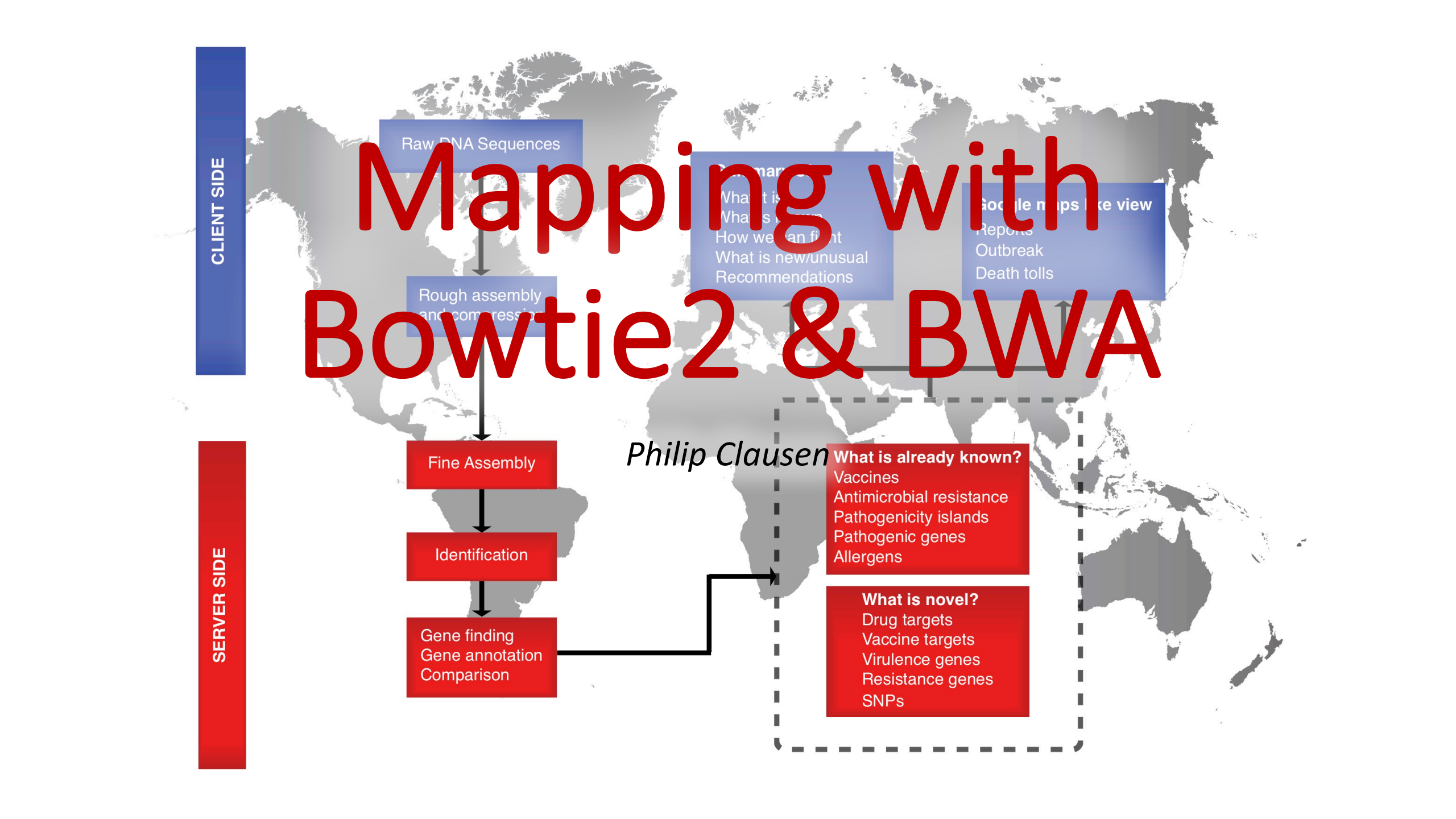
