genome*|.

```
$banana
a$banan
ana$ban
anana$b
banana$
na$bana
nana$ba
```

- Can we invert BWT with less space?

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

A Strange Observation

```
$panamabananas
abananas$panam
amabananas$pan
anamabananas$p
ananas$panamab
anas$panamaban
as$panamabanan
bananas$panama
mabananas$pana
namabananas$pa
nanas$panamaba
nas$panamabana
panamabananas$
s$panamabanana
```

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

A Strange Observation

```
$panamabananas
abananas$panam
amabananas$pan
anamabananas$p
ananas$panamab
anas$panamaban
as$panamabanan
bananas$panama
mabananas$pana
namabananas$pa
nanas$panamaba
nas$panamabana
panamabananas$
s$panamabanana
```

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

A Strange Observation

```
$panamabananas
abananas$panam
amabananas$pan
anamabananas$p
ananas$panamab
anas$panamaban
as$panamabanan
bananas$panama
mabananas$pana
namabananas$pa
nanas$panamaba
nas$panamabana
panamabananas$
s$panamabanana
```

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

A Strange Observation

```
$panamabananas
abananas$panam
amabananas$pan
anamabananas$p
ananas$panamab
anas$panamaban
as$panamabanan
bananas$panama
mabananas$pana
namabananas$pa
nanas$panamaba
nas$panamabana
panamabananas$
s$panamabanana
```

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

A Strange Observation

```
$panamabananas
abananas$panam
amabananas$pan
anamabananas$p
ananas$panamab
anas$panamaban
as$panamabanan
bananas$panama
mabananas$pana
namabananas$pa
nanas$panamaba
nas$panamabana
panamabananas$
s$panamabanana
```

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

A Strange Observation

```
$panamabananas
abananas$panam
amabananas$pan
anamabananas$p
ananas$panamab
anas$panamaban
as$panamabanan
bananas$panama
mabananas$pana
namabananas$pa
nanas$panamaba
nas$panamabana
panamabananas$
s$panamabanana
```

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

A Strange Observation

```
$panamabananas
abananas$panam
amabananas$pan
anamabananas$p
ananas$panamab
anas$panamaban
as$panamabanan
bananas$panama
mabananas$pana
namabananas$pa
nanas$panamaba
nas$panamabana
panamabananas$
s$panamabanana
```

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

A Strange Observation

```
$panamabananas
abananas$panam
amabananas$pan
anamabananas$p
ananas$panamab
anas$panamaban
as$panamabanan
bananas$panama
mabananas$pana
namabananas$pa
nanas$panamaba
nas$panamabana
panamabananas$
s$panamabanana
```

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

A Strange Observation

```
$panamabananas
abananas$panam
amabananas$pan
anamabananas$p
ananas$panamab
anas$panamaban
as$panamabanan
bananas$panama
mabananas$pana
namabananas$pa
nanas$panamaba
nas$panamabana
panamabananas$
s$panamabanana
```

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

Is It True in General?

```
  $panamabananas
1 abananas$panam
2 amabananas$pan
3 anamabananas$p
4 ananas$panamab
5 anas$panamaban
6 as$panamabanan
  bananas$panama
  mabananas$pana
  namabananas$pa
  nanas$panamaba
  nas$panamabana
  panamabananas$
  s$panamabanana
```

These strings are sorted

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

Is It True in General?

```
  $panamabananas
1 abananas$panam
2 amabananas$pan
3 anamabananas$p
4 ananas$panamab
5 anas$panamaban
6 as$panamabanan
  bananas$panama
  mabananas$pana
  namabananas$pa
  nanas$panamaba
  nas$panamabana
  panamabananas$
  s$panamabanana
```

Chop off **a**

```
bananas$panam
mabananas$pan
namabananas$p
nanas$panamab
nas$panamaban
s$panamabanan
```

These strings are sorted

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

Is It True in General?

```
  $panamabananas
1 abananas$panam
2 amabananas$pan
3 anamabananas$p
4 ananas$panamab
5 anas$panamaban
6 as$panamabanan
  bananas$panama
  mabananas$pana
  namabananas$pa
  nanas$panamaba
  nas$panamabana
  panamabananas$
  s$panamabanana
```

These strings are sorted

Chop off **a**

```
bananas$panam
mabananas$pan
namabananas$p
nanas$panamab
nas$panamaban
s$panamabanan
```

Still sorted

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

Is It True in General?

```
  $panamabananas
1 abananas$panam
2 amabananas$pan
3 anamabananas$p
4 ananas$panamab
5 anas$panamaban
6 as$panamabanan
  bananas$panama
  mabananas$pana
  namabananas$pa
  nanas$panamaba
  nas$panamabana
  panamabananas$
  s$panamabanana
```

These strings are sorted

Chop off **a**

```
bananas$panam
mabananas$pan
namabananas$p
nanas$panamab
nas$panamaban
s$panamabanan
```

Still sorted

Add **a** to end

```
bananas$panama
mabananas$pana
namabananas$pa
nanas$panamaba
nas$panamabana
s$panamabanana
```

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

Is It True in General?

```
  $panamabananas
1 abananas$panam
2 amabananas$pan
3 anamabananas$p
4 ananas$panamab
5 anas$panamaban
6 as$panamabanan
  bananas$panama
  mabananas$pana
  namabananas$pa
  nanas$panamaba
  nas$panamabana
  panamabananas$
  s$panamabanana
```

These strings are sorted

Chop off **a**

```
bananas$panam
mabananas$pan
namabananas$p
nanas$panamab
nas$panamaban
s$panamabanan
```

Still sorted

Add **a** to end

```
bananas$panama
mabananas$pana
namabananas$pa
nanas$panamaba
nas$panamabana
s$panamabanana
```

Still sorted

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

Is It True in General?

```
  $panamabananas
1 abananas$panam
2 amabananas$pan
3 anamabananas$p
4 ananas$panamab
5 anas$panamaban
6 as$panamabanan
```

Chop off **a**

```
bananas$panam
mabananas$pan
namabananas$p
nanas$panamab
nas$panamaban
s$panamabanan
```

Still sorted

Add **a** to end

```
bananas$panama
mabananas$pana
namabananas$pa
nanas$panamaba
nas$panamabana
s$panamabanana
```

Still sorted

Ordering doesn't change!

```
bananas$panama 1
mabananas$pana 2
namabananas$pa 3
nanas$panamaba 4
nas$panamabana 5
panamabananas$
s$panamabanana 6
```

These strings are sorted

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

Is It True in General?

- **First-Last Property**: The *k*-th occurrence of *symbol* in *FirstColumn* and the *k*-th occurrence of *symbol* in *LastColumn* correspond to the same position of *symbol* in *Genome*.

$_1$panamabananas$_1$
a$_1$bananas$panam$_1$
a$_2$mabananaspan_1$
a$_3$namabananas$p$_1$
a$_4$nanas$panamab$_1$
a$_5$nas$panamaban$_2$
a$_6$s$panamabanan$_3$
b$_1$ananas$panama$_1$
m$_1$abananas$pana$_2$
n$_1$amabananaspa_3$
n$_2$anas$panamaba$_4$
n$_3$as$panamabana$_5$
p$_1$anamabananas$$_1$
s$_1$$panamabanana$_6$

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

More Efficient BWT Decompression

$_1$panamabananas$_1$
a$_1$bananas$panam$_1$
a$_2$mabananaspan_1$
a$_3$namabananas$p$_1$
a$_4$nanas$panamab$_1$
a$_5$nas$panamaban$_2$
a$_6$s$panamabanan$_3$
b$_1$ananas$panama$_1$
m$_1$abananas$pana$_2$
n$_1$amabananaspa_3$
n$_2$anas$panamaba$_4$
n$_3$as$panamabana$_5$
p$_1$anamabananas$$_1$
s$_1$$panamabanana$_6$

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

More Efficient BWT Decompression

$_1$panamabananas$_1$
a$_1$bananas$panam$_1$
a$_2$mabananaspan_1$
a$_3$namabananas$p$_1$
a$_4$nanas$panamab$_1$
a$_5$nas$panamaban$_2$
a$_6$s$panamabanan$_3$
b$_1$ananas$panama$_1$
m$_1$abananas$pana$_2$
n$_1$amabananaspa_3$
n$_2$anas$panamaba$_4$
n$_3$as$panamabana$_5$
p$_1$anamabananas$$_1$
s$_1$$panamabanana$_6$

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

More Efficient BWT Decompression

$_1$panamabananas_1
$a_1$bananas$panam$_1$
$a_2$mabananaspan_1$
$a_3$namabananas$p$_1$
$a_4$nanas$panamab$_1$
$a_5$nas$panamaban$_2$
a_6s$panamabanan$_3$
$b_1$ananas$panama$_1$
$m_1$abananas$pana$_2$
$n_1$amabananaspa_3$
$n_2$anas$panamaba$_4$
$n_3$as$panamabana$_5$
$p_1$anamabananas$$_1$
$s_1$$panamabanana$_6$

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

More Efficient BWT Decompression

$_1$panamabanana**s**$_1$
a$_1$bananas$panam$_1$
a$_2$mabananaspan_1$
a$_3$namabananas$p$_1$
a$_4$nanas$panamab$_1$
a$_5$nas$panamaban$_2$
a$_6$s$panamabanan$_3$
b$_1$ananas$panama$_1$
m$_1$abananas$pana$_2$
n$_1$amabananaspa_3$
n$_2$anas$panamaba$_4$
n$_3$as$panamabana$_5$
p$_1$anamabananas$$_1$
s$_1$$panamabanana$_6$

