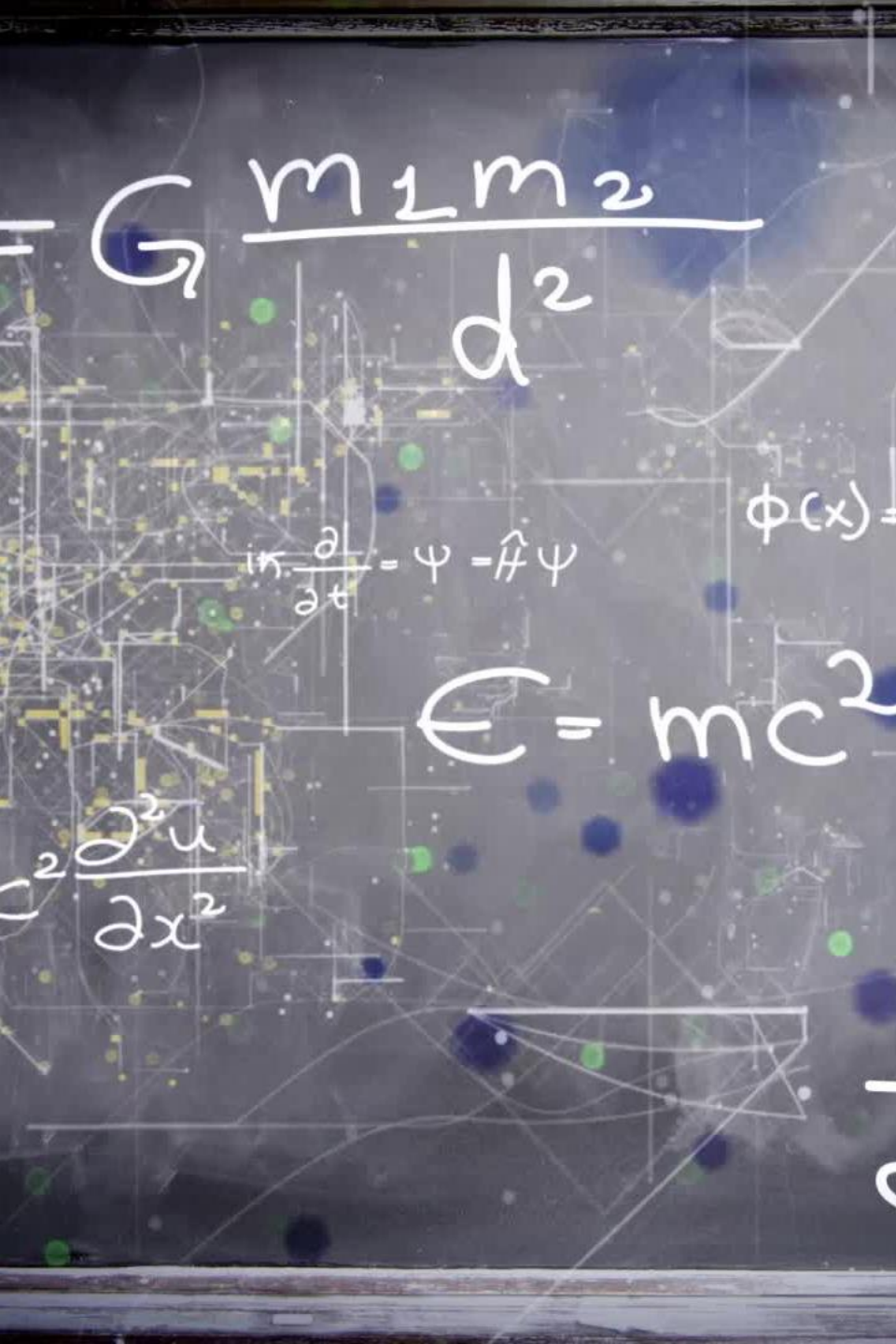




Burrows Wheeler Aligner (BWA)