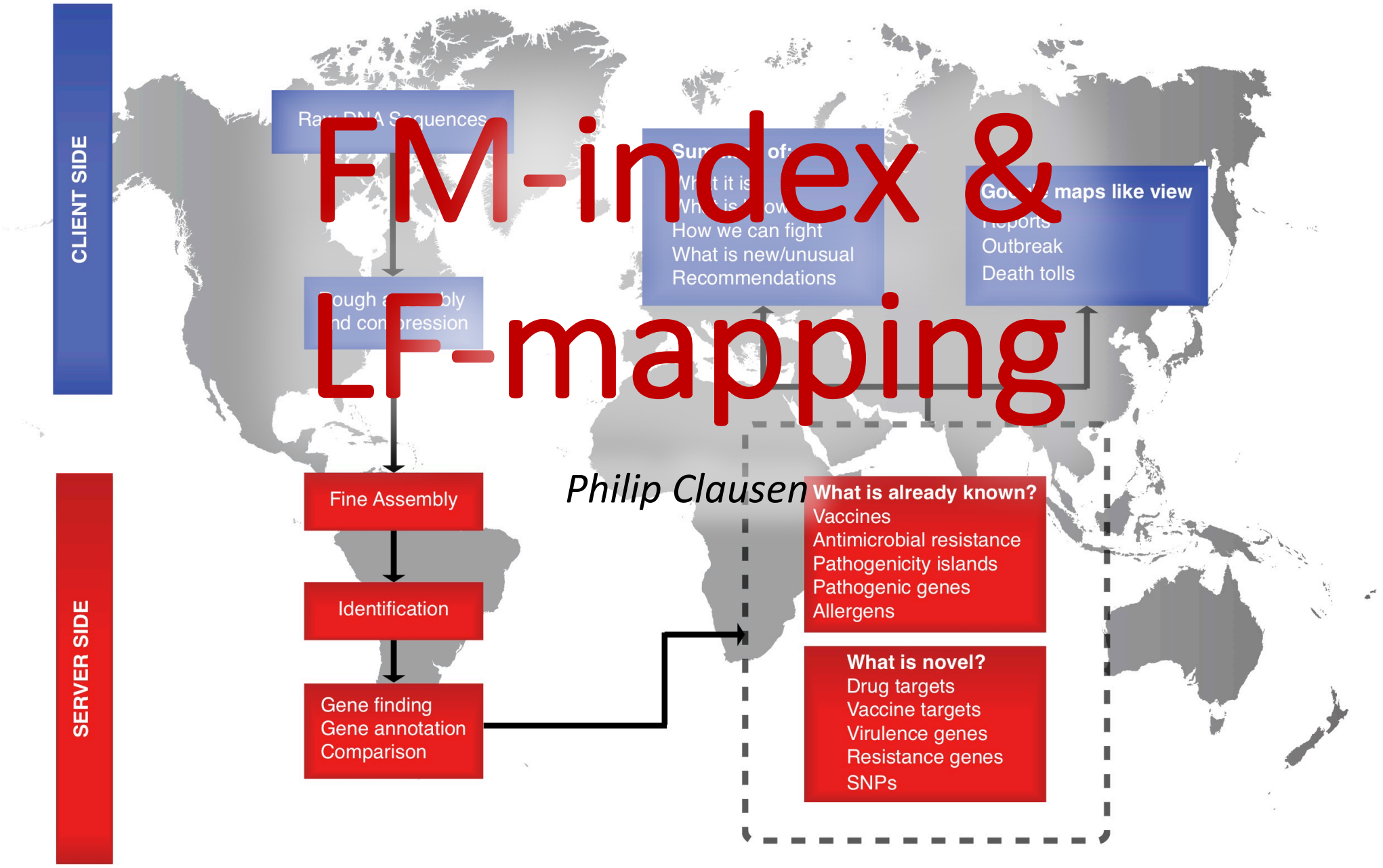