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

More Efficient BWT Decompression

$_1$panamabananas$_1$
a$_1$bananas$panam$_1$
a$_2$mabananaspan_1$
a$_3$namabananas$p$_1$
a$_4$nanas$panamab$_1$
a$_5$nas$panamaban$_2$
a$_6$s$panamabanan$_3$
b$_1$ananas$panama$_1$
m$_1$abananas$pana$_2$
n$_1$amabananaspa_3$
n$_2$anas$panamaba$_4$
n$_3$as$panamabana$_5$
p$_1$anamabananas$$_1$
s$_1$ $panamabanan**a$_6$**

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

More Efficient BWT Decompression

$_1$panamabananas$_1$
a$_1$bananas$panam$_1$
a$_2$mabananaspan_1$
a$_3$namabananas$p$_1$
a$_4$nanas$panamab$_1$
a$_5$nas$panamaban$_2$
a$_6$s$panamabanan$_3$
b$_1$ananas$panama$_1$
m$_1$abananas$pana$_2$
n$_1$amabananaspa_3$
n$_2$anas$panamaba$_4$
n$_3$as$panamabana$_5$
p$_1$anamabananas$$_1$
s$_1$$panamabanan**a$_6$**

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

More Efficient BWT Decompression

$_1$panamabananas$_1$
a$_1$bananas$panam$_1$
a$_2$mabananaspan_1$
a$_3$namabananas$p$_1$
a$_4$nanas$panamab$_1$
a$_5$nas$panamaban$_2$
a$_6$s$panamabanan$_3$**
b$_1$ananas$panama$_1$
m$_1$abananas$pana$_2$
n$_1$amabananaspa_3$
n$_2$anas$panamaba$_4$
n$_3$as$panamabana$_5$
p$_1$anamabananas$$_1$
s$_1$$panamabanana$_6$

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

More Efficient BWT Decompression

$_1$panamabananas$_1$
a$_1$bananas$panam$_1$
a$_2$mabananaspan_1$
a$_3$namabananas$p$_1$
a$_4$nanas$panamab$_1$
a$_5$nas$panamaban$_2$
a$_6$s$panamabanan$_3$
b$_1$ananas$panama$_1$
m$_1$abananas$pana$_2$
n$_1$amabananaspa_3$
n$_2$anas$panamaba$_4$
n$_3$as$panamabana$_5$
p$_1$anamabananas$$_1$
s$_1$$panamabanana$_6$

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

More Efficient BWT Decompression

$_1$panamabananas$_1$
a$_1$bananas$panam$_1$
a$_2$mabananaspan_1$
a$_3$namabananas$p$_1$
a$_4$nanas$panamab$_1$
a$_5$nas$panamaban$_2$
a$_6$s$panamabanan$_3$
b$_1$ananas$panama$_1$
m$_1$abananas$pana$_2$
n$_1$amabananaspa_3$
n$_2$anas$panamaba$_4$
n$_3$as$panamaban**a$_5$**
p$_1$anamabananas$$_1$
s$_1$$panamabanana$_6$

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

More Efficient BWT Decompression

$_1$panamabananas$_1$
a$_1$bananas$panam$_1$
a$_2$mabananaspan_1$
a$_3$namabananas$p$_1$
a$_4$nanas$panamab$_1$
a$_5$nas$panamaban$_2$
a$_6$s$panamabanan$_3$
b$_1$ananas$panama$_1$
m$_1$abananas$pana$_2$
n$_1$amabananaspa_3$
n$_2$anas$panamaba$_4$
n$_3$as$panamaban**a$_5$**
p$_1$anamabananas$$_1$
s$_1$$panamabanana$_6$

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

More Efficient BWT Decompression

$_1$panamabananas$_1$
a$_1$bananas$panam$_1$
a$_2$mabananaspan_1$
a$_3$namabananas$p$_1$
a$_4$nanas$panamab$_1$
a$_5$nas$panamaba**n$_2$**
a$_6$s$panamabanan$_3$
b$_1$ananas$panama$_1$
m$_1$abananas$pana$_2$
n$_1$amabananaspa_3$
n$_2$anas$panamaba$_4$
n$_3$as$panamabana$_5$
p$_1$anamabananas$$_1$
s$_1$$panamabanana$_6$

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

More Efficient BWT Decompression

$_1$panamabananas$_1$
a$_1$bananas$panam$_1$
a$_2$mabananaspan_1$
a$_3$namabananas$p$_1$
a$_4$nanas$panamab$_1$
a$_5$nas$panamaba**n$_2$**
a$_6$s$panamabanan$_3$
b$_1$ananas$panama$_1$
m$_1$abananas$pana$_2$
n$_1$amabananaspa_3$
n$_2$anas$panamaba$_4$
n$_3$as$panamabana$_5$
p$_1$anamabananas$$_1$
s$_1$$panamabanana$_6$

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

More Efficient BWT Decompression

$_1$panamabananas$_1$
a$_1$bananas$panam$_1$
a$_2$mabananaspan_1$
a$_3$namabananas$p$_1$
a$_4$nanas$panamab$_1$
a$_5$nas$panamaban$_2$
a$_6$s$panamabanan$_3$
b$_1$ananas$panama$_1$
m$_1$abananas$pana$_2$
n$_1$amabananaspa_3$
n$_2$anas$panamab**a$_4$**
n$_3$as$panamabana$_5$
p$_1$anamabananas$$_1$
s$_1$$panamabanana$_6$

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

More Efficient BWT Decompression

$_1$panamabananas$_1$
a$_1$bananas$panam$_1$
a$_2$mabananaspan_1$
a$_3$namabananas$p$_1$
a$_4$nanas$panamab$_1$
a$_5$nas$panamaban$_2$
a$_6$s$panamabanan$_3$
b$_1$ananas$panama$_1$
m$_1$abananas$pana$_2$
n$_1$amabananaspa_3$
n$_2$anas$panamab**a$_4$**
n$_3$as$panamabana$_5$
p$_1$anamabananas$$_1$
s$_1$$panamabanana$_6$

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

More Efficient BWT Decompression

$_1$panamabananas$_1$
a$_1$bananas$panam$_1$
a$_2$mabananaspan_1$
a$_3$namabananas$p$_1$
a$_4$nanas$panama**b$_1$**
a$_5$nas$panamaban$_2$
a$_6$s$panamabanan$_3$
b$_1$ananas$panama$_1$
m$_1$abananas$pana$_2$
n$_1$amabananaspa_3$
n$_2$anas$panamaba$_4$
n$_3$as$panamabana$_5$
p$_1$anamabananas$$_1$
s$_1$$panamabanana$_6$

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

More Efficient BWT Decompression

$_1$panamabananas$_1$
a$_1$bananas$panam$_1$
a$_2$mabananaspan_1$
a$_3$namabananas$p$_1$
a$_4$nanas$panama**b$_1$**
a$_5$nas$panamaban$_2$
a$_6$s$panamabanan$_3$
b$_1$ananas$panama$_1$
m$_1$abananas$pana$_2$
n$_1$amabananaspa_3$
n$_2$anas$panamaba$_4$
n$_3$as$panamabana$_5$
p$_1$anamabananas$$_1$
s$_1$$panamabanana$_6$

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

More Efficient BWT Decompression

$_1$panamabananas$_1$
a$_1$bananas$panam$_1$
a$_2$mabananaspan_1$
a$_3$namabananas$p$_1$
a$_4$nanas$panamab$_1$
a$_5$nas$panamaban$_2$
a$_6$s$panamabanan$_3$
b$_1$ananas$panam**a$_1$**
m$_1$abananas$pana$_2$
n$_1$amabananaspa_3$
n$_2$anas$panamaba$_4$
n$_3$as$panamabana$_5$
p$_1$anamabananas$$_1$
s$_1$$panamabanana$_6$

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

More Efficient BWT Decompression

$_1$panamabananas$_1$
a$_1$bananas$panam$_1$
a$_2$mabananaspan_1$
a$_3$namabananas$p$_1$
a$_4$nanas$panamab$_1$
a$_5$nas$panamaban$_2$
a$_6$s$panamabanan$_3$
b$_1$ananas$panam**a$_1$**
m$_1$abananas$pana$_2$
n$_1$amabananaspa_3$
n$_2$anas$panamaba$_4$
n$_3$as$panamabana$_5$
p$_1$anamabananas$$_1$
s$_1$$panamabanana$_6$

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

More Efficient BWT Decompression

$_1$panamabananas$_1$
a$_1$bananas$panam**m$_1$**
a$_2$mabananaspan_1$
a$_3$namabananas$p$_1$
a$_4$nanas$panamab$_1$
a$_5$nas$panamaban$_2$
a$_6$s$panamabanan$_3$
b$_1$ananas$panama$_1$
m$_1$abananas$pana$_2$
n$_1$amabananaspa_3$
n$_2$anas$panamaba$_4$
n$_3$as$panamabana$_5$
p$_1$anamabananas$$_1$
s$_1$$panamabanana$_6$

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

More Efficient BWT Decompression

$_1$panamabananas$_1$
a$_1$bananas$panam$_1$
a$_2$mabananaspan_1$
a$_3$namabananas$p$_1$
a$_4$nanas$panamab$_1$
a$_5$nas$panamaban$_2$
a$_6$s$panamabanan$_3$
b$_1$ananas$panama$_1$
m$_1$abananas$pana$_2$
n$_1$amabananaspa_3$
n$_2$anas$panamaba$_4$
n$_3$as$panamabana$_5$
p$_1$anamabananas$$_1$
s$_1$$panamabanana$_6$

