Lecture 9: Mapping Reads to a Reference – Burrows Wheeler Transform and FM Index

Spring 2020
March 3,5, 2020

Outline

- Problem Definition
- Different Solutions
- Burrows-Wheeler Transformation (BWT)
- Ferragina-Manzini (FM) Index
- Search Using FM Index
- Alignment Using FM Index

Mapping Reads

Problem: We are given a read, *R,* and a reference sequence, *S*. Find the best or all occurrences of *R* in *S*.

Example:

R = AAACGAGTTA

S = TTAATGC*AAACGAGTTA*CCCAATATATAT*AAACCAGTTA*TT

Considering no error: one occurrence.

Considering up to 1 substitution error: two occurrences.

Considering up to 10 substitution errors: many meaningless occurrences!

Mapping Reads (continued)

Variations:

- Sequencing error
  - No error: *R* is a perfect subsequence of *S.*
  - Only substitution error: *R* is a subsequence of *S* up to a few substitutions.
  - Indel and substitution error: *R* is a subsequence of *S* up to a few short indels and substitutions.
- Junctions (for instance in alternative splicing)
  - Fixed order/orientation

    $R = R_1R_2…R_n$ and R_i map to different non-overlapping loci in *S*, but to the same strand and preserving the order.
  - Arbitrary order/orientation

    $R = R_1R_2…R_n$ and R_i map to different non-overlapping loci in *S.*

Different Solutions

- Alignment, such as Smith-Waterman algorithm:
  - Pro: adequate for all variations.
  - Con: computationally expensive, not suitable for next-generation sequencing.
- Seed-and-Extend
  - Pro: can handle errors and junctions more efficiently.
  - Con: slow when no (few) error(s).
- Ferragina Manzini (FM) Index Search
  - Pro: computationally efficient, when no error.
  - Con: exponential in the maximum number of errors.

Burrows-Wheeler Transformation
Example: mississippi

1. Append to the input string a special char, $, smaller than all alphabet.

mississippi$

Burrows-Wheeler Transformation (cnt'd)

Example: mississippi

2. Generate all rotations.

m	i	s	s	i	s	s	i	p	p	i	$
i	s	s	i	s	s	i	p	p	i	$	m
s	s	i	s	s	i	p	p	i	$	m	i
s	i	s	s	i	p	p	i	$	m	i	s
i	s	s	i	p	p	i	$	m	i	s	s
s	s	i	p	p	i	$	m	i	s	s	i
s	i	p	p	i	$	m	i	s	s	i	s
i	p	p	i	$	m	i	s	s	i	s	s
p	p	i	$	m	i	s	s	i	s	s	i
p	i	$	m	i	s	s	i	s	s	i	p
i	$	m	i	s	s	i	s	s	i	p	p
$	m	i	s	s	i	s	s	i	p	p	i

Burrows-Wheeler Transformation (cnt'd)

Example: mississippi

3. Sort rotations according to the alphabetical order.

$	m	i	s	s	i	s	s	i	p	p	i
i	$	m	i	s	s	i	s	s	i	p	p
i	p	p	i	$	m	i	s	s	i	s	s
i	s	s	i	p	p	i	$	m	i	s	s
i	s	s	i	s	s	i	p	p	i	$	m
m	i	s	s	i	s	s	i	p	p	i	$
p	i	$	m	i	s	s	i	s	s	i	p
p	p	i	$	m	i	s	s	i	s	s	i
s	i	p	p	i	$	m	i	s	s	i	s
s	i	s	s	i	p	p	i	$	m	i	s
s	s	i	p	p	i	$	m	i	s	s	i
s	s	i	s	s	i	p	p	i	$	m	i

Burrows-Wheeler Transformation (cnt'd)

Example: mississippi

4. Output the last column.

$	m	i	s	s	i	s	s	i	p	p	i
i	$	m	i	s	s	i	s	s	i	p	p
i	p	p	i	$	m	i	s	s	i	s	s
i	s	s	i	p	p	i	$	m	i	s	s
i	s	s	i	s	s	i	p	p	i	$	m
m	i	s	s	i	s	s	i	p	p	i	$
p	i	$	m	i	s	s	i	s	s	i	p
p	p	i	$	m	i	s	s	i	s	s	i
s	i	p	p	i	$	m	i	s	s	i	s
s	i	s	s	i	p	p	i	$	m	i	s
s	s	i	p	p	i	$	m	i	s	s	i
s	s	i	s	s	i	p	p	i	$	m	i