Mapping with Bowtie2 & BWA

Philip Clausen

Last time: Assembly

1. Somewhat complesome and computer heavy to carry out.
2. Very intuitive output.
3. Allows for use of our favorite alignment tool, BLAST.
4. Opens a big box of NGS tools to use, we only touched upon a very small section of these last time.



FM-index & LF-mapping

Philip Clausen

Agenda:

1. **What**
2. **Why**
3. **How**
4. **Bowtie2, BWA & samtools**

1. What

```
@ILLUMINA-3BDE4F_0027:2:1:10636:1724#GCCAAT/1
TGCACAGGTAGCCCCTACGCCGCGNATGAACGACCGGAAACGCCGTCACA
+
a^DG_DPGGDQDFFFFQFKD\\Y\Bab]ac[NY[acaXa]cc_ccYcccc
@ILLUMINA-3BDE4F_0027:2:1:8882:2205#GCCAAT/1
TCCGAATTAAGCGCATATTCTGGTNCACGACTTTGAAGTGTCGCCCTTTT
+
fffffdfffdfcfffdacef_ffcccfcef`WWW^^W^addeccYccbb^
@ILLUMINA-3BDE4F_0027:2:1:14729:12149#GCCAAT/1
TTCCGGCGCATGAGTGTGTATCTTNGCTTTCAACATATTCCCGGGCAATG
+
^^EF[FKFKFZFFFFFQJQF_]__BcaaaccI[`^ca\abccccccccccc
@ILLUMINA-3BDE4F_0027:2:2:15011:1176#GCCAAT/1
TGCGGGATATTATTAAACACATCAAGGCACAGCGCCTTATCTAAACAACTA
+
gggggfbfaGSZK]NKZSRQcccddgfgfgggggggcggggffdgggegg
```

```
ILLUMINA-3BDE4F_0027:2:1:10636:1724#GCCAAT 23 blaIMP-42_1_AB753456 48 1
TGCACAGGTAGCCCCTACGCCGCGNATGAACGACCGGAAACGCCGTCACA
a^DG_DPGGDQDFFFFQFKD\\Y\Bab]ac[NY[acaXa]cc_ccYcccc
AS:i:0 XS:i:0

ILLUMINA-3BDE4F_0027:2:1:8882:2205#GCCAAT 40 sul1_33_AJ564903 48 1
TCCGAATTAAGCGCATATTCTGGTNCACGACTTTGAAGTGTCGCCCTTTT
fffffdfffdfcfffdacef_ffcccfcef`WWW^^W^addeccYccbb^
AS:i:0 XS:i:0

ILLUMINA-3BDE4F_0027:2:2:15011:1176#GCCAAT 57 tet(A)_2_X00006 48 1
TGCGGGATATTATTAAACACATCAAGGCACAGCGCCTTATCTAAACAACTA
NGGGGGNNNAGNNNNNNNNNCCCNNGNGNGGGGGGGGCGGGGNNNGGGNG
G AS:i:0 XS:i:0
```

2. Why

Assembly is computationally heavy, if avoidable we skip it.