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

More Efficient BWT Decompression

$_1$panamabananas$_1$
a$_1$bananas$panam$_1$
a$_2$mabananaspan_1$
a$_3$namabananas$p$_1$
a$_4$nanas$panamab$_1$
a$_5$nas$panamaban$_2$
a$_6$s$panamabanan$_3$
b$_1$ananas$panama$_1$
m$_1$abananas$pan**a$_2$**
n$_1$amabananaspa_3$
n$_2$anas$panamaba$_4$
n$_3$as$panamabana$_5$
p$_1$anamabananas$$_1$
s$_1$$panamabanana$_6$

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

More Efficient BWT Decompression

$_1$panamabananas$_1$
a$_1$bananas$panam$_1$
a$_2$mabananaspan_1$
a$_3$namabananas$p$_1$
a$_4$nanas$panamab$_1$
a$_5$nas$panamaban$_2$
a$_6$s$panamabanan$_3$
b$_1$ananas$panama$_1$
m$_1$abananas$pan**a$_2$**
n$_1$amabananaspa_3$
n$_2$anas$panamaba$_4$
n$_3$as$panamabana$_5$
p$_1$anamabananas$$_1$
s$_1$$panamabanana$_6$

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

More Efficient BWT Decompression

$_1$panamabananas$_1$
a$_1$bananas$panam$_1$
a$_2$mabananas$pa**n$_1$**
a$_3$namabananas$p$_1$
a$_4$nanas$panamab$_1$
a$_5$nas$panamaban$_2$
a$_6$s$panamabanan$_3$
b$_1$ananas$panama$_1$
m$_1$abananas$pana$_2$
n$_1$amabananaspa_3$
n$_2$anas$panamaba$_4$
n$_3$as$panamabana$_5$
p$_1$anamabananas$$_1$
s$_1$$panamabanana$_6$

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

More Efficient BWT Decompression

$_1$panamabananas$_1$
a$_1$bananas$panam$_1$
a$_2$mabananas$pa**n$_1$**
a$_3$namabananas$p$_1$
a$_4$nanas$panamab$_1$
a$_5$nas$panamaban$_2$
a$_6$s$panamabanan$_3$
b$_1$ananas$panama$_1$
m$_1$abananas$pana$_2$
n$_1$amabananaspa_3$
n$_2$anas$panamaba$_4$
n$_3$as$panamabana$_5$
p$_1$anamabananas$$_1$
s$_1$$panamabanana$_6$

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

More Efficient BWT Decompression

$_1$panamabananas$_1$
a$_1$bananas$panam$_1$
a$_2$mabananaspan_1$
a$_3$namabananas$p$_1$
a$_4$nanas$panamab$_1$
a$_5$nas$panamaban$_2$
a$_6$s$panamabanan$_3$
b$_1$ananas$panama$_1$
m$_1$abananas$pana$_2$
n$_1$amabananas$p**a$_3$**
n$_2$anas$panamaba$_4$
n$_3$as$panamabana$_5$
p$_1$anamabananas$$_1$
s$_1$$panamabanana$_6$

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

More Efficient BWT Decompression

$_1$panamabananas$_1$
a$_1$bananas$panam$_1$
a$_2$mabananaspan_1$
a$_3$namabananas$p$_1$
a$_4$nanas$panamab$_1$
a$_5$nas$panamaban$_2$
a$_6$s$panamabanan$_3$
b$_1$ananas$panama$_1$
m$_1$abananas$pana$_2$
n$_1$amabananas$p**a$_3$**
n$_2$anas$panamaba$_4$
n$_3$as$panamabana$_5$
p$_1$anamabananas$$_1$
s$_1$$panamabanana$_6$

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

More Efficient BWT Decompression

$_1$panamabananas$_1$
a$_1$bananas$panam$_1$
a$_2$mabananaspan_1$
a$_3$namabananas$**p$_1$**
a$_4$nanas$panamab$_1$
a$_5$nas$panamaban$_2$
a$_6$s$panamabanan$_3$
b$_1$ananas$panama$_1$
m$_1$abananas$pana$_2$
n$_1$amabananaspa_3$
n$_2$anas$panamaba$_4$
n$_3$as$panamabana$_5$
p$_1$anamabananas$$_1$
s$_1$$panamabanana$_6$

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

More Efficient BWT Decompression

$_1$panamabananas$_1$
a$_1$bananas$panam$_1$
a$_2$mabananaspan_1$
a$_3$namabananas$**p$_1$**
a$_4$nanas$panamab$_1$
a$_5$nas$panamaban$_2$
a$_6$s$panamabanan$_3$
b$_1$ananas$panama$_1$
m$_1$abananas$pana$_2$
n$_1$amabananaspa_3$
n$_2$anas$panamaba$_4$
n$_3$as$panamabana$_5$
p$_1$anamabananas$$_1$
s$_1$$panamabanana$_6$

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

More Efficient BWT Decompression

$_1$panamabananas$_1$
a$_1$bananas$panam$_1$
a$_2$mabananaspan_1$
a$_3$namabananas$p$_1$
a$_4$nanas$panamab$_1$
a$_5$nas$panamaban$_2$
a$_6$s$panamabanan$_3$
b$_1$ananas$panama$_1$
m$_1$abananas$pana$_2$
n$_1$amabananaspa_3$
n$_2$anas$panamaba$_4$
n$_3$as$panamabana$_5$
p$_1$anamabananas**$$_1$**
s$_1$$panamabanana$_6$

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

More Efficient BWT Decompression

$_1$panamabananas$_1$
a$_1$bananas$panam$_1$
a$_2$mabananaspan_1$
a$_3$namabananas$p$_1$
a$_4$nanas$panamab$_1$
a$_5$nas$panamaban$_2$
a$_6$s$panamabanan$_3$
b$_1$ananas$panama$_1$
m$_1$abananas$pana$_2$
n$_1$amabananaspa_3$
n$_2$anas$panamaba$_4$
n$_3$as$panamabana$_5$
p$_1$anamabananas$$_1$
s$_1$$panamabanana$_6$

- Memory: 2|*Genome*| = O(|*Genome*|).

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

But Why Should We Care?

- What happened to pattern matching?

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

Outline

- Why do we map reads?
- Using the Trie
- From a Trie to a Suffix Tree
- String Compression and the Burrows-Wheeler Transform
- Inverting Burrows-Wheeler
- **Using Burrows-Wheeler for Pattern Matching**
- Finding the Matched Patterns
- Setting Up Checkpoints
- Inexact Matching
- Further Applications of Read Mapping

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

Recalling Our Goal

- Suffix Tree Pattern Matching:
  - Runtime: O(|*Genome*| + |*Patterns*|)
  - Memory: O(|*Genome*|)
  - Problem: suffix tree takes 20 x |*Genome*| space

- Can we use BWT(*Genome*) as our data structure instead?

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

Finding Pattern Matches Using BWT

- Searching for **ana** in p**ana**mab**anana**s

$_1$panamabananas$_1$
a$_1$bananas$panam$_1$
a$_2$mabananaspan_1$
a$_3$namabananas$p$_1$
a$_4$nanas$panamab$_1$
a$_5$nas$panamaban$_2$
a$_6$s$panamabanan$_3$
b$_1$ananas$panama$_1$
m$_1$abananas$pana$_2$
n$_1$amabananaspa_3$
n$_2$anas$panamaba$_4$
n$_3$as$panamabana$_5$
p$_1$anamabananas$$_1$
s$_1$$panamabanana$_6$

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

Finding Pattern Matches Using BWT

- Searching for `ana` in `panamabananas`

$_1$panamabananas$_1$
a$_1$bananas$panam$_1$
a$_2$mabananaspan_1$
a$_3$namabananas$p$_1$
a$_4$nanas$panamab$_1$
a$_5$nas$panamaban$_2$
a$_6$s$panamabanan$_3$
b$_1$ananas$panama$_1$
m$_1$abananas$pana$_2$
n$_1$amabananaspa_3$
n$_2$anas$panamaba$_4$
n$_3$as$panamabana$_5$
p$_1$anamabananas$$_1$
s$_1$$panamabanana$_6$

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

Finding Pattern Matches Using BWT

- Searching for `ana` in `panamabananas`

$_1$panamabananas$_1$
a$_1$bananas$panam$_1$
a$_2$mabananaspan_1$
a$_3$namabananas$p$_1$
a$_4$nanas$panamab$_1$
a$_5$nas$panamaban$_2$
a$_6$s$panamabanan$_3$
b$_1$ananas$panama$_1$
m$_1$abananas$pana$_2$
n$_1$amabananaspa_3$
n$_2$anas$panamaba$_4$
n$_3$as$panamabana$_5$
p$_1$anamabananas$$_1$
s$_1$$panamabanana$_6$

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

Finding Pattern Matches Using BWT

- Searching for **ana** in panamabananas

$_1$panamabananas$_1$
a$_1$bananas$panam**m**$_1$
a$_2$mabananas$pa**n**$_1$
a$_3$namabananas$**p**$_1$
a$_4$nanas$panama**b**$_1$
a$_5$nas$panamaba**n**$_2$
a$_6$s$panamabana**n**$_3$
b$_1$ananas$panama$_1$
m$_1$abananas$pana$_2$
n$_1$amabananaspa_3$
n$_2$anas$panamaba$_4$
n$_3$as$panamabana$_5$
p$_1$anamabananas$$_1$
s$_1$$panamabanana$_6$

