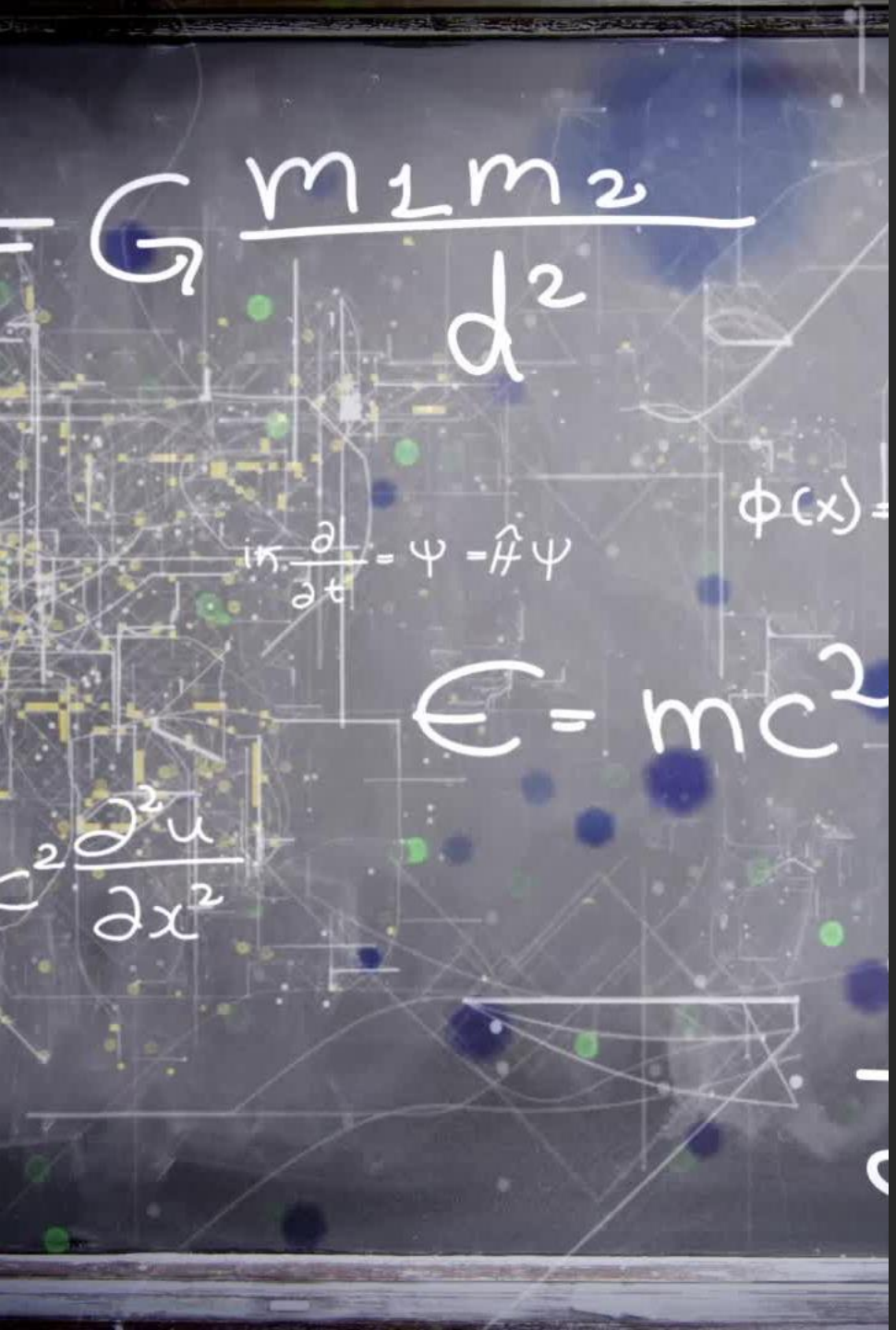
Burrows Wheeler Aligner (BWA)

SBS BIOHACKATHON || PYTHON



Scope

1. Introduction to Indexing
2. Suffix Array
3. Burrows Wheeler Aligner
4. Auxiliary Data Structures for BWA – FM-Index
5. Combining Workshop 1, 2 & 3!
6. Additional Resources

Background

NTU SBS Graduate 2020

Bioinformatics Specialist

- A*STAR Genome Institute of Singapore (GIS)
- Nalagenetics

Part Time Masters in BioMedical Data Science

Indexing

Grouping

Ordering

Suffix

What is a suffix?