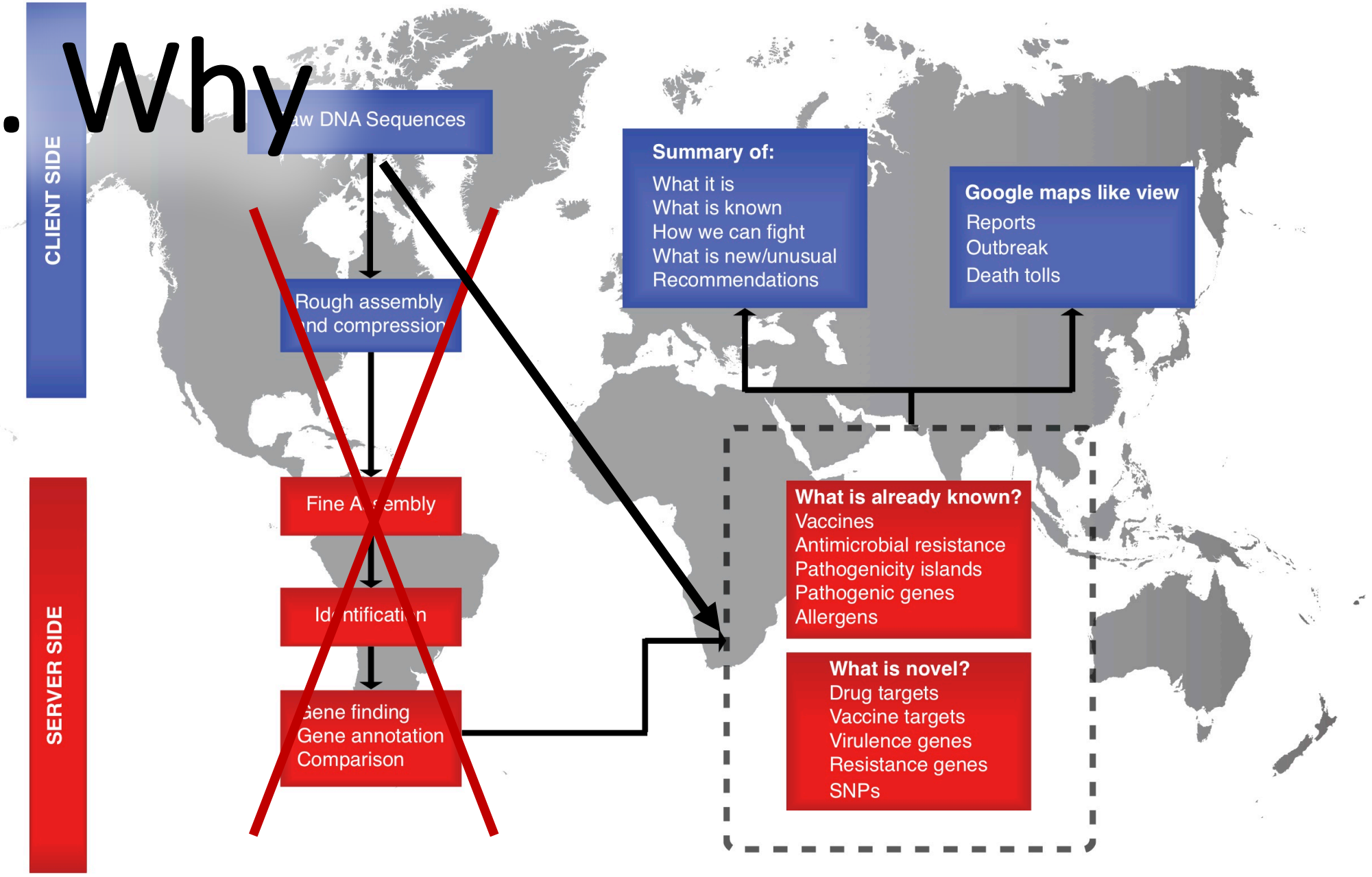
Time complexities:

Assembly: $O(\sum_k n + m_k)$

Mapping: O(n)

I.e. if you know what you a searching for, you might as well start looking for it right away.

2. Why


CLIENT SIDE
SERVER SIDE
aw DNA Sequences
Rough assembly
nd compression
Fine A sembly
Id ntificati n
ene finding
Gene annotation
Comparison
Summary of:
What it is
What is known
How we can fight
What is new/unusual
Recommendations
Google maps like view
Reports
Outbreak
Death tolls
What is already known?
Vaccines
Antimicrobial resistance
Pathogenicity islands
Pathogenic genes
Allergens
What is novel?
Drug targets
Vaccine targets
Virulence genes
Resistance genes
SNPs