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

Finding Pattern Matches Using BWT

- Searching for a**na** in panamabananas

$_1$panamabananas$_1$
a$_1$bananas$panam$_1$
a$_2$mabananas$pa**n$_1$**
a$_3$namabananas$p$_1$
a$_4$nanas$panamab$_1$
a$_5$nas$panamaba**n$_2$**
a$_6$s$panamabana**n$_3$**
b$_1$ananas$panama$_1$
m$_1$abananas$pana$_2$
n$_1$amabananaspa_3$
n$_2$anas$panamaba$_4$
n$_3$as$panamabana$_5$
p$_1$anamabananas$$_1$
s$_1$$panamabanana$_6$

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

Finding Pattern Matches Using BWT

- Searching for a**na** in panamabananas

$_1$panamabananas$_1$
a$_1$bananas$panam$_1$
a$_2$mabananas$pa**n**$_1$
a$_3$namabananas$p$_1$
a$_4$nanas$panamab$_1$
a$_5$nas$panamaba**n**$_2$
a$_6$s$panamabana**n**$_3$
b$_1$ananas$panama$_1$
m$_1$abananas$pana$_2$
n$_1$amabananaspa_3$
n$_2$anas$panamaba$_4$
n$_3$as$panamabana$_5$
p$_1$anamabananas$$_1$
s$_1$$panamabanana$_6$

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

Finding Pattern Matches Using BWT

- Searching for a**na** in panamabananas

$_1$panamabananas$_1$
a$_1$bananas$panam$_1$
a$_2$mabananaspan_1$
a$_3$namabananas$p$_1$
a$_4$nanas$panamab$_1$
a$_5$nas$panamaban$_2$
a$_6$s$panamabanan$_3$
b$_1$ananas$panama$_1$
m$_1$abananas$pana$_2$
n$_1$amabananaspa_3$
n$_2$anas$panamaba$_4$
n$_3$as$panamabana$_5$
p$_1$anamabananas$$_1$
s$_1$$panamabanana$_6$

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

Finding Pattern Matches Using BWT

- Searching for **ana** in panamabananas

$_1$panamabananas$_1$
a$_1$bananas$panam$_1$
a$_2$mabananaspan_1$
a$_3$namabananas$p$_1$
a$_4$nanas$panamab$_1$
a$_5$nas$panamaban$_2$
a$_6$s$panamabanan$_3$
b$_1$ananas$panama$_1$
m$_1$abananas$pana$_2$
n$_1$amabananas$p**a$_3$**
n$_2$anas$panamab**a$_4$**
n$_3$as$panamaban**a$_5$**
p$_1$anamabananas$$_1$
s$_1$$panamabanana$_6$

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

Finding Pattern Matches Using BWT

- Searching for **ana** in panamabananas

$_1$panamabananas$_1$
a$_1$bananas$panam$_1$
a$_2$mabananaspan_1$
a$_3$namabananas$p$_1$
a$_4$nanas$panamab$_1$
a$_5$nas$panamaban$_2$
a$_6$s$panamabanan$_3$
b$_1$ananas$panama$_1$
m$_1$abananas$pana$_2$
n$_1$amabananaspa_3$
n$_2$anas$panamaba$_4$
n$_3$as$panamabana$_5$
p$_1$anamabananas$$_1$
s$_1$$panamabanana$_6$

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

Outline

- Why do we map reads?
- Using the Trie
- From a Trie to a Suffix Tree
- String Compression and the Burrows-Wheeler Transform
- Inverting Burrows-Wheeler
- Using Burrows-Wheeler for Pattern Matching
- **Finding the Matched Patterns**
- Setting Up Checkpoints
- Inexact Matching
- Further Applications of Read Mapping

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

Where Are the Matches?

- **Multiple Pattern Matching Problem:**
  - **Input:** A collection of strings *Patterns* and a string *Genome.*
  - **Output:** All positions in *Genome* where one of *Patterns* appears as a substring.

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

Where Are the Matches?

- **Multiple Pattern Matching Problem:**
  - **Input:** A collection of strings *Patterns* and a string *Genome.*
  - **Output:** All **positions** in *Genome* where one of *Patterns* appears as a substring.

- Where are the **positions**? BWT has not revealed them.

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

Where Are the Matches?

- Example: We know that **ana** occurs 3 times, but where?

$_1$panamabananas$_1$
a$_1$bananas$panam$_1$
a$_2$mabananaspan_1$
a$_3$namabananas$p$_1$
a$_4$nanas$panamab$_1$
a$_5$nas$panamaban$_2$
a$_6$s$panamabanan$_3$
b$_1$ananas$panama$_1$
m$_1$abananas$pana$_2$
n$_1$amabananaspa_3$
n$_2$anas$panamaba$_4$
n$_3$as$panamabana$_5$
p$_1$anamabananas$$_1$
s$_1$$panamabanana$_6$

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

Using the Suffix Array to Find Matches

- **Suffix array:** holds starting position of each suffix beginning a row.

$_1$panamabananas$_1$
a$_1$bananas$panam$_1$
a$_2$mabananaspan_1$
a$_3$namabananas$p$_1$
a$_4$nanas$panamab$_1$
a$_5$nas$panamaban$_2$
a$_6$s$panamabanan$_3$
b$_1$ananas$panama$_1$
m$_1$abananas$pana$_2$
n$_1$amabananaspa_3$
n$_2$anas$panamaba$_4$
n$_3$as$panamabana$_5$
p$_1$anamabananas$$_1$
s$_1$$panamabanana$_6$

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

Using the Suffix Array to Find Matches

- **Suffix array:** holds starting position of each suffix beginning a row.

panamabananas**$**

13	$_1$panamabananas$_1$
	a$_1$bananas$panam$_1$
	a$_2$mabananaspan_1$
	a$_3$namabananas$p$_1$
	a$_4$nanas$panamab$_1$
	a$_5$nas$panamaban$_2$
	a$_6$s$panamabanan$_3$
	b$_1$ananas$panama$_1$
	m$_1$abananas$pana$_2$
	n$_1$amabananaspa_3$
	n$_2$anas$panamaba$_4$
	n$_3$as$panamabana$_5$
	p$_1$anamabananas$$_1$
	s$_1$$panamabanana$_6$

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

Using the Suffix Array to Find Matches

- **Suffix array:** holds starting position of each suffix beginning a row.

panam**abananas$**

13	$\$_1$panamabananas$_1$
5	**a$_1$bananas$**panam$_1$
	a$_2$mabananaspan_1$
	a$_3$namabananas$p$_1$
	a$_4$nanas$panamab$_1$
	a$_5$nas$panamaban$_2$
	a$_6$s$panamabanan$_3$
	b$_1$ananas$panama$_1$
	m$_1$abananas$pana$_2$
	n$_1$amabananaspa_3$
	n$_2$anas$panamaba$_4$
	n$_3$as$panamabana$_5$
	p$_1$anamabananas$\$$_1$
	s$_1$$panamabanana$_6$

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

Using the Suffix Array to Find Matches

- **Suffix array:** holds starting position of each suffix beginning a row.

panamabananas$

13	$_1panamabananas_1
5	a_1bananas$panam_1
3	a_2mabananas$pan_1
	a_3namabananas$p_1
	a_4nanas$panamab_1
	a_5nas$panamaban_2
	a_6s$panamabanan_3
	b_1ananas$panama_1
	m_1abananas$pana_2
	n_1amabananas$pa_3
	n_2anas$panamaba_4
	n_3as$panamabana_5
	p_1anamabananas$_1
	s_1$panamabanana_6

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

Using the Suffix Array to Find Matches

- **Suffix array:** holds starting position of each suffix beginning a row.

p**anamabananas$**

13	$\$_1$panamabananas$_1$
5	a_1**bananas$**panam$_1$
3	a_2**mabananas$**pan$_1$
1	a_3**namabananas$**p$_1$
	$a_4$nanas$panamab$_1$
	$a_5$nas$panamaban$_2$
	a_6s$panamabanan$_3$
	$b_1$ananas$panama$_1$
	$m_1$abananas$pana$_2$
	$n_1$amabananaspa_3$
	$n_2$anas$panamaba$_4$
	$n_3$as$panamabana$_5$
	$p_1$anamabananas$$_1$
	$s_1$$panamabanana$_6$

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

Using the Suffix Array to Find Matches

- **Suffix array:** holds starting position of each suffix beginning a row.

panamab**ananas$**

13	**$**$_1$panamabananas$_1$
5	**a**$_1$**bananas$**panam$_1$
3	**a**$_2$**mabananas$**pan$_1$
1	**a**$_3$**namabananas$**p$_1$
7	**a**$_4$**nanas$**panamab$_1$
	a$_5$nas$panamaban$_2$
	a$_6$s$panamabanan$_3$
	b$_1$ananas$panama$_1$
	m$_1$abananas$pana$_2$
	n$_1$amabananaspa_3$
	n$_2$anas$panamaba$_4$
	n$_3$as$panamabana$_5$
	p$_1$anamabananas$$_1$
	s$_1$$panamabanana$_6$

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

Using the Suffix Array to Find Matches

- **Suffix array:** holds starting position of each suffix beginning a row.