A suffix is a **substring** at the end of a string of characters. For our purpose, suffixes are non-empty.

Suffixes of ‘Horse’

E

SE

RSE

ORSE

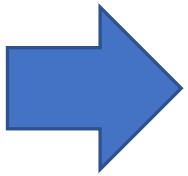
HORSE

Suffix Array

What is a Suffix Array?

A suffix array is an array which contains all the **sorted** suffixes of a string (eg. lexicographically)

Index	Suffix
0	camel
1	amel
2	mel
3	el
4	l



Index	Suffix
1	amel
0	camel
3	el
4	l
2	mel

Suffix Array

What is a Suffix Array?

The actual ‘suffix array’ is the array of sorted indices.

This provides a compressed representation of the sorted suffixes without actually needing to store the suffixes

Index	Suffix
1	amel
0	camel
3	el
4	l
2	mel

Suffix Array

Basic Code Implementation

Key Concepts:

- Loop
- Slicing
- Data Structure: Dictionary
- Sort The Dictionary (Lexicographically)
- Retrieve the Keys of the Dictionary (Indices)

Burrow Wheeler Aligner

Burrows-Wheeler Transform (or, BWT) is a **block compression algorithm** and is used in programs like bzip. The compression is **lossless** and allows for **fast querying**.

Burrow Wheeler Aligner

Let’s say our word of the day is:

MISSISSIPPI

Step 1:

Add a terminator symbol ($)

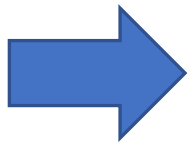
MISSISSIPPI$

Burrow Wheeler Aligner

Step 2:

Create a suffix array

1	mississippi$
2	ississippi$
3	ssissippi$
4	sissippi$
5	issippi$
6	ssippi$
7	sippi$
8	ippi$
9	ppi$
10	pi$
11	i$
12	$



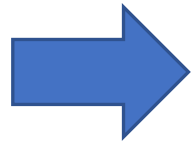
SA	
12	$
11	i$
8	ippi$
5	issippi$
2	ississippi$
1	mississippi$
10	pi$
9	ppi$
7	sippi$
4	sissippi$
6	ssippi$
3	ssissippi$

Burrow Wheeler Aligner

Step 3:

Burrows Wheeler Transform

SA	
12	$
11	i$
8	ippi$
5	issippi$
2	ississippi$
1	mississippi$
10	pi$
9	ppi$
7	sippi$
4	sissippi$
6	ssippi$
3	ssissippi$



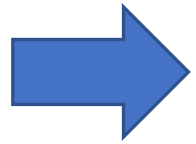
$mississippi
i$mississipp
ippi$
issippi$
ississippi$
mississippi$
pi$
ppi$
sippi$
sissippi$
ssippi$
ssissippi$

Burrow Wheeler Aligner

Step 3:

Burrows Wheeler Transform

SA		
12	$	$mississippi
11	i$	i$mississipp
8	ippi$	ippi$mississ
5	issippi$	issippi$miss
2	ississippi$	ississippi$m
1	mississippi$	mississippi$
10	pi$	pi$mississip
9	ppi$	ppi$mississi
7	sippi$	sippi$missis
4	sissippi$	sissippi$mis
6	ssippi$	ssippi$missi
3	ssissippi$	ssissippi$mi

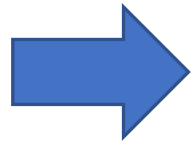


Burrow Wheeler Aligner

Step 3:

Burrows Wheeler Transform

SA		BWT
12	$mississippi	i
11	i$mississipp	p
8	ippi$mississ	s
5	issippi$miss	s
2	ississippi$m	m
1	mississippi$	$
10	pi$mississip	p
9	ppi$mississi	i
7	sippi$missis	s
4	sissippi$mis	s
6	ssippi$missi	i
3	ssissippi$mi	i

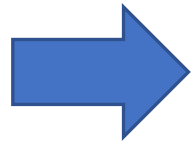


Burrow Wheeler Aligner

Step 3:

Burrows Wheeler Transform

SA	C	BWT
12	$mississippi	i
11	i$mississipp	p
8	ippi$mississ	s
5	issippi$miss	s
2	ississippi$m	m
1	mississippi$	$
10	pi$mississip	p
9	ppi$mississi	i
7	sippi$missis	s
4	sissippi$mis	s
6	ssippi$missi	i
3	ssissippi$mi	i

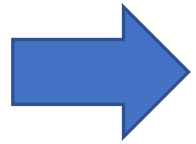


SA	BWT	C		C
12	i	$	0	$:0
11	p	i	1	i:1
8	s	i	2	m:5
5	s	i	3	p:6
2	m	i	4	s:8
1	$	m	5	
10	p	p	6	
9	i	p	7	
7	s	s	8	
4	s	s	9	
6	i	s	10	
3	i	s	11	

Burrow Wheeler Aligner

Essentially What We Need

SA	C	BWT
12	$mississippi	i
11	i$mississipp	p
8	ippi$mississ	s
5	issippi$miss	s
2	ississippi$m	m
1	mississippi$	$
10	pi$mississip	p
9	ppi$mississi	i
7	sippi$missis	s
4	sissippi$mis	s
6	ssippi$missi	i
3	sissippi$mi	i

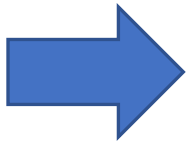


SA	BWT	C
12	i	$:0
11	p	i:1
8	s	m:5
5	s	p:6
2	m	s:8
1	$	
10	p	
9	i	
7	s	
4	s	
6	i	
3	i	

Burrow Wheeler Aligner

When we index the BWT!!

SA	C	BWT
12	$ mississippi	$_0$
11	i$_0$ mississipp	i$_1$
8	i$_1$ ppi$mississ	p$_0$
5	i$_2$ ssippi$miss	i$_2$
2	i$_3$ ssissippi$m	i$_3$
1	m$_0$ ississippi$	m$_0$
10	p$_0$ i$mississi	s$_0$
9	p$_1$ pi$mississ	s$_1$
7	s$_0$ ippi$missi	p$_1$
4	s$_1$ issippi$mi	s$_2$
6	s$_2$ sippi$miss	s$_3$
3	s$_3$ sissippi$m	i$_4$