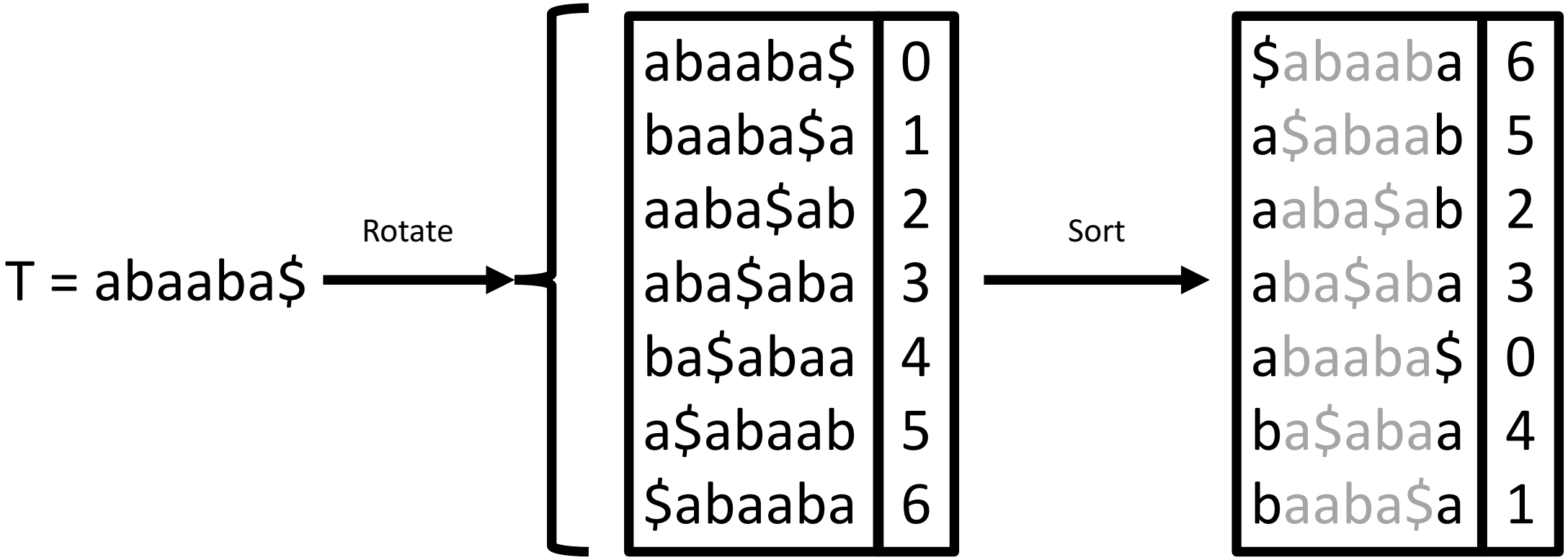
3. How

1. FM-index

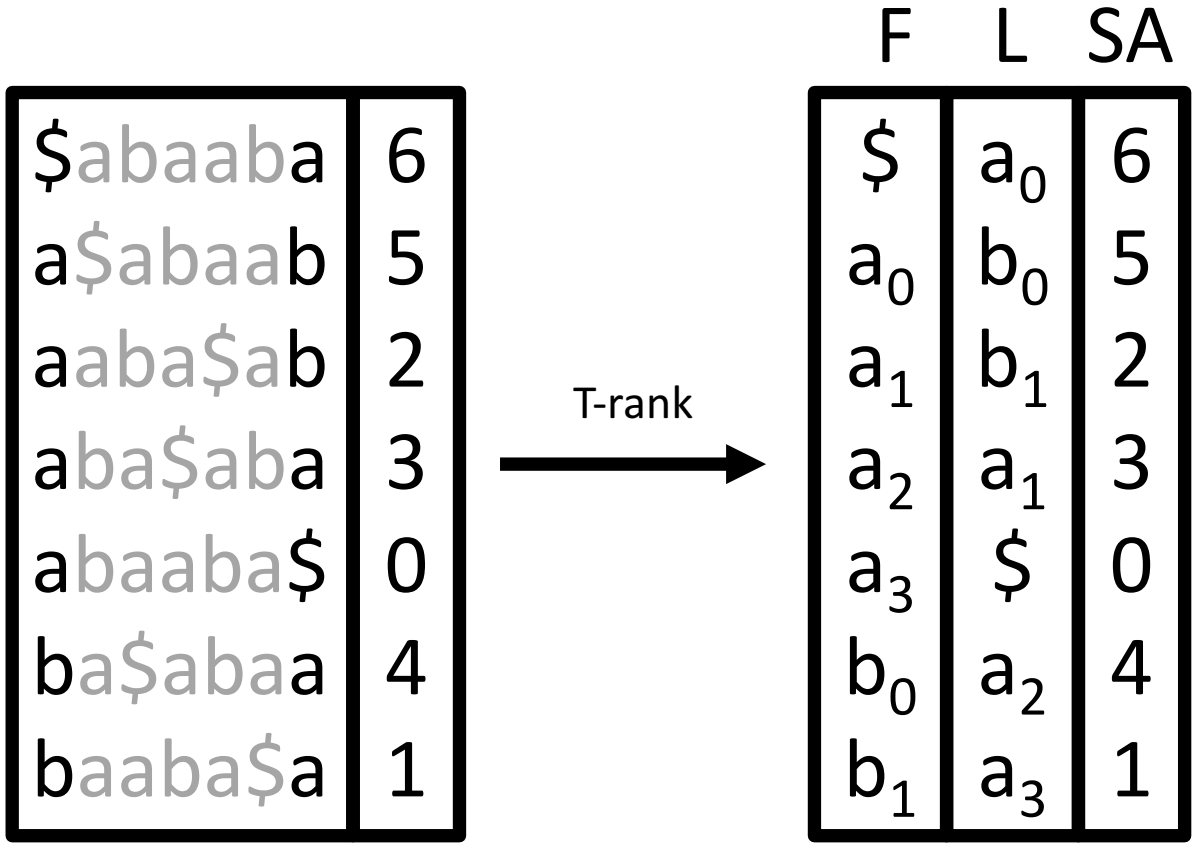
2. LF-mapping

FM-index

Make an indexing of the reference, so that any query