panamaban**anas$**

13	**$**$_1$panamabananas$_1$
5	**a**$_1$**bananas$**panam$_1$
3	**a**$_2$**mabananas$**pan$_1$
1	**a**$_3$**namabananas$**p$_1$
7	**a**$_4$**nanas$**panamab$_1$
9	**a**$_5$**nas$**panamaban$_2$
	a$_6$s$panamabanan$_3$
	b$_1$ananas$panama$_1$
	m$_1$abananas$pana$_2$
	n$_1$amabananaspa_3$
	n$_2$anas$panamaba$_4$
	n$_3$as$panamabana$_5$
	p$_1$anamabananas$$_1$
	s$_1$$panamabanana$_6$

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

Using the Suffix Array to Find Matches

- **Suffix array:** holds starting position of each suffix beginning a row.

panamabanan**as$**

13	**$**$_1$panamabananas$_1$
5	**a**$_1$**bananas$**panam$_1$
3	**a**$_2$**mabananas$**pan$_1$
1	**a**$_3$**namabananas$**p$_1$
7	**a**$_4$**nanas$**panamab$_1$
9	**a**$_5$**nas$**panamaban$_2$
11	**a**$_6$**s$**panamabanan$_3$
	b$_1$ananas$panama$_1$
	m$_1$abananas$pana$_2$
	n$_1$amabananaspa_3$
	n$_2$anas$panamaba$_4$
	n$_3$as$panamabana$_5$
	p$_1$anamabananas$$_1$
	s$_1$$panamabanana$_6$

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

Using the Suffix Array to Find Matches

- **Suffix array:** holds starting position of each suffix beginning a row.

panama**bananas$**

13	$$_1$panamabananas$_1$
5	a$_1$bananas$panam$_1$
3	a$_2$mabananaspan_1$
1	a$_3$namabananas$p$_1$
7	a$_4$nanas$panamab$_1$
9	a$_5$nas$panamaban$_2$
11	a$_6$s$panamabanan$_3$
6	b$_1$ananas$panama$_1$
	m$_1$abananas$pana$_2$
	n$_1$amabananaspa_3$
	n$_2$anas$panamaba$_4$
	n$_3$as$panamabana$_5$
	p$_1$anamabananas$$_1$
	s$_1$$panamabanana$_6$

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

Using the Suffix Array to Find Matches

- **Suffix array:** holds starting position of each suffix beginning a row.

panamabananas$

13	$$_1$panamabananas$_1$
5	a$_1$bananas$panam$_1$
3	a$_2$mabananaspan_1$
1	a$_3$namabananas$p$_1$
7	a$_4$nanas$panamab$_1$
9	a$_5$nas$panamaban$_2$
11	a$_6$s$panamabanan$_3$
6	b$_1$ananas$panama$_1$
4	m$_1$abananas$pana$_2$
	n$_1$amabananaspa_3$
	n$_2$anas$panamaba$_4$
	n$_3$as$panamabana$_5$
	p$_1$anamabananas$$_1$
	s$_1$$panamabanana$_6$

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

Using the Suffix Array to Find Matches

- **Suffix array:** holds starting position of each suffix beginning a row.

panamabananas$

13	$$_1$panamabananas$_1$
5	a$_1$bananas$panam$_1$
3	a$_2$mabananaspan_1$
1	a$_3$namabananas$p$_1$
7	a$_4$nanas$panamab$_1$
9	a$_5$nas$panamaban$_2$
11	a$_6$s$panamabanan$_3$
6	b$_1$ananas$panama$_1$
4	m$_1$abananas$pana$_2$
2	n$_1$amabananaspa_3$
	n$_2$anas$panamaba$_4$
	n$_3$as$panamabana$_5$
	p$_1$anamabananas$$_1$
	s$_1$$panamabanana$_6$

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

Using the Suffix Array to Find Matches

- **Suffix array:** holds starting position of each suffix beginning a row.

panamaba**nanas$**

13	$$_1$panamabananas$_1$
5	a$_1$bananas$panam$_1$
3	a$_2$mabananaspan_1$
1	a$_3$namabananas$p$_1$
7	a$_4$nanas$panamab$_1$
9	a$_5$nas$panamaban$_2$
11	a$_6$s$panamabanan$_3$
6	b$_1$ananas$panama$_1$
4	m$_1$abananas$pana$_2$
2	n$_1$amabananaspa_3$
8	n$_2$anas$panamaba$_4$
	n$_3$as$panamabana$_5$
	p$_1$anamabananas$$_1$
	s$_1$$panamabanana$_6$

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

Using the Suffix Array to Find Matches

- **Suffix array:** holds starting position of each suffix beginning a row.

panamabana**nas$**

13	$$_1$panamabananas$_1$
5	a$_1$bananas$panam$_1$
3	a$_2$mabananaspan_1$
1	a$_3$namabananas$p$_1$
7	a$_4$nanas$panamab$_1$
9	a$_5$nas$panamaban$_2$
11	a$_6$s$panamabanan$_3$
6	b$_1$ananas$panama$_1$
4	m$_1$abananas$pana$_2$
2	n$_1$amabananaspa_3$
8	n$_2$anas$panamaba$_4$
10	n$_3$as$panamabana$_5$
	p$_1$anamabananas$$_1$
	s$_1$$panamabanana$_6$

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

Using the Suffix Array to Find Matches

- **Suffix array:** holds starting position of each suffix beginning a row.

panamabananas$

13	$$_1$panamabananas$_1$
5	a$_1$bananas$panam$_1$
3	a$_2$mabananaspan_1$
1	a$_3$namabananas$p$_1$
7	a$_4$nanas$panamab$_1$
9	a$_5$nas$panamaban$_2$
11	a$_6$s$panamabanan$_3$
6	b$_1$ananas$panama$_1$
4	m$_1$abananas$pana$_2$
2	n$_1$amabananaspa_3$
8	n$_2$anas$panamaba$_4$
10	n$_3$as$panamabana$_5$
0	p$_1$anamabananas$$_1$
	s$_1$$panamabanana$_6$

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

Using the Suffix Array to Find Matches

- **Suffix array:** holds starting position of each suffix beginning a row.

panamabanana**s$**

13	$_1$panamabananas$_1$
5	a$_1$bananas$panam$_1$
3	a$_2$mabananaspan_1$
1	a$_3$namabananas$p$_1$
7	a$_4$nanas$panamab$_1$
9	a$_5$nas$panamaban$_2$
11	a$_6$s$panamabanan$_3$
6	b$_1$ananas$panama$_1$
4	m$_1$abananas$pana$_2$
2	n$_1$amabananaspa_3$
8	n$_2$anas$panamaba$_4$
10	n$_3$as$panamabana$_5$
0	p$_1$anamabananas$$_1$
12	s$_1$$panamabanana$_6$

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

Using the Suffix Array to Find Matches

- **Suffix array:** holds starting position of each suffix beginning a row.

panamabananas$

13	$$_1$panamabananas$_1$
5	a$_1$bananas$panam$_1$
3	a$_2$mabananaspan_1$
1	a$_3$namabananas$p$_1$
7	a$_4$nanas$panamab$_1$
9	a$_5$nas$panamaban$_2$
11	a$_6$s$panamabanan$_3$
6	b$_1$ananas$panama$_1$
4	m$_1$abananas$pana$_2$
2	n$_1$amabananaspa_3$
8	n$_2$anas$panamaba$_4$
10	n$_3$as$panamaban$_5$
0	p$_1$anamabananas$$_1$
12	s$_1$$panamabanan$_6$

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

Using the Suffix Array to Find Matches

- **Suffix array:** holds starting position of each suffix beginning a row.

- Thus, **ana** occurs at positions **1**, **7**, **9** of panamab**anana**s$.

13	$$_1$panamabananas$_1$
5	a$_1$bananas$panam$_1$
3	a$_2$mabananaspan_1$
1	**a$_3$na**mabananas$p$_1$
7	**a$_4$na**nas$panamab$_1$
9	**a$_5$na**s$panamaban$_2$
11	a$_6$s$panamabanan$_3$
6	b$_1$ananas$panama$_1$
4	m$_1$abananas$pana$_2$
2	n$_1$amabananaspa_3$
8	n$_2$anas$panamaba$_4$
10	n$_3$as$panamabana$_5$
0	p$_1$anamabananas$$_1$
12	s$_1$$panamabanana$_6$

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

The Suffix Array: Memory Once Again

- Memory: ~ 4 x |*Genome*|.

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

The Suffix Array: Memory Once Again

- Memory: ~ 4 x |*Genome*|.

[13 **5** 3 1 7 9 11 6 4 2 8 10 0 12]

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

The Suffix Array: Memory Once Again

- Memory: ~ 4 x |*Genome*|.

[13 5 **3** 1 7 9 11 6 4 2 8 10 0 12]

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

The Suffix Array: Memory Once Again

- Memory: ~ 4 x |*Genome*|.

[13 5 3 **1** 7 9 11 6 4 2 8 10 0 12]

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

The Suffix Array: Memory Once Again

- Memory: ~ 4 x |*Genome*|.

[13 5 3 1 **7** 9 11 6 4 2 8 10 0 12]

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

The Suffix Array: Memory Once Again

- Memory: ~ 4 x |*Genome*|.

[13 5 3 1 7 **9** 11 6 4 2 8 10 0 12]

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

The Suffix Array: Memory Once Again

- Memory: ~ 4 x |*Genome*|.

[13 5 3 1 7 9 **11** 6 4 2 8 10 0 12]

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

The Suffix Array: Memory Once Again

- Memory: ~ 4 x |*Genome*|.