Burrows-Wheeler Transformation (cnt'd)

Example: mississippi

ipssm$pissii

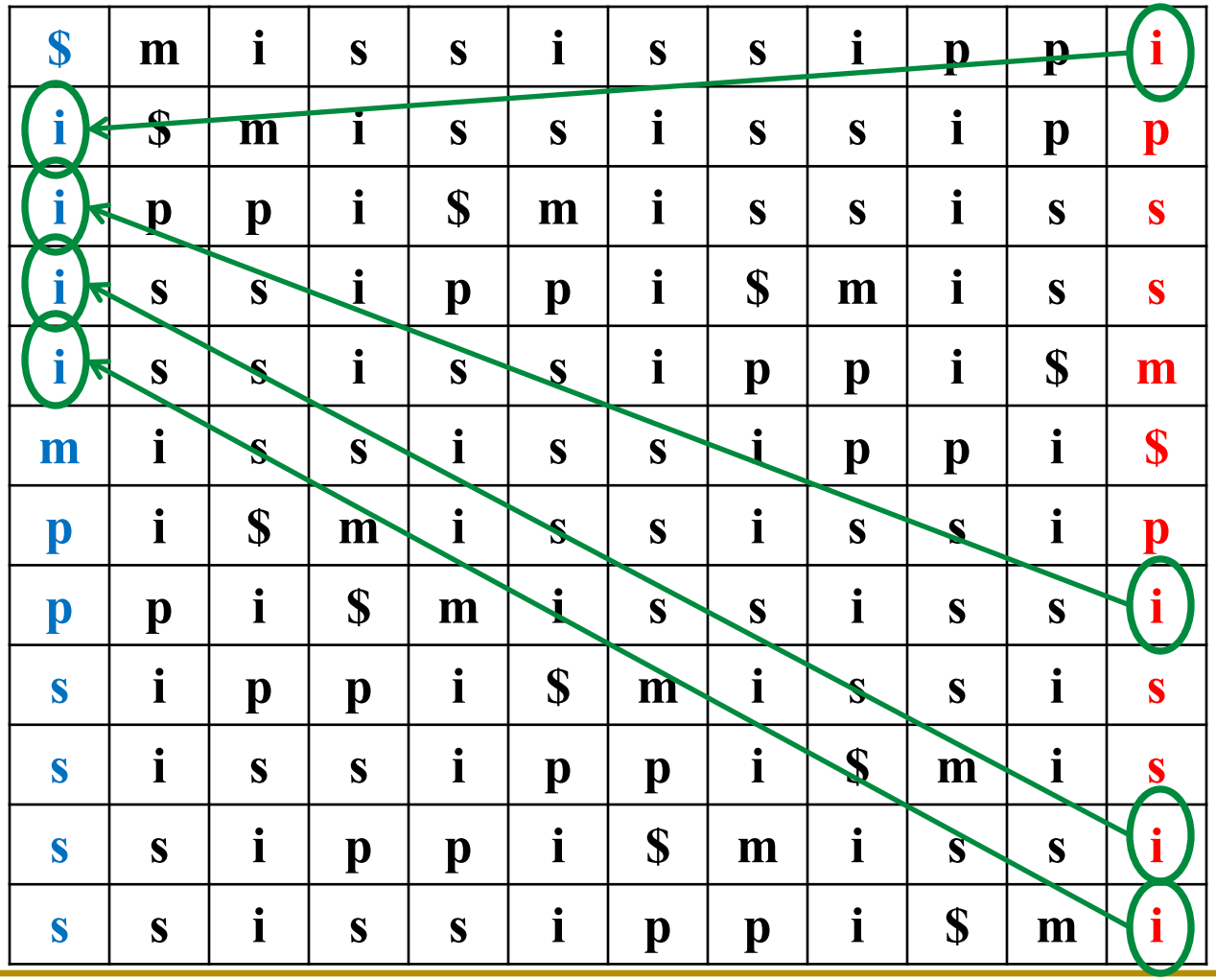
Ferragina-Manzini Index
Example: mississippi

First column: F

Last column: L

Let's make an L to F map.

Observation: The n^{th} i in L is the n^{th} i in F.