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

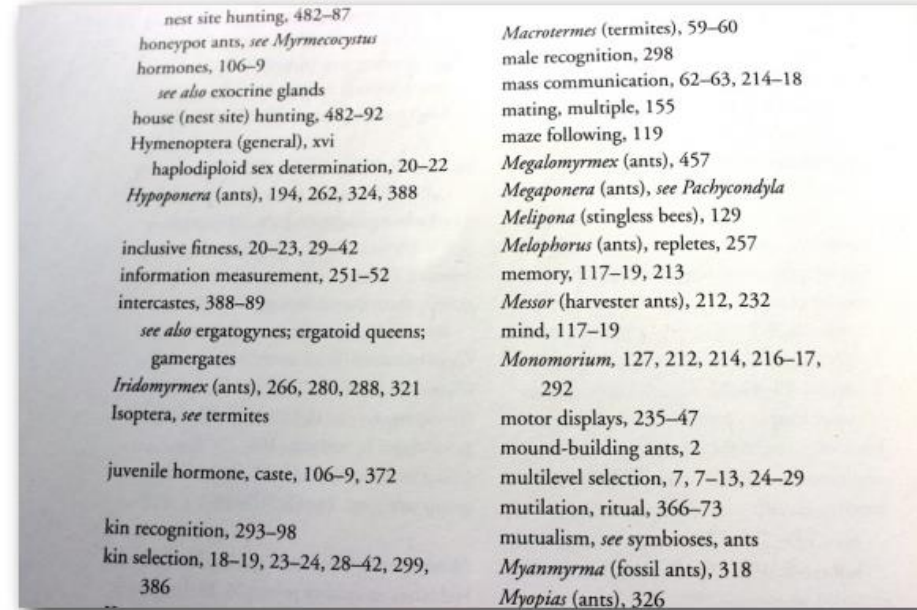
Part Time Masters in BioMedical Data Science



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

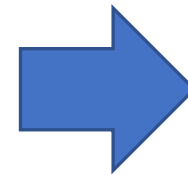
HORSE

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

| | <u>SA</u> |
|-----------------|-----------------|
| 1 mississippi\$ | 12 \$ |
| 2 ississippi\$ | 11 i\$ |
| 3 ssissippi\$ | 8 ippi\$ |
| 4 sissippi\$ | 5 issippi\$ |
| 5 issippi\$ | 2 ississippi\$ |
| 6 ssippi\$ | 1 mississippi\$ |
| 7 sippi\$ | 10 pi\$ |
| 8 ippi\$ | 9 ppi\$ |
| 9 ppi\$ | 7 sippi\$ |
| 10 pi\$ | 4 sissippi\$ |
| 11 i\$ | 6 ssippi\$ |
| 12 \$ | 3 ssissippi\$ |

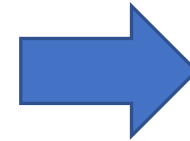
Burrow Wheeler Aligner

Step 3:

Burrows Wheeler Transform

SA

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



| | |
|---------------|-------------|
| \$ | mississippi |
| i\$ | mississippi |
| ippi\$ | |
| issippi\$ | |
| issippiippi\$ | |
| mississippi\$ | |
| pi\$ | |
| ppi\$ | |
| sippi\$ | |
| sissippi\$ | |
| ssippi\$ | |
| ssissippi\$ | |

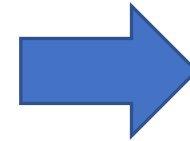
Burrow Wheeler Aligner

Step 3:

Burrows Wheeler Transform

SA

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



| | |
|---------------|-------------|
| \$ | mississippi |
| i\$ | mississipp |
| ippi\$ | mississ |
| issippi\$ | miss |
| issippiippi\$ | m |
| mississippi\$ | |
| pi\$ | mississip |
| ppi\$ | mississi |
| sippi\$ | missis |
| sissippi\$ | mis |
| ssippi\$ | missi |
| ssissippi\$ | mi |

Burrow Wheeler Aligner

Step 3:

Burrows Wheeler Transform

| <u>SA</u> | | <u>BWT</u> |
|-----------|------------------------|------------|
| 12 | \$mississipp <i>i</i> | i |
| 11 | i\$mississipp <i>p</i> | p |
| 8 | ippi\$mississ <i>s</i> | s |
| 5 | issippi\$miss <i>s</i> | s |
| 2 | ississipp <i>i</i> \$m | m |
| 1 | mississipp <i>i</i> \$ | \$ |
| 10 | pi\$mississip <i>p</i> | p |
| 9 | ppi\$mississ <i>i</i> | i |
| 7 | sippi\$missis <i>s</i> | s |
| 4 | sisippi\$miss <i>s</i> | s |
| 6 | ssippi\$miss <i>i</i> | i |
| 3 | ssissipp <i>i</i> \$m | i |

Burrow Wheeler Aligner

Step 3:

Burrows Wheeler Transform

| <u>SA</u> | <u>C</u> | <u>BWT</u> | <u>SA</u> | <u>BWT</u> | <u>C</u> | <u>C</u> |
|-----------|----------------|------------|-----------|------------|----------|----------|
| 12 | \$mississippi | i | 12 | i | \$ 0 | \$:0 |
| 11 | i\$mississipp | p | 11 | p | i 1 | i:1 |
| 8 | ippi\$mississ | s | 8 | s | i 2 | m:5 |
| 5 | issippi\$miss | s | 5 | s | i 3 | p:6 |
| 2 | ississippi\$m | m | 2 | m | i 4 | s:8 |
| 1 | mississippi\$ | \$ | 1 | \$ | m 5 | |
| 10 | pi\$mississip | p | 10 | p | p 6 | |
| 9 | ppi\$mississi | i | 9 | i | p 7 | |
| 7 | sippi\$missis | s | 7 | s | s 8 | |
| 4 | ssissippi\$mis | s | 4 | s | s 9 | |
| 6 | ssippi\$missi | i | 6 | i | s 10 | |
| 3 | ssissippi\$m | i | 3 | i | s 11 | |

Burrow Wheeler Aligner

Essentially What We Need

| <u>SA</u> | <u>C</u> | <u>BWT</u> | <u>SA</u> | <u>BWT</u> | <u>C</u> |
|-----------|----------------|------------|-----------|------------|----------|
| 12 | \$mississippi | i | 12 | i | \$:0 |
| 11 | i\$mississipp | p | 11 | p | i:1 |
| 8 | ippi\$mississ | s | 8 | s | m:5 |
| 5 | issippi\$miss | s | 5 | s | p:6 |
| 2 | ississippi\$m | m | 2 | m | s:8 |
| 1 | mississippi\$ | \$ | 1 | \$ | |
| 10 | pi\$mississip | p | 10 | p | |
| 9 | ppi\$mississi | i | 9 | i | |
| 7 | sippi\$missis | s | 7 | s | |
| 4 | ssissippi\$mis | s | 4 | s | |
| 6 | ssippi\$missi | i | 6 | i | |
| 3 | ssissippi\$mi | i | 3 | i | |

Burrow Wheeler Aligner

When we index the BWT!!

| <u>SA</u> | <u>C</u> | <u>BWT</u> | <u>SA</u> | <u>BWT</u> | <u>C</u> |
|-----------|-----------------------|-------------------------------------------|-----------|------------|----------|
| 12 | \$ | mississippi <i>i</i> ₀ | 12 | <i>i</i> | \$:0 |
| 11 | <i>i</i> ₀ | \$mississipp <i>p</i> ₀ | 11 | <i>p</i> | i:1 |
| 8 | <i>i</i> ₁ | ppi\$missis <i>s</i> ₀ | 8 | <i>s</i> | m:5 |
| 5 | <i>i</i> ₂ | ssippi\$mis <i>s</i> ₁ | 5 | <i>s</i> | p:6 |
| 2 | <i>i</i> ₃ | ssissipp <i>i</i> \$m ₀ | 2 | <i>m</i> | s:8 |
| 1 | <i>m</i> ₀ | ississipp <i>i</i> \$ | 1 | <i>\$</i> | |
| 10 | <i>p</i> ₀ | i\$mississ <i>i</i> <i>p</i> ₁ | 10 | <i>p</i> | |
| 9 | <i>p</i> ₁ | pi\$mississ <i>i</i> ₁ | 9 | <i>i</i> | |
| 7 | <i>s</i> ₀ | ippi\$missis <i>s</i> ₂ | 7 | <i>s</i> | |
| 4 | <i>s</i> ₁ | issippi\$mis <i>s</i> ₃ | 4 | <i>s</i> | |
| 6 | <i>s</i> ₂ | sippi\$miss <i>i</i> ₂ | 6 | <i>i</i> | |
| 3 | <i>s</i> ₃ | ssissippi\$mi <i>i</i> ₃ | 3 | <i>i</i> | |