[13 5 3 1 7 9 11 **6** 4 2 8 10 0 12]

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

The Suffix Array: Memory Once Again

- Memory: ~ 4 x |*Genome*|.

[13 5 3 1 7 9 11 6 **4** 2 8 10 0 12]

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

The Suffix Array: Memory Once Again

- Memory: ~ 4 x |*Genome*|.

[13 5 3 1 7 9 11 6 4 **2** 8 10 0 12]

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

The Suffix Array: Memory Once Again

- Memory: ~ 4 x |*Genome*|.

[13 5 3 1 7 9 11 6 4 2 **8** 10 0 12]

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

The Suffix Array: Memory Once Again

- Memory: ~ 4 x |*Genome*|.

[13 5 3 1 7 9 11 6 4 2 8 **10** 0 12]

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

The Suffix Array: Memory Once Again

- Memory: ~ 4 x |*Genome*|.

[13 5 3 1 7 9 11 6 4 2 8 10 **0** 12]

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

The Suffix Array: Memory Once Again

- Memory: ~ 4 x |*Genome*|.

[13 5 3 1 7 9 11 6 4 2 8 10 0 **12**]

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

The Suffix Array: Memory Once Again

- Memory: ~ 4 x |*Genome*|.

[13 5 3 1 7 9 11 6 4 2 8 10 0 12]

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

Outline

- Why do we map reads?
- Using the Trie
- From a Trie to a Suffix Tree
- String Compression and the Burrows-Wheeler Transform
- Inverting Burrows-Wheeler
- Using Burrows-Wheeler for Pattern Matching
- Finding the Matched Patterns
- **Setting Up Checkpoints**
- Inexact Matching
- Further Applications of Read Mapping

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

Reducing Suffix Array Size

- We don’t want to have to store all of the suffix array; can we store only part of it? Show how checkpointing can be used to store 1/100 the suffix array.

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

A Return to Constants

- Explain that using a checkpointed array increases runtime by a constant factor, but in practice it is a worthwhile trade-off.

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

```
BWMatching(FirstColumn, LastColumn, Pattern, LastToFirst)
    top ← 0
    bottom ← |LastColumn| − 1
    while top ≤ bottom
        if Pattern is nonempty
            symbol ← last letter in Pattern
            remove last letter from Pattern
            if positions from top to bottom in LastColumn contain symbol
                topIndex ← first position of symbol among positions from top to bottom
                           in LastColumn
                bottomIndex ← last position of symbol among positions from top to
                              bottom in LastColumn
                top ← LastToFirst(topIndex)
                bottom ← LastToFirst(bottomIndex)
            else
                return 0
        else
            return bottom − top + 1
```

Speeding Up Burrows-Wheeler Pattern Matching

COUNT$_{symbol}$***(i, LastColumn)***, which returns the number of occurrences of symbol in the first **i** positions of **LastColumn**.

COUNT$_{"n"}$(10, "smnpbnnaaaaa$a") = 3
COUNT$_{"a"}$(4, "smnpbnnaaaaa$a") = 0

i	*FirstColumn*	*LastColumn*	LastToFirst(i)	Count						
				$	a	b	m	n	p	s
0	$\$_1$	s_1	13	0	0	0	0	0	0	0
1	a_1	m_1	8	0	0	0	0	0	0	1
2	a_2	n_1	9	0	0	0	1	0	0	1
3	a_3	p_1	12	0	0	0	1	1	0	1
4	a_4	b_1	7	0	0	0	1	1	1	1
5	a_5	n_2	10	0	0	1	1	1	1	1
6	a_6	n_3	11	0	0	1	1	2	1	1
7	b_1	a_1	1	0	0	1	1	3	1	1
8	m_1	a_2	2	0	1	1	1	3	1	1
9	n_1	a_3	3	0	2	1	1	3	1	1
10	n_2	a_4	4	0	3	1	1	3	1	1
11	n_3	a_5	5	0	4	1	1	3	1	1
12	p_1	$\$_1$	0	0	5	1	1	3	1	1
13	s_1	a_6	6	1	5	1	1	3	1	1
				1	6	1	1	3	1	1

i	*FirstColumn*	FIRSTOCCURRENCE
0	$\$_1$	0
1	a_1	1
2	a_2	
3	a_3	
4	a_4	
5	a_5	
6	a_6	
7	b_1	7
8	m_1	8
9	n_1	9
10	n_2	
11	n_3	
12	p_1	12
13	s_1	13

Getting rid of ***FirstColumn***

For DNA **FIRSTOCCURRENCE** contains only **five** elements.

Speeding Up Burrows-Wheeler Pattern Matching

Updating Index — in constant time

$$top \leftarrow \textsc{FirstOccurrence}(symbol) + \textsc{Count}_{symbol}(top, LastColumn)$$
$$bottom \leftarrow \textsc{FirstOccurrence}(symbol) + \textsc{Count}_{symbol}(bottom + 1, LastColumn) - 1$$

When **`top = 1`**, **`bottom = 6`**, and symbol = "**n**", we have that

$$\text{FIRSTOCCURRENCE("n")} = 9$$
$$\text{COUNT}_{\text{"n"}}(top, LastColumn) = 0$$
$$\text{COUNT}_{\text{"n"}}(bottom + 1, LastColumn) = 3$$
$$top = 9 + 0 = 9$$
$$bottom = 9 + 3 - 1 = 11$$

```
BETTERBWMATCHING(FIRSTOCCURRENCE, LastColumn, Pattern, COUNT)
    top ← 0
    bottom ← |LastColumn| − 1
    while top ≤ bottom
        if Pattern is nonempty
            symbol ← last letter in Pattern
            remove last letter from Pattern
            top ← FIRSTOCCURRENCE(symbol) + COUNT_symbol(top, LastColumn)
            bottom ← FIRSTOCCURRENCE(symbol) + COUNT_symbol(bottom + 1,
                        LastColumn) − 1
        else
            return bottom − top + 1
    return
```

Where are the Matched Patterns?

BETTERBWMATCHING

— counts the number of occurrences of Pattern in Text

— does not tell us where these occurrences are located in Text

We can use the **suffix array** to find the locations. But if we add the suffix array, then we are right back where we started!

We will build a **partial suffix array** of Text, denoted **SUFFIXARRAY$_K$(Text)**, which only contains values that are multiples of some positive integer **K**. In real applications, **K = 100**, reducing memory usage by a factor of 100 compared to a full suffix array.

Partial Suffix Array

panamabananas$	panamabananas$	panamabananas$	Partial Suffix Array
$$_1$panamabananas$_1$	$$_1$panamabananas$_1$	$$_1$panamabananas$_1$	13
a$_1$bananas$panam$_1$	a$_1$bananas$panam$_1$	a$_1$bananas$panam$_1$	5
a$_2$mabananaspan_1$	a$_2$mabananaspan_1$	a$_2$mabananaspan_1$	3
a$_3$namabananas$p$_1$	a$_3$namabananas$p$_1$	a$_3$namabananas$p$_1$	1
a$_4$nanas$panamab$_1$	a$_4$nanas$panamab$_1$	a$_4$nanas$panamab$_1$	7
a$_5$nas$panamaban$_2$	a$_5$nas$panamaban$_2$	a$_5$nas$panamaban$_2$	9
a$_6$s$panamabanan$_3$	a$_6$s$panamabanan$_3$	a$_6$s$panamabanan$_3$	11
b$_1$ananas$panama$_1$	b$_1$ananas$panama$_1$	b$_1$ananas$panama$_1$	6
m$_1$abananas$pana$_2$	m$_1$abananas$pana$_2$	m$_1$abananas$pana$_2$	4
n$_1$amabananaspa_3$	n$_1$amabananaspa_3$	n$_1$amabananaspa_3$	2
n$_2$anas$panamaba$_4$	n$_2$anas$panamaba$_4$	n$_2$anas$panamaba$_4$	8
n$_3$as$panamabana$_5$	n$_3$as$panamabana$_5$	n$_3$as$panamabana$_5$	10
p$_1$anamabananas$$_1$	p$_1$anamabananas$$_1$	p$_1$anamabananas$$_1$	0
s$_1$$panamabanana$_6$	s$_1$$panamabanana$_6$	s$_1$$panamabanana$_6$	12

One of the matches of **"ana"** in **"panamabananas$"** is highlighted in the matrix on the right. By walking backward, we find that **"ana"** is preceded by **"b1"**, which in turn is preceded by **"a1"**. The partial suffix array above, generated for **K = 5**, indicates that **"a1"** occurs at position 5 of **"panamabananas"**. Since it took us two steps to walk backward to **"a1"**, we conclude that this occurrence of **"ana"** begins at position **5 + 2 = 7**.