Ferragina-Manzini Index (cnt'd)

L to F map

Store/compute a two dimensional Occ(*j*,'c') table of the number of occurrences of char 'c' up to position *j* (inclusive).

and a one dimensional Cnt('c') table.

	$	i	m	p	s
i	0	1	0	0	0
p	0	1	0	1	0
s	0	1	0	1	1
s	0	1	0	1	2
m	0	1	1	1	2
$	1	1	1	1	2
p	1	1	1	2	2
i	1	2	1	2	2
s	1	2	1	2	3
s	1	2	1	2	4
i	1	3	1	2	4
i	1	4	1	2	4

Occ(*j*,'c')

Cnt('c')

$	i	m	p	s
1	4	1	2	4

Ferragina-Manzini Index
L to F map

[Cnt('$') +

Cnt('i') +

Cnt('m') +

Cnt('p') = 8]

+

[Occ(9, 's')= 3]

= 11 before 's' →

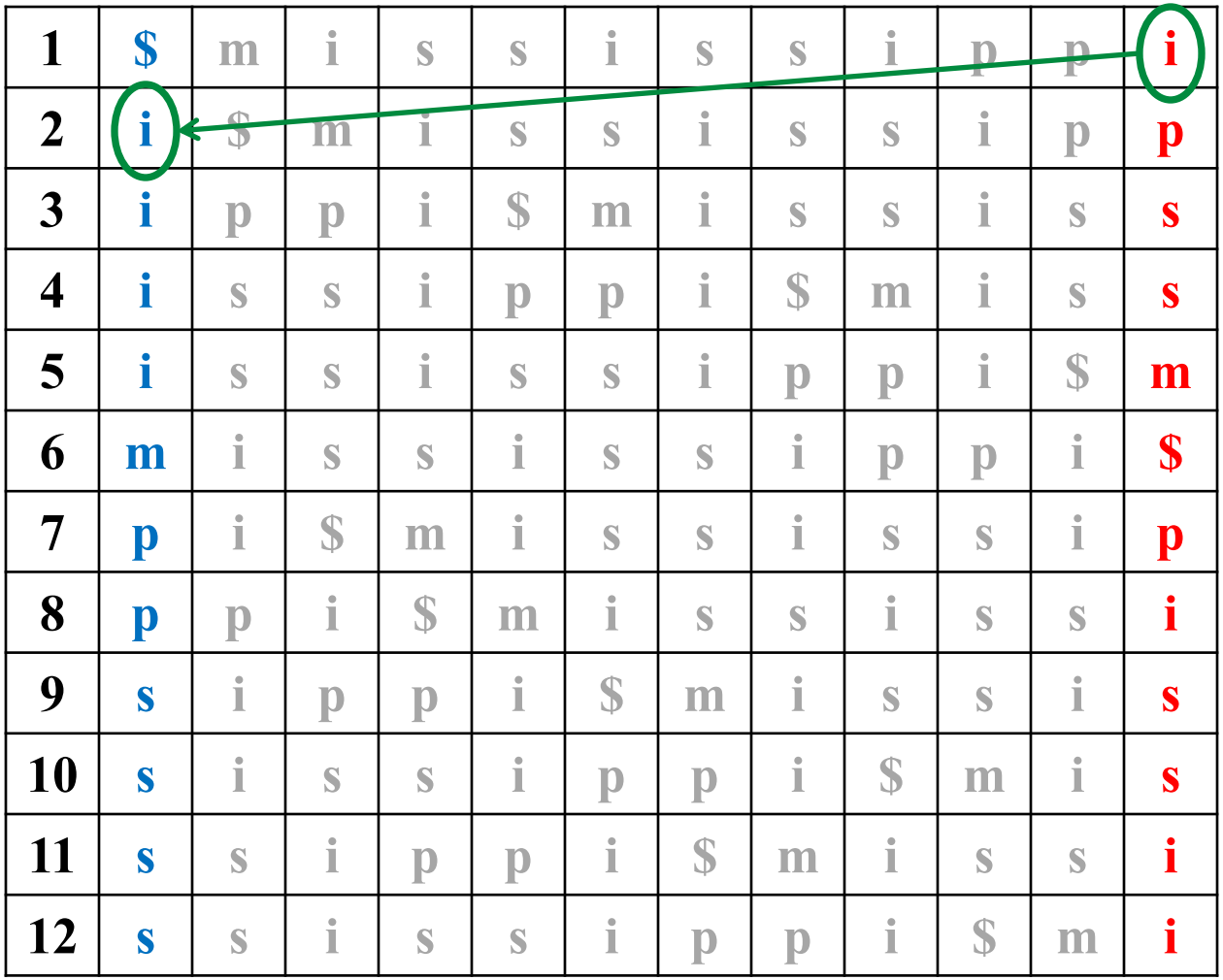
's' section →

1	$	m	i	s	s	i	s	s	i	p	p	i
2	i	$	m	i	s	s	i	s	s	i	p	p
3	i	p	p	i	$	m	i	s	s	i	s	s
4	i	s	s	i	p	p	i	$	m	i	s	s
5	i	s	s	i	s	s	i	p	p	i	$	m
6	m	i	s	s	i	s	s	i	p	p	i	$
7	p	i	$	m	i	s	s	i	s	s	i	p
8	p	p	i	$	m	i	s	s	i	s	s	i
9	s	i	p	p	i	$	m	i	s	s	i	s
10	s	i	s	s	i	p	p	i	$	m	i	s
11	s	s	i	p	p	i	$	m	i	s	s	i
12	s	s	i	s	s	i	p	p	i	$	m	i

Ferragina-Manzini Index

Reverse traversal

(1) i
(2) p
(7) p
(8) i
(3) s
(9) s
(11) i
(4) s
(10) s
(12) i
(5) m
(6) $



1	$	m	i	s	s	i	s	s	i	p	p	i
2	i	$	m	i	s	s	i	s	s	i	p	p
3	i	p	p	i	$	m	i	s	s	i	s	s
4	i	s	s	i	p	p	i	$	m	i	s	s
5	i	s	s	i	s	s	i	p	p	i	$	m