Burrow Wheeler Aligner

Step 3:

Burrows Wheeler Transform

| <u>SA</u> | <u>BWT</u> | <u>BWT</u> |
|-----------|------------|--------------------------------|
| 12 | i_0 | \$ mississippi i_0 |
| 11 | p_0 | i_0 \$mississipp p_0 |
| 8 | s_0 | i_1 ppi\$mississ s_0 |
| 5 | s_1 | i_2 ssippi\$miss s_1 |
| 2 | m_0 | i_3 ssissippi i_3 \$ m_0 |
| 1 | $\$$ | m_0 ississippi \$ |
| 10 | p_1 | p_0 i\$mississipp p_1 |
| 9 | i_1 | p_1 pi\$mississ i_1 |
| 7 | s_2 | s_0 ippi\$mississ s_2 |
| 4 | s_3 | s_1 issippi\$miss s_3 |
| 6 | i_2 | s_2 sippi\$miss i_2 |
| 3 | i_3 | s_3 sissippi\$m i_3 |

Right Context

BWA Querying

Let's say Pattern = iss

| F | L | SA |
|-----------------------------------------------------|----------|------------------------------------|
| \$ mississippi ₀ | | <u>12</u> \$ |
| ₀ \$mississipp ₀ | | <u>11</u> i\$ |
| ₁ ppi\$mississ ₀ | | <u>8</u> ippi\$ |
| <u>[3,5)</u> ₂ ssippi\$miss ₁ | | <u>[3,5)</u> <u>5</u> issippi\$ |
| ₃ ssissippim ₀ | | <u>2</u> ississippim ₀ |
| m ₀ ississippim ₀ | | <u>1</u> mississippim ₀ |
| p ₀ i\$mississipp ₁ | | <u>10</u> pi\$ |
| p ₁ pi\$mississipp ₁ | | <u>9</u> ppi\$ |
| s ₀ ippi\$mississ ₂ | | <u>7</u> sippi\$ |
| s ₁ issippimiss ₃ | | <u>4</u> sissippimiss ₃ |
| s ₂ ssippimiss ₂ | | <u>6</u> ssippimiss ₂ |
| s ₃ ssissippim ₃ | | <u>3</u> ssissippim ₃ |

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

BWA Querying

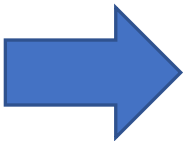
This is the information we have now
How do we query?

| <u>SA</u> | <u>BWT</u> | <u>C</u> |
|-----------|------------|----------|
| 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

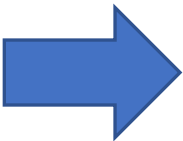
ISS

| F | | L |
|-------|--------------|-------|
| \$ | mississippi | i_0 |
| i_0 | \$mississipp | p_0 |
| i_1 | ppi\$mississ | s_0 |
| i_2 | ssippi\$miss | s_1 |
| i_3 | ssissippip | m_0 |
| m_0 | ississippip | i_1 |
| p_0 | i\$mississip | p_1 |
| p_1 | pi\$mississ | i_2 |
| s_0 | ippi\$missi | s_2 |
| s_1 | issippi\$mi | s_3 |
| s_2 | sippi\$miss | i_3 |
| s_3 | ssissippi\$m | i_4 |