Burrows and Wheeler Set Up Checkpoints

i	*LastColumn*	COUNT						
		$	a	b	m	n	p	s
0	s_1	**0**	**0**	**0**	**0**	**0**	**0**	**0**
1	m_1	0	0	0	0	0	0	1
2	n_1	0	0	0	1	0	0	1
3	p_1	0	0	0	1	1	0	1
4	b_1	0	0	0	1	1	1	1
5	n_2	**0**	**0**	**1**	**1**	**1**	**1**	**1**
6	n_3	0	0	1	1	2	1	1
7	a_1	0	0	1	1	3	1	1
8	a_2	0	1	1	1	3	1	1
9	a_3	0	2	1	1	3	1	1
10	a_4	**0**	**3**	**1**	**1**	**3**	**1**	**1**
11	a_5	0	4	1	1	3	1	1
12	$\$_1$	0	5	1	1	3	1	1
13	a_6	1	5	1	1	3	1	1
		1	6	1	1	3	1	1

Outline

- Why do we map reads?
- Using the Trie
- From a Trie to a Suffix Tree
- String Compression and the Burrows-Wheeler Transform
- Inverting Burrows-Wheeler
- Using Burrows-Wheeler for Pattern Matching
- Finding the Matched Patterns
- Setting Up Checkpoints
- **Inexact Matching**
- Further Applications of Read Mapping

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

Returning to Our Original Problem

- We need to look at INEXACT matching in order to find variants.

- **Approx. Pattern Matching Problem**:
  - **Input**: A string *Pattern*, a string *Genome*, and an integer *d*.
  - **Output:** All positions in *Genome* where *Pattern* appears as a substring with at most *d* mismatches.

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

Returning to Our Original Problem

- We need to look at INEXACT matching in order to find variants.

- **Multiple Approx. Pattern Matching Problem**:
  - **Input**: A **collection** of strings *Patterns*, a string *Genome*, and an integer *d*.
  - **Output:** All positions in *Genome* where a string from *Patterns* appears as a substring with at most *d* mismatches.

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

Method 1: Seeding

- Say that *Pattern* appears in *Genome* with 1 mismatch:

Pattern	`actt`**`t`**`ggct`
Genome	`…ggcac``actt`**`a`**`ggct``cc…`

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

Method 1: Seeding

- Say that *Pattern* appears in *Genome* with 1 mismatch:

Pattern actt**t**ggct

Genome …ggcacact**a**ggctcc…

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

Method 1: Seeding

- Say that *Pattern* appears in *Genome* with 1 mismatch:

Pattern actt|ggct

Genome …ggcacact|aggctcc…

- One of the substrings must match!

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

Method 1: Seeding

- **Theorem:** If *Pattern* occurs in *Genome* with *d* mismatches, then we can divide *Pattern* into *d* + 1 "equal" pieces and find at least one exact match.

xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

Method 1: Seeding

- **Theorem:** If *Pattern* occurs in *Genome* with *d* mismatches, then we can divide *Pattern* into *d* + 1 "equal" pieces and find at least one exact match.

```
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
```

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

Method 1: Seeding

- **Theorem:** If *Pattern* occurs in *Genome* with *d* mismatches, then we can divide *Pattern* into *d* + 1 "equal" pieces and find at least one exact match.

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

Method 1: Seeding

- **Theorem:** If *Pattern* occurs in *Genome* with *d* mismatches, then we can divide *Pattern* into *d* + 1 "equal" pieces and find at least one exact match.

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

Method 1: Seeding

- **Theorem:** If *Pattern* occurs in *Genome* with *d* mismatches, then we can divide *Pattern* into *d* + 1 "equal" pieces and find at least one exact match.

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

Method 1: Seeding

- Say we are looking for at most *d* mismatches.

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

Method 1: Seeding

- Say we are looking for at most *d* mismatches.
- Divide each of our strings into *d* + 1 smaller pieces, called **seeds**.

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

Method 1: Seeding

- Say we are looking for at most *d* mismatches.
- Divide each of our strings into *d* + 1 smaller pieces, called **seeds**.
- Check if each *Pattern* has a seed that matches *Genome* exactly.

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

Method 1: Seeding

- Say we are looking for at most *d* mismatches.
- Divide each of our strings into *d* + 1 smaller pieces, called **seeds**.
- Check if each *Pattern* has a seed that matches *Genome* exactly.
- If so, check the entire *Pattern* against *Genome*.

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

Method 2: BWT Saves the Day Again

- Recall: searching for `ana` in `panamabananas`

$_1$panamabananas$_1$
a$_1$bananas$panam$_1$
a$_2$mabananaspan_1$
a$_3$namabananas$p$_1$
a$_4$nanas$panamab$_1$
a$_5$nas$panamaban$_2$
a$_6$s$panamabanan$_3$
b$_1$ananas$panama$_1$
m$_1$abananas$pana$_2$
n$_1$amabananaspa_3$
n$_2$anas$panamaba$_4$
n$_3$as$panamabana$_5$
p$_1$anamabananas$$_1$
s$_1$$panamabanana$_6$

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

Method 2: BWT Saves the Day Again

- Recall: searching for `ana` in `panamabananas`

$_1$panamabananas$_1$
a$_1$bananas$panam$_1$
a$_2$mabananaspan_1$
a$_3$namabananas$p$_1$
a$_4$nanas$panamab$_1$
a$_5$nas$panamaban$_2$
a$_6$s$panamabanan$_3$
b$_1$ananas$panama$_1$
m$_1$abananas$pana$_2$
n$_1$amabananaspa_3$
n$_2$anas$panamaba$_4$
n$_3$as$panamabana$_5$
p$_1$anamabananas$$_1$
s$_1$$panamabanana$_6$

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

Method 2: BWT Saves the Day Again

- Recall: searching for a**na** in panamabananas

$_1$panamabananas$_1$
a$_1$bananas$panam**m**$_1$
a$_2$mabananas$pa**n**$_1$
a$_3$namabananas$**p**$_1$
a$_4$nanas$panama**b**$_1$
a$_5$nas$panamaba**n**$_2$
a$_6$s$panamabana**n**$_3$
b$_1$ananas$panama$_1$
m$_1$abananas$pana$_2$
n$_1$amabananaspa_3$
n$_2$anas$panamaba$_4$
n$_3$as$panamabana$_5$
p$_1$anamabananas$$_1$
s$_1$$panamabanana$_6$

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

Method 2: BWT Saves the Day Again

- Recall: searching for **ana** in panamabananas

If we allow 1 mismatch, then we need to keep the red letters around.

$_1$panamabananas$_1$
a$_1$bananas$panam**m**$_1$
a$_2$mabananas$pa**n**$_1$
a$_3$namabananas$**p**$_1$
a$_4$nanas$panama**b**$_1$
a$_5$nas$panamaba**n**$_2$
a$_6$s$panamabana**n**$_3$
b$_1$ananas$panama$_1$
m$_1$abananas$pana$_2$
n$_1$amabananaspa_3$
n$_2$anas$panamaba$_4$
n$_3$as$panamabana$_5$
p$_1$anamabananas$$_1$
s$_1$$panamabanana$_6$

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

Method 2: BWT Saves the Day Again

- Recall: searching for **ana** in panamabananas

If we allow 1 mismatch, then we need to keep the red letters around.

Mismatches

$_1$panamabananas$_1$
a$_1$bananas$panam$_1$
a$_2$mabananaspan_1$
a$_3$namabananas$p$_1$
a$_4$nanas$panamab$_1$
a$_5$nas$panamaban$_2$
a$_6$s$panamabanan$_3$
b$_1$ananas$panama$_1$
m$_1$abananas$pana$_2$
n$_1$amabananaspa_3$
n$_2$anas$panamaba$_4$
n$_3$as$panamabana$_5$
p$_1$anamabananas$$_1$
s$_1$$panamabanana$_6$

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

Method 2: BWT Saves the Day Again

- Recall: searching for **ana** in panamabananas

If we allow 1 mismatch, then we need to keep the red letters around.

	# Mismatches
$_1$panamabananas$_1$	
a$_1$bananas$panam**m**$_1$	1
a$_2$mabananas$pa**n**$_1$	0
a$_3$namabananas$**p**$_1$	1
a$_4$nanas$panama**b**$_1$	1
a$_5$nas$panamaba**n**$_2$	0
a$_6$s$panamabana**n**$_3$	0
b$_1$ananas$panama$_1$	
m$_1$abananas$pana$_2$	
n$_1$amabananaspa_3$	
n$_2$anas$panamaba$_4$	
n$_3$as$panamabana$_5$	
p$_1$anamabananas$$_1$	
s$_1$$panamabanana$_6$	

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

Method 2: BWT Saves the Day Again

- Recall: searching for **ana** in panamabananas

Now we extend all strings with at most 1 mismatch.

	# Mismatches
$_1$panamabananas$_1$	
a$_1$bananas$panam$_1$	1
a$_2$mabananaspan_1$	0
a$_3$namabananas$p$_1$	1
a$_4$nanas$panamab$_1$	1
a$_5$nas$panamaban$_2$	0
a$_6$s$panamabanan$_3$	0
b$_1$ananas$panama$_1$	
m$_1$abananas$pana$_2$	
n$_1$amabananaspa_3$	
n$_2$anas$panamaba$_4$	
n$_3$as$panamabana$_5$	
p$_1$anamabananas$$_1$	
s$_1$$panamabanana$_6$	

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

Method 2: BWT Saves the Day Again

- Recall: searching for **ana** in panamabananas

Now we extend all strings with at most 1 mismatch.

	# Mismatches
$_1$panamabananas$_1$	
a$_1$bananas$panam$_1$	1
a$_2$mabananaspan_1$	0
a$_3$namabananas$p$_1$	1
a$_4$nanas$panamab$_1$	1
a$_5$nas$panamaban$_2$	0
a$_6$s$panamabanan$_3$	0
b$_1$ananas$panama$_1$	
m$_1$abananas$pana$_2$	
n$_1$amabananaspa_3$	
n$_2$anas$panamaba$_4$	
n$_3$as$panamabana$_5$	
p$_1$anamabananas$$_1$	
s$_1$$panamabanana$_6$	

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

Method 2: BWT Saves the Day Again

- Recall: searching for **ana** in panamabananas

Now we extend all strings with at most 1 mismatch.