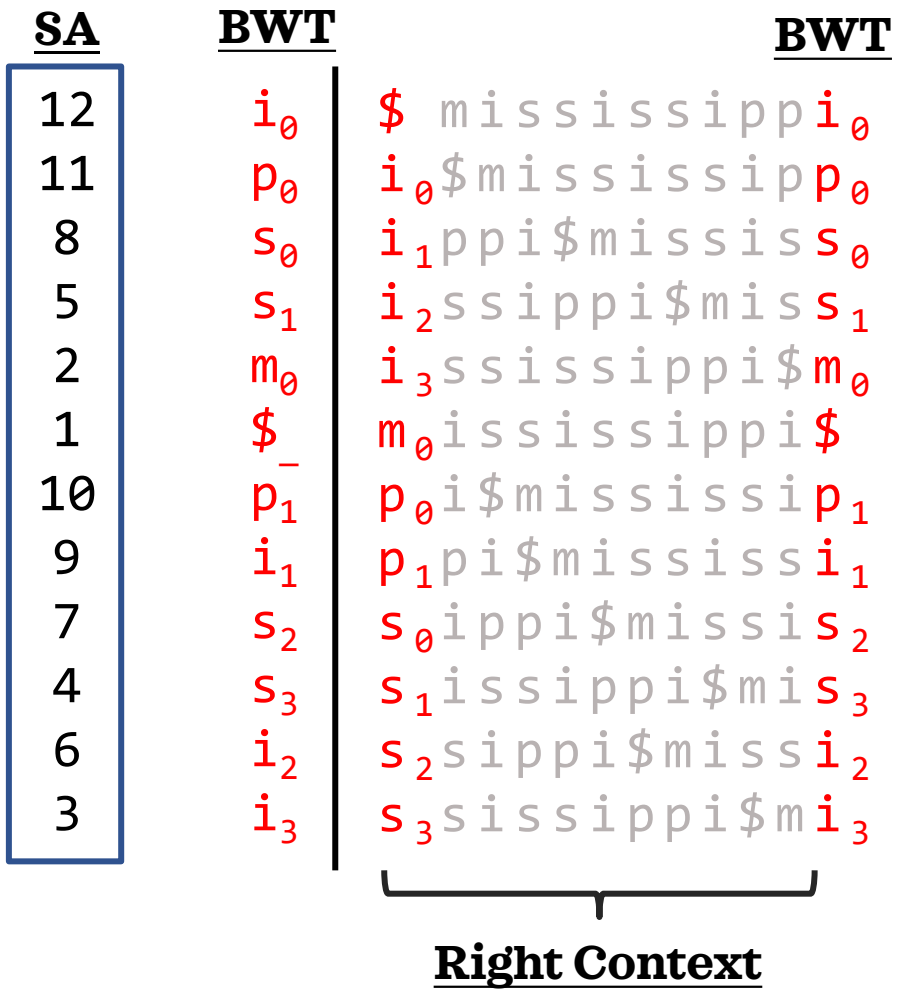


SA	BWT	C
12	i	$:0
11	p	i:1
8	s	m:5
5	s	p:6
2	m	s:8
1	$	
10	p	
9	i	
7	s	
4	s	
6	i	
3	i	

Burrow Wheeler Aligner

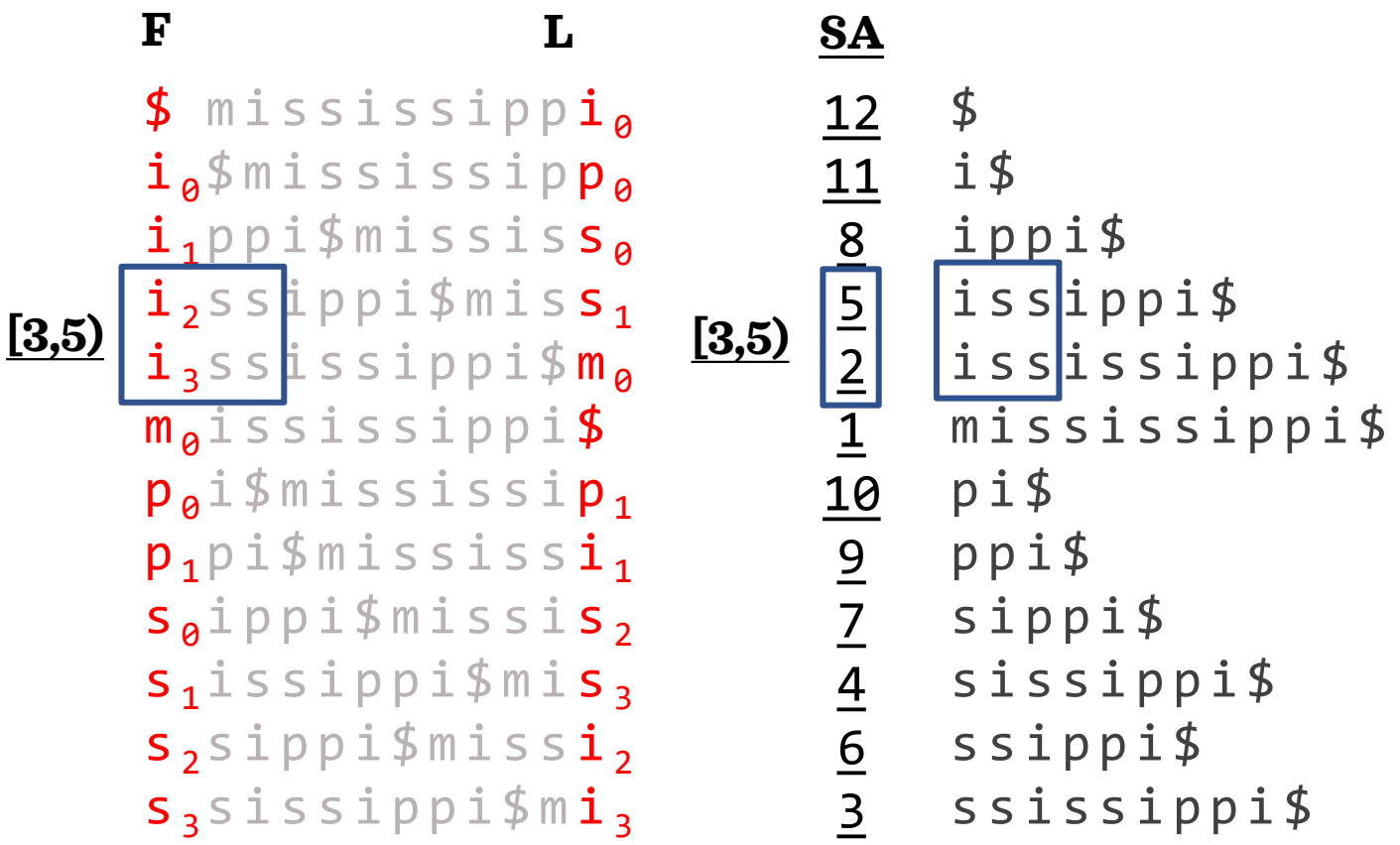
Step 3:

Burrows Wheeler Transform



BWA Querying

Let's say Pattern = iss



Unlike Suffix Array, we don't immediately know where the matches are in Text

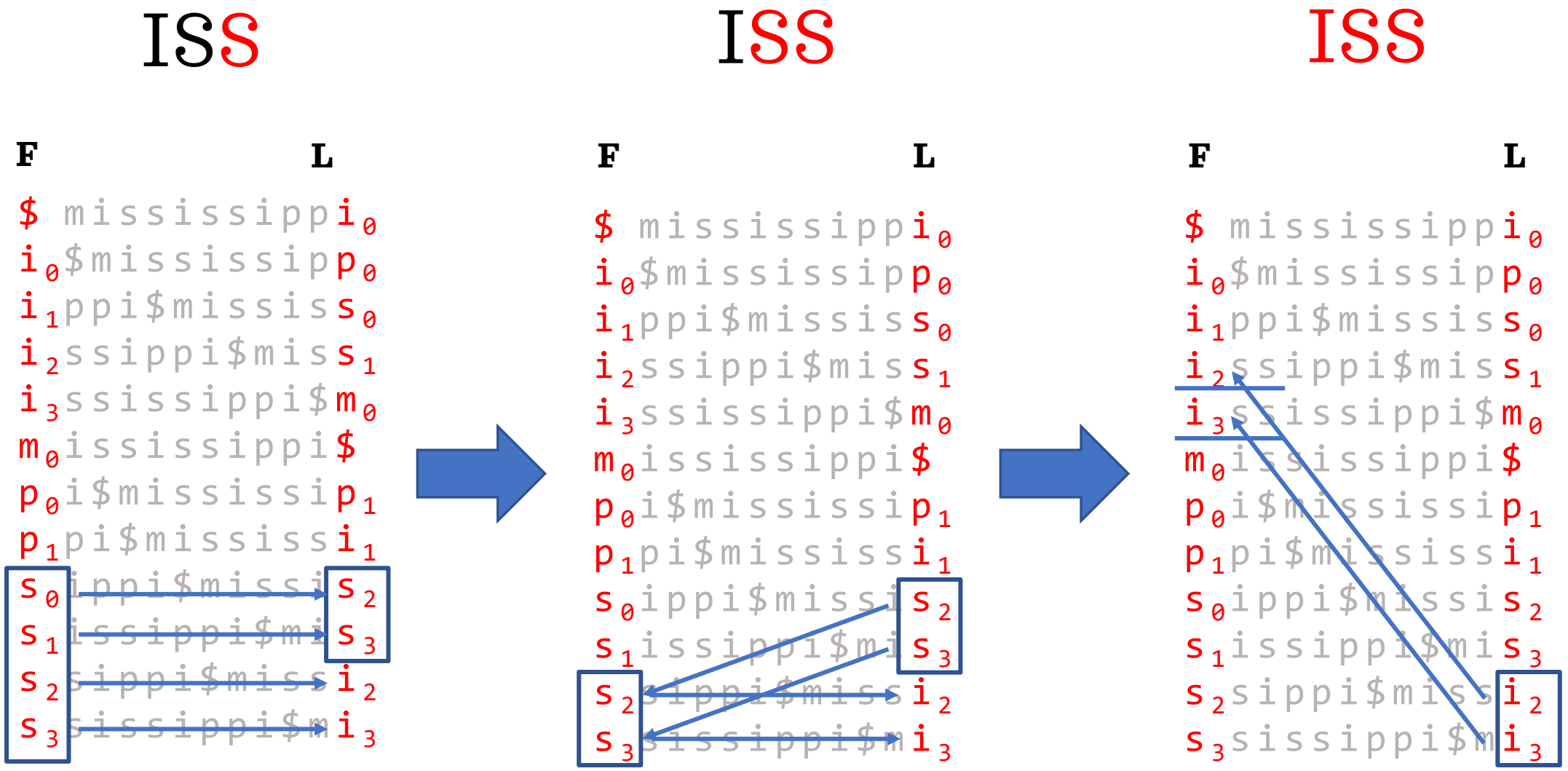
BWA Querying

This is the information we have now

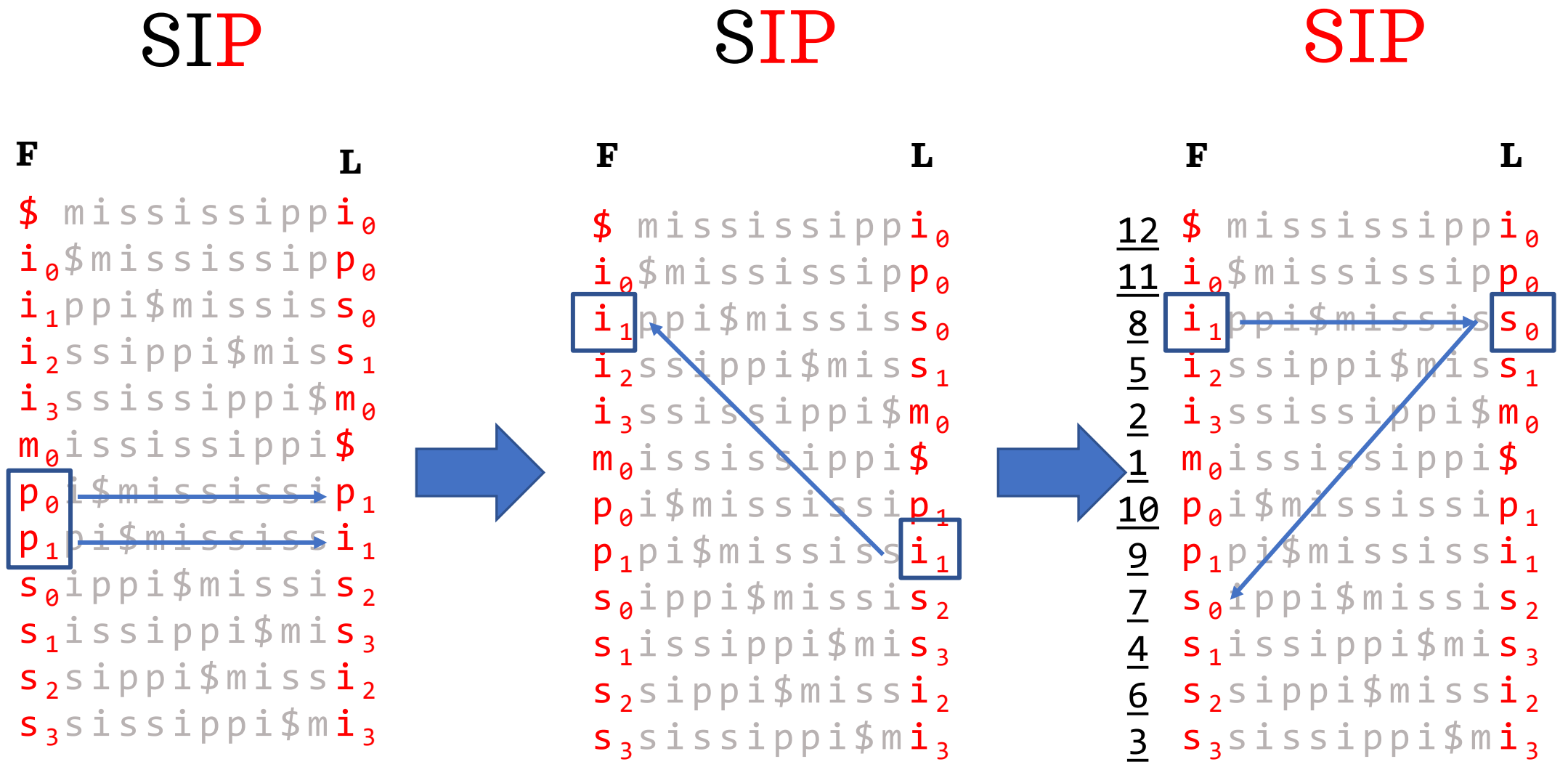
How do we query?