ISS

| F | | L |
|-------|--------------|-------|
| \$ | mississippi | i_0 |
| i_0 | \$mississipp | p_0 |
| i_1 | ppi\$mississ | s_0 |
| i_2 | ssippi\$miss | s_1 |
| i_3 | ssissippip | m_0 |
| m_0 | ississippip | i_1 |
| p_0 | i\$mississip | p_1 |
| p_1 | pi\$mississ | i_2 |
| s_0 | ippi\$missi | s_2 |
| s_1 | issippi\$mi | s_3 |
| s_2 | sippi\$miss | i_3 |
| s_3 | ssissippi\$m | i_4 |



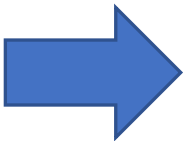
ISS

| F | | L |
|-------|--------------|-------|
| \$ | mississippi | i_0 |
| i_0 | \$mississipp | p_0 |
| i_1 | ppi\$mississ | s_0 |
| i_2 | ssippi\$miss | s_1 |
| i_3 | ssissippip | m_0 |
| m_0 | ississippip | i_1 |
| p_0 | i\$mississip | p_1 |
| p_1 | pi\$mississ | i_2 |
| s_0 | ippi\$missi | s_2 |
| s_1 | issippi\$mi | s_3 |
| s_2 | sippi\$miss | i_3 |
| s_3 | ssissippi\$m | i_4 |

Pattern = SIP

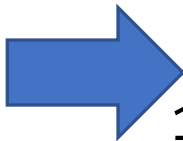
SIP

| F | | L |
|-------|--------------|-------|
| \$ | mississippi | i_0 |
| i_0 | \$mississipp | p_0 |
| i_1 | ppi\$missis | s_0 |
| i_2 | ssippi\$mis | s_1 |
| i_3 | ssissippip | m_0 |
| m_0 | ississippip | \$ |
| p_0 | i\$mississip | p_1 |
| p_1 | pi\$mississ | i_1 |
| s_0 | ippi\$missis | s_2 |
| s_1 | issippip\$mi | s_3 |
| s_2 | sippip\$miss | i_2 |
| s_3 | ssissippip | m_1 |



SIP

| F | | L |
|-------|--------------|-------|
| \$ | mississippi | i_0 |
| i_0 | \$mississipp | p_0 |
| i_1 | ppi\$missis | s_0 |
| i_2 | ssippi\$mis | s_1 |
| i_3 | ssissippip | m_0 |
| m_0 | ississippip | \$ |
| p_0 | i\$mississip | p_1 |
| p_1 | pi\$mississ | i_1 |
| s_0 | ippi\$missis | s_2 |
| s_1 | issippip\$mi | s_3 |
| s_2 | sippip\$miss | i_2 |
| s_3 | ssissippip | m_1 |



SIP

| | F | | L |
|-----------|-------|--------------|-------|
| <u>12</u> | \$ | mississippi | i_0 |
| <u>11</u> | i_0 | \$mississipp | p_0 |
| <u>8</u> | i_1 | ppi\$missis | s_0 |
| <u>5</u> | i_2 | ssippi\$mis | s_1 |
| <u>2</u> | i_3 | ssissippip | m_0 |
| <u>1</u> | m_0 | ississippip | \$ |
| <u>10</u> | p_0 | i\$mississip | p_1 |
| <u>9</u> | p_1 | pi\$mississ | i_1 |
| <u>7</u> | s_0 | ippi\$missis | s_2 |
| <u>4</u> | s_1 | issippip\$mi | s_3 |
| <u>6</u> | s_2 | sippip\$miss | i_2 |
| <u>3</u> | s_3 | ssissippip | m_1 |

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**

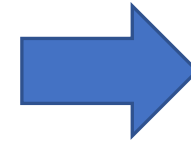
| | |
|----|----|
| 15 | N |
| 7 | H |
| 10 | K |
| 3 | S |
| 1 | S |
| 8 | A |
| 6 | O |
| 12 | T |
| 4 | B |
| 9 | C |
| 14 | O |
| 5 | I |
| 13 | H |
| 2 | B |
| 0 | \$ |
| 11 | A |