	# Mismatches
$_1$panamabananas$_1$	
a$_1$bananas$panam$_1$	1
a$_2$mabananaspan_1$	0
a$_3$namabananas$p$_1$	1
a$_4$nanas$panamab$_1$	1
a$_5$nas$panamaban$_2$	0
a$_6$s$panamabanan$_3$	0
b$_1$ananas$panama$_1$	
m$_1$abananas$pana$_2$	
n$_1$amabananaspa_3$	
n$_2$anas$panamaba$_4$	
n$_3$as$panamabana$_5$	
p$_1$anamabananas$$_1$	
s$_1$$panamabanana$_6$	

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

Method 2: BWT Saves the Day Again

- Recall: searching for **ana** in panamabananas

Now we extend all strings with at most 1 mismatch.

	# Mismatches
$\$_1$panamabananas$_1$	
a$_1$bananas$panam$_1$	
a$_2$mabananaspan_1$	
a$_3$namabananas$p$_1$	
a$_4$nanas$panamab$_1$	
a$_5$nas$panamaban$_2$	
a$_6$s$panamabanan$_3$	
b$_1$ananas$panama$_1$	1
m$_1$abananas$pana$_2$	1
n$_1$amabananaspa_3$	0
n$_2$anas$panamaba$_4$	0
n$_3$as$panamaban$a$_5$	0
p$_1$anamabananas$\$_1$	1
s$_1$$panamabanana$_6$	

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

Method 2: BWT Saves the Day Again

- Recall: searching for **ana** in panamabananas

One string produces a second mismatch (the $), so we discard it.

	# Mismatches
$$_1$panamabananas$_1$	
a$_1$bananas$panam$_1$	
a$_2$mabananaspan_1$	
a$_3$namabananas$p$_1$	
a$_4$nanas$panamab$_1$	
a$_5$nas$panamaban$_2$	
a$_6$s$panamabanan$_3$	
b$_1$ananas$panam**a$_1$**	1
m$_1$abananas$pan**a$_2$**	1
n$_1$amabananas$p**a$_3$**	0
n$_2$anas$panamab**a$_4$**	0
n$_3$as$panaman**a$_5$**	0
p$_1$anamabananas**$$_1$**	2
s$_1$$panamabanan**a$_6$**	

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

Method 2: BWT Saves the Day Again

- Recall: searching for **ana** in panamabananas

In the end, we have five 3-mers with at most 1 mismatch.

	# Mismatches
$\$_1$panamabananas$_1$	
a$_1$bananas$panam$_1$	1
a$_2$mabananaspan_1$	1
a$_3$namabananas$p$_1$	0
a$_4$nanas$panamab$_1$	0
a$_5$nas$panamaban$_2$	0
a$_6$s$panamabanan$_3$	
b$_1$ananas$panam**a$_1$**	
m$_1$abananas$pan**a$_2$**	
n$_1$amabananas$p**a$_3$**	
n$_2$anas$panamab**a$_4$**	
n$_3$as$panamaban**a$_5$**	
p$_1$anamabananas**$$_1$**	
s$_1$$panamabanana$_6$	

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

Method 2: BWT Saves the Day Again

- Recall: searching for **ana** in panamabananas

In the end, we have five 3-mers with at most 1 mismatch.

	# Mismatches
$\$_1$panamabananas$_1$	
a$_1$bananas$panam$_1$	1
a$_2$mabananaspan_1$	1
a$_3$namabananas$p$_1$	0
a$_4$nanas$panamab$_1$	0
a$_5$nas$panamaban$_2$	0
a$_6$s$panamabanan$_3$	
b$_1$ananas$panama$_1$	
m$_1$abananas$pana$_2$	
n$_1$amabananaspa_3$	
n$_2$anas$panamaba$_4$	
n$_3$as$panamabana$_5$	
p$_1$anamabananas$\$_1$	
s$_1$$panamabanana$_6$	

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

Method 2: BWT Saves the Day Again

- Recall: searching for **ana** in panamabananas

In the end, we have five 3-mers with at most 1 mismatch.

	Suffix Array
$\$_1$panamabananas$_1$	
a$_1$bananas$panam$_1$	5
a$_2$mabananaspan_1$	3
a$_3$namabananas$p$_1$	1
a$_4$nanas$panamab$_1$	7
a$_5$nas$panamaban$_2$	9
a$_6$s$panamabanan$_3$	
b$_1$ananas$panama$_1$	
m$_1$abananas$pana$_2$	
n$_1$amabananaspa_3$	
n$_2$anas$panamaba$_4$	
n$_3$as$panamabana$_5$	
p$_1$anamabananas$\$_1$	
s$_1$$panamabanana$_6$	

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

Method 2: BWT Saves the Day Again

- Recall: searching for **ana** in panamabananas

In the end, we have five 3-mers with at most 1 mismatch.

	Suffix Array
$_1$panamabananas$_1$	
a$_1$bananas$panam$_1$	**5**
a$_2$mabananaspan_1$	3
a$_3$namabananas$p$_1$	1
a$_4$nanas$panamab$_1$	7
a$_5$nas$panamaban$_2$	9
a$_6$s$panamabanan$_3$	
b$_1$ananas$panama$_1$	
m$_1$abananas$pana$_2$	
n$_1$amabananaspa_3$	
n$_2$anas$panamaba$_4$	
n$_3$as$panamabana$_5$	
p$_1$anamabananas$$_1$	
s$_1$$panamabanana$_6$	

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

Method 2: BWT Saves the Day Again

- Recall: searching for **ana** in panamabananas

In the end, we have five 3-mers with at most 1 mismatch.

	Suffix Array
$\$_1$panamabananas$_1$	
a$_1$bananas$panam$_1$	5
a$_2$mabananaspan_1$	**3**
a$_3$namabananas$p$_1$	1
a$_4$nanas$panamab$_1$	7
a$_5$nas$panamaban$_2$	9
a$_6$s$panamabanan$_3$	
b$_1$ananas$panama$_1$	
m$_1$abananas$pana$_2$	
n$_1$amabananaspa_3$	
n$_2$anas$panamaba$_4$	
n$_3$as$panamabana$_5$	
p$_1$anamabananas$\$_1$	
s$_1$$panamabanana$_6$	

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

Method 2: BWT Saves the Day Again

- Recall: searching for **ana** in panamabananas

In the end, we have five 3-mers with at most 1 mismatch.

	Suffix Array
$\$_1$panamabananas$_1$	
a$_1$bananas$panam$_1$	5
a$_2$mabananaspan_1$	3
a$_3$namabananas$p$_1$	**1**
a$_4$nanas$panamab$_1$	7
a$_5$nas$panamaban$_2$	9
a$_6$s$panamabanan$_3$	
b$_1$ananas$panama$_1$	
m$_1$abananas$pana$_2$	
n$_1$amabananaspa_3$	
n$_2$anas$panamaba$_4$	
n$_3$as$panamaban$a$_5$	
p$_1$anamabananas$\$_1$	
s$_1$$panamabanan$a$_6$	

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

Method 2: BWT Saves the Day Again

- Recall: searching for **ana** in panamab**ana**nas

In the end, we have five 3-mers with at most 1 mismatch.

	Suffix Array
$\$_1$panamabananas$_1$	
a$_1$bananas$panam$_1$	5
a$_2$mabananaspan_1$	3
a$_3$namabananas$p$_1$	1
a$_4$nanas$panamab$_1$	**7**
a$_5$nas$panamaban$_2$	9
a$_6$s$panamabanan$_3$	
b$_1$ananas$panama$_1$	
m$_1$abananas$pana$_2$	
n$_1$amabananaspa_3$	
n$_2$anas$panamaba$_4$	
n$_3$as$panamaban$a$_5$	
p$_1$anamabananas$\$_1$	
s$_1$$panamabanan$a$_6$	

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

Method 2: BWT Saves the Day Again

- Recall: searching for **ana** in panamaban**ana**s

In the end, we have five 3-mers with at most 1 mismatch.

	Suffix Array
$\$_1$panamabananas$_1$	
a$_1$bananas$panam$_1$	5
a$_2$mabananaspan_1$	3
a$_3$namabananas$p$_1$	1
a$_4$nanas$panamab$_1$	7
a$_5$nas$panamaban$_2$	**9**
a$_6$s$panamabanan$_3$	
b$_1$ananas$panama$_1$	
m$_1$abananas$pana$_2$	
n$_1$amabananaspa_3$	
n$_2$anas$panamaba$_4$	
n$_3$as$panamabana$_5$	
p$_1$anamabananas$\$_1$	
s$_1$$panamabanana$_6$	

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

Outline

- Why do we map reads?
- Using the Trie
- From a Trie to a Suffix Tree
- String Compression and the Burrows-Wheeler Transform
- Inverting Burrows-Wheeler
- Using Burrows-Wheeler for Pattern Matching
- Finding the Matched Patterns
- Setting Up Checkpoints
- Inexact Matching
- **Further Applications of Read Mapping**

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

Read Mapping Is Even Harder

- Genomes have more than just point mutations. They have insertions, deletions, duplications, inversions. How can we map reads in this case?

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.

Variant Calling by Read Mapping

Bioinformatics Algorithms: An Active Learning Approach.
Copyright 2018 Compeau and Pevzner.