SA	BWT	C
12	i	$:0
11	p	i:1
8	s	m:5
5	s	p:6
2	m	s:8
1	$	
10	p	
9	i	
7	s	
4	s	
6	i	
3	i	

Pattern = iss


ISS
ISS
ISS
F
L
$ mississippi i0
i0 $mississip p0
i1 ppi$missis s0
i2 ssippi$mis s1
i3 ssissippi$m m0
m0 ississippi$
p0 i$mississi p1
p1 pi$mississ i1
s0 ippi$missi s2
s1 issippi$mi s3
s2 sippi$miss i2
s3 sissippi$m i3

Pattern = SIP


SIP
SIP
SIP
F
L
$ mississippi i0
i0 $mississipp p0
i1 ppi$mississ s0
i2 ssippi$miss s1
i3 ssissippi$m m0
m0 ississippi$
p0 i$mississi p1
p1 pi$mississ i1
s0 ippi$missi s2
s1 issippi$mis s3
s2 sippi$miss i2
s3 sissippi$m i3
12
11
8
5
2
1
10
9
7
4
6
3

Now Its your Turn!

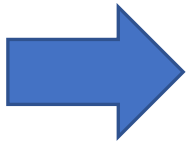
Let’s say we have this information; can you reform the word? Hint, first step is to order the letters in BWT Lexicographically

SA	BWT
15	N
7	H
10	K
3	S
1	S
8	A
6	O
12	T
4	B
9	C
14	O
5	I
13	H
2	B
0	$
11	A

Now Its your Turn!

Let’s say we have this information; can you reform the word? Hint, first step is to order the letters in BWT Lexicographically

SA	BWT
15	N
7	H
10	K
3	S
1	S
8	A
6	O
12	T
4	B
9	C
14	O
5	I
13	H
2	B
0	$
11	A



$	N
A	H
A	K
B	S
B	S
C	A
H	O
H	T
I	B
K	C
N	O
O	I
O	H
S	B
S	$
T	A

Now Its your Turn!

Let’s say we have this information; can you help query the string “GAT”?