can be quickly mutated into the reference.

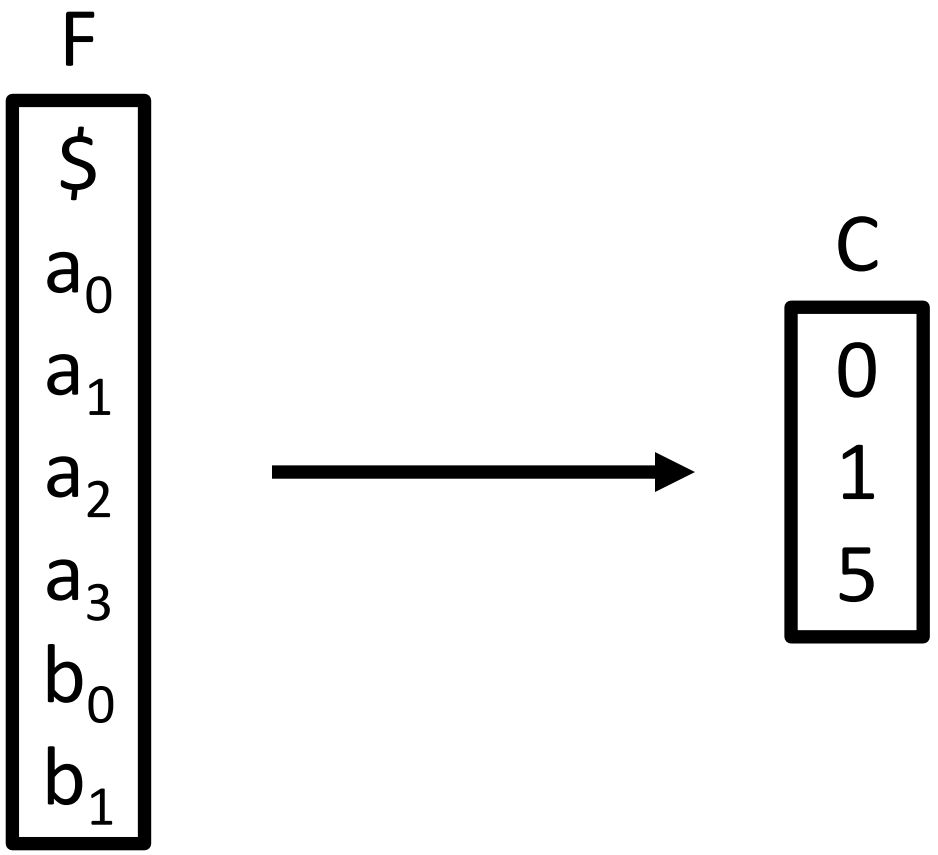


Burrows Wheeler Transform: LF mapping



Bzip compression

Store repeats in one variable.