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

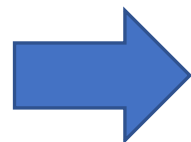
SA **BWT**

| | |
|----|----|
| 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 ₁ |
| A ₁ | H ₁ |
| A ₂ | K ₁ |
| B ₁ | S ₁ |
| B ₂ | S ₂ |
| C ₁ | A ₁ |
| H ₁ | O ₁ |
| H ₂ | T ₁ |
| I ₁ | B ₁ |
| K ₁ | C ₁ |
| N ₁ | O ₂ |
| O ₁ | I ₁ |
| O ₂ | H ₂ |
| S ₁ | B ₂ |
| S ₂ | \$ |
| T ₁ | A ₂ |

| N | | O | | H | | T | | A | |
|----|----|----|----|----------------|----------------|----------------|----|----|----|
| \$ | N | \$ | N | \$ | N | \$ | N | \$ | N |
| A | H | A | H | A | H | A | H | A | H |
| A | K | A | K | A | K | A | K | A | K |
| B | S | B | S | B | S | B | S | B | S |
| B | S | B | S | B | S | B | S | B | S |
| C | A | C | A | C | A | C | A | C | A |
| H | O | H | O | H | O | H | O | H | O |
| H | T | H | T | H | T | H ₂ | T | H | T |
| I | B | I | B | I | B | I | B | I | B |
| K | C | K | C | K | C | K | C | K | C |
| N | O | N | O | N | O | N | O | N | O |
| O | I | O | I | O | I | O | I | O | I |
| O | H | O | H | O ₂ | H ₂ | O | H | O | H |
| S | B | S | B | S | B | S | B | S | B |
| S | \$ | S | \$ | S | \$ | S | \$ | S | \$ |
| T | A | T | A | T | A | T | A | T | A |



\$ N

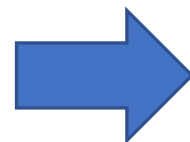
N O₂

O₂ H₂

H₂ T

T A

| N | | O | | H | | T | | A | |
|----|----|----|----|----------------|----------------|----------------|----|----|----|
| \$ | N | \$ | N | \$ | N | \$ | N | \$ | N |
| A | H | A | H | A | H | A | H | A | H |
| A | K | A | K | A | K | A | K | A | K |
| B | S | B | S | B | S | B | S | B | S |
| B | S | B | S | B | S | B | S | B | S |
| C | A | C | A | C | A | C | A | C | A |
| H | O | H | O | H | O | H | O | H | O |
| H | T | H | T | H | T | H ₂ | T | H | T |
| I | B | I | B | I | B | I | B | I | B |
| K | C | K | C | K | C | K | C | K | C |
| N | O | N | O | N | O | N | O | N | O |
| O | I | O | I | O | I | O | I | O | I |
| O | H | O | H | O ₂ | H ₂ | O | H | O | H |
| S | B | S | B | S | B | S | B | S | B |
| S | \$ | S | \$ | S | \$ | S | \$ | S | \$ |
| T | A | T | A | T | A | T | A | T | A |



SBSBIOHACKATHON\$

\$NOHTAKCAHOIBSBS

Now Its your
Turn!

Let's say we have this information; can you help query the string "GAT"?

| <u>SA</u> | <u>BWT</u> |
|-----------|------------|
| 10 | T |
| 2 | G |
| 6 | G |
| 4 | T |
| 1 | G |
| 5 | C |
| 0 | \$ |
| 8 | T |
| 9 | G |
| 3 | A |
| 7 | A |

Now Its your
Turn!