SA	BWT
10	T
2	G
6	G
4	T
1	G
5	C
0	$
8	T
9	G
3	A
7	A

Auxiliary Data Structure

Fm-Index!

FM-Index

OCC Data Structure – Store BWA details

	BWT	I	M	P	S
$mississippi	i	1	0	0	0
i$mississipp	p	1	0	1	0
ippi$mississ	s				
issippi$miss	s				
ississippi$m	m				
mississippi$	$				
pi$mississip	p				
ppi$mississi	i				
sippi$missis	s				
sissippi$mis	s				
ssippi$missi	i				
ssissippi$mi	i				

FM-Index

OCC Data Structure

	BWT	I	M	P	S
$mississippi	i	1	0	0	0
i$mississipp	p	1	0	1	0
ippi$missis	s	1	0	1	1
issippi$miss	s	1	0	1	2
ississippi$m	m	1	1	1	2
mississippi$	$	1	1	1	2
pi$mississip	p	1	1	2	2
ppi$mississ	i	2	1	2	2
sippi$missis	s	2	1	2	3
sissippi$mis	s	2	1	2	4
ssippi$missi	i	3	1	2	4
ssissippi$mi	i	4	1	2	4

OCC Data Structure: Pattern = iss

F	BWT	I	M	P	S
$	i	1	0	0	0
i	p	1	0	1	0
i	s	1	0	1	1
i	s	1	0	1	2
i	m	1	1	1	2
m	$	1	1	1	2
p	p	1	1	2	2
p	i	2	1	2	2
s	s	2	1	2	3
s	s	2	1	2	4
s	i	3	1	2	4
s	i	4	1	2	4

F	BWT	I	M	P	S
$	i	1	0	0	0
i	p	1	0	1	0
i	s	1	0	1	1
i	s	1	0	1	2
i	m	1	1	1	2
m	$	1	1	1	2
p	p	1	1	2	2
p	i	2	1	2	2
s	s	2	1	2	3
s	s	2	1	2	4
s	i	3	1	2	4
s	i	4	1	2	4

F	BWT	I	M	P	S
$	i	1	0	0	0
i	p	1	0	1	0
i	s	1	0	1	1
i	s	1	0	1	2
i	m	1	1	1	2
m	$	1	1	1	2
p	p	1	1	2	2
p	i	2	1	2	2
s	s	2	1	2	3
s	s	2	1	2	4
s	i	3	1	2	4
s	i	4	1	2	4

BWA FM-Index

Code Implementation!

Exact Matching

Pipeline example

Burrows Wheeler Aligner

Bit By Bit

BackTracking [CIGAR score]

Edit Distance

4 situations:

1. Match (Move diagonally)
2. Substitution (Move diagonally + 1)
3. Insertion (Move right + 1)
4. Deletion (Move down + 1)

} Find the lowest score

Query: TGATA
Hit : TGGACT

		T	G	A	T	A
	0	1	2	3	4	5
T	1	0				
G	2					
G	3					
A	4					
C	5					
T	6					

		T	G	A	T	A
	0	1	2	3	4	5
T	1	0	1	2	3	4
G	2	1	0	1	2	3
G	3	2	1	1	2	3
A	4	3	2	1	2	2
C	5	4	3	2	2	3
T	6	4	3	3	2	3

References

Burrows-Wheeler Indexing By Ben Langmead:
https://www.youtube.com/watch?v=5G2Db41pSHE&list=PL2mpR0RYFQsADmYpW2YWBrXJZ_6EL_3nu

BWA Paper:

https://academic.oup.com/bioinformatics/article/25/14/1754/225615

GitHub Repository:

https://github.com/micro-irfan/SBSBiohackathon

Where to Next?

Rosalind Bioinformatics Challenge

Coursera Bioinformatics (UC San Diego / John Hopkins)

William Fiset Data Structures Course

RNA-seq Youtube Code Along

Find a lab attachment

Kaggle (For machine learning; data analyst / scientist)

Build a github repository!

Code code code!