6	m	i	s	s	i	s	s	i	p	p	i	$
7	p	i	$	m	i	s	s	i	s	s	i	p
8	p	p	i	$	m	i	s	s	i	s	s	i
9	s	i	p	p	i	$	m	i	s	s	i	s
10	s	i	s	s	i	p	p	i	$	m	i	s
11	s	s	i	p	p	i	$	m	i	s	s	i
12	s	s	i	s	s	i	p	p	i	$	m	i

Ferragina-Manzini Index

Search *issi*

(1)-(12)

i (2)-(5)

si (9)-(10)

ssi (11)-(12)

issi (4)-(5)

1	**$**	m	i	s	s	i	s	s	i	p	p	**i**
2	**i**	$	m	i	s	s	i	s	s	i	p	**p**
3	**i**	p	p	i	$	m	i	s	s	i	s	**s**
4	**i**	s	s	i	p	p	i	$	m	i	s	**s**
5	**i**	s	s	i	s	s	i	p	p	i	$	**m**
6	**m**	i	s	s	i	s	s	i	p	p	i	**$**
7	**p**	i	$	m	i	s	s	i	s	s	i	**p**
8	**p**	p	i	$	m	i	s	s	i	s	s	**i**
9	**s**	i	p	p	i	$	m	i	s	s	i	**s**
10	**s**	i	s	s	i	p	p	i	$	m	i	**s**
11	**s**	s	i	p	p	i	$	m	i	s	s	**i**
12	**s**	s	i	s	s	i	p	p	i	$	m	**i**

Ferragina-Manzini Index
Search *pi*

(1)-(12)

i

pi

1	**$**	m	i	s	s	i	s	s	i	p	p	**i**
2	**i**	$	m	i	s	s	i	s	s	i	p	**p**
3	**i**	p	p	i	$	m	i	s	s	i	s	**s**
4	**i**	s	s	i	p	p	i	$	m	i	s	**s**
5	**i**	s	s	i	s	s	i	p	p	i	$	**m**
6	**m**	i	s	s	i	s	s	i	p	p	i	**$**
7	**p**	i	$	m	i	s	s	i	s	s	i	**p**
8	**p**	p	i	$	m	i	s	s	i	s	s	**i**
9	**s**	i	p	p	i	$	m	i	s	s	i	**s**
10	**s**	i	s	s	i	p	p	i	$	m	i	**s**
11	**s**	s	i	p	p	i	$	m	i	s	s	**i**
12	**s**	s	i	s	s	i	p	p	i	$	m	**i**