Let's say we have this information; can you help query the string "GAT"?

| <u>SA</u> | <u>BWT</u> | <u>C</u> | <u>BWT</u> |
|-----------|------------|----------------|----------------|
| 10 | T | \$ | T ₁ |
| 2 | G | A ₁ | G ₁ |
| 6 | G | A ₂ | G ₂ |
| 4 | T | C ₁ | T ₂ |
| 1 | G | G ₁ | G ₃ |
| 5 | C | G ₂ | C ₁ |
| 0 | \$ | G ₃ | \$ |
| 8 | T | G ₄ | T ₃ |
| 9 | G | T ₁ | G ₄ |
| 3 | A | T ₂ | A ₁ |
| 7 | A | T ₃ | A ₂ |

Let's say we have this information;
can you help query the string "GAT"?

| <u>C</u> | <u>BWT</u> |
|----------------|----------------|
| \$ | T ₁ |
| A ₁ | G ₁ |
| A ₂ | G ₂ |
| C ₁ | T ₂ |
| G ₁ | G ₃ |
| G ₂ | C ₁ |
| G ₃ | \$ |
| G ₄ | T ₃ |
| T ₁ | G ₄ |
| T ₂ | A ₁ |
| T ₃ | A ₂ |

| <u>C</u> | <u>BWT</u> |
|----------------|----------------|
| \$ | T ₁ |
| A ₁ | G ₁ |
| A ₂ | G ₂ |
| C ₁ | T ₂ |
| G ₁ | G ₃ |
| G ₂ | C ₁ |
| G ₃ | \$ |
| G ₄ | T ₃ |
| T ₁ | G ₄ |
| T ₂ | A ₁ |
| T ₃ | A ₂ |

| <u>C</u> | <u>BWT</u> |
|----------------|----------------|
| \$ | T ₁ |
| A ₁ | G ₁ |
| A ₂ | G ₂ |
| C ₁ | T ₂ |
| G ₁ | G ₃ |
| G ₂ | C ₁ |
| G ₃ | \$ |
| G ₄ | T ₃ |
| T ₁ | G ₄ |
| T ₂ | A ₁ |
| T ₃ | A ₂ |

GGATCGATGT

Now Its your
Turn!

Let's say we have this information; can you help query the string "GAT"?

| <u>SA</u> | <u>BWT</u> | |
|-----------|------------|--------------|
| 10 | T | \$GGATCGATGT |
| 2 | G | ATCGATGT\$GG |
| 6 | G | ATGT\$GGATCG |
| 4 | T | CGATGT\$GGAT |
| 1 | G | GATCGATGT\$G |
| 5 | C | GATGT\$GGATC |
| 0 | \$ | GGATCGATGT\$ |
| 8 | T | GT\$GGATCGAT |
| 9 | G | T\$GGATCGATG |
| 3 | A | TCGATGT\$GGA |
| 7 | A | TGT\$GGATCGA |

Auxiliary
Data
Structure

Fm-Index!

FM-Index

OCC Data Structure – Store BWA details

| | BWT | I | M | P | S |
|-----------------------------------------|-----|----------|---|----------|----------|
| \$ mississippi i | i | 1 | 0 | 0 | 0 |
| i \$mississipp p | p | 1 | 0 | 1 | 0 |
| i pp i \$mississ s | s | 1 | 0 | 1 | 1 |
| i ssipp i \$miss s | s | 1 | 0 | 1 | 2 |
| i ssissippi i \$m | m | 1 | 1 | 1 | 2 |
| m ississippi \$ | \$ | | | | |
| p i\$mississip | p | | | | |
| p pi\$mississi | i | | | | |
| s ippi\$missis | s | | | | |
| s issippi\$mis | s | | | | |
| s sippi\$missi | i | | | | |
| s 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 |
| i ppi\$mississ | s | 1 | 0 | 1 | 1 |
| i ssippi\$miss | s | 1 | 0 | 1 | 2 |
| i ssissippi\$m | m | 1 | 1 | 1 | 2 |
| m ississippi\$ | \$ | 1 | 1 | 1 | 2 |
| p i\$mississip | p | 1 | 1 | 2 | 2 |
| p pi\$mississi | i | 2 | 1 | 2 | 2 |
| s ippi\$mississ | s | 2 | 1 | 2 | 3 |
| s issippi\$miss | s | 2 | 1 | 2 | 4 |
| s ssippi\$missi | i | 3 | 1 | 2 | 4 |
| s 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!