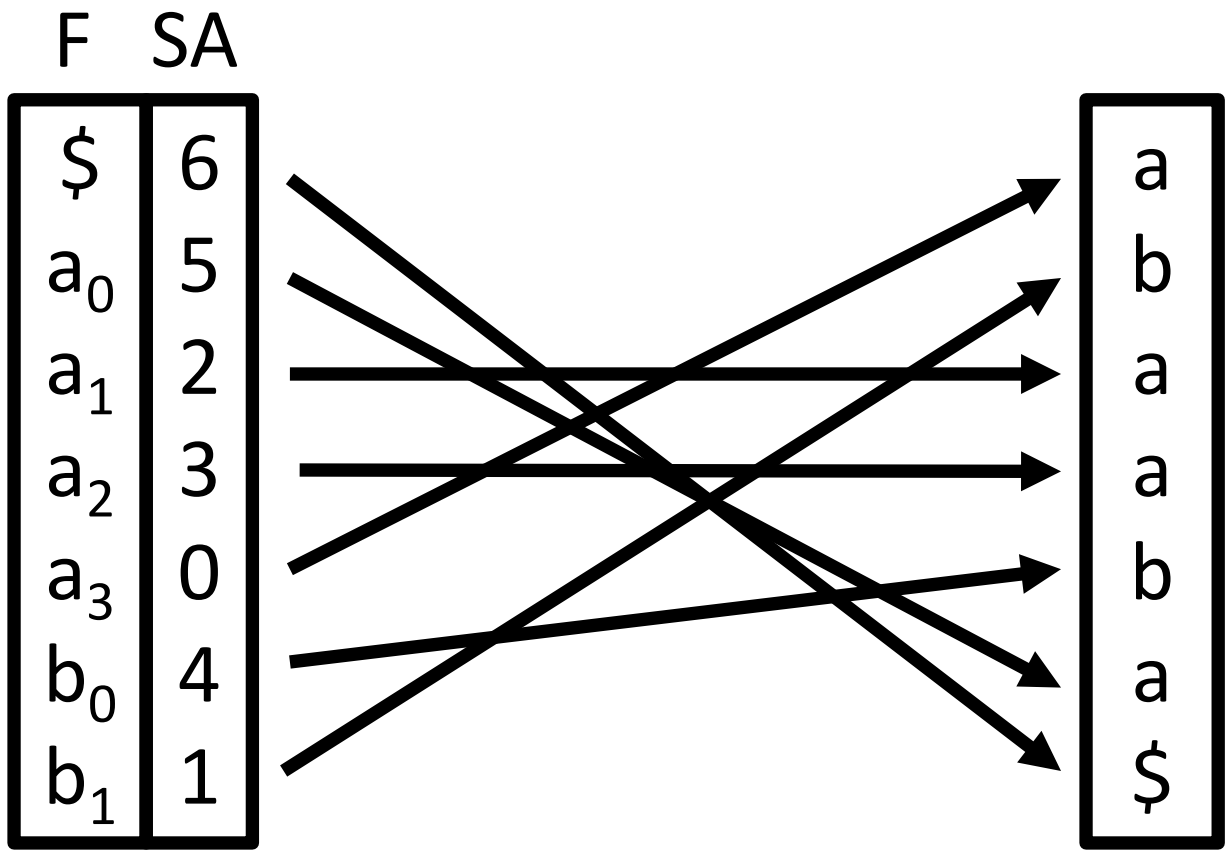
Bzip compression, example:

Indexing has been performed on a DNA sequence with:
120 A's, 200 C's, 130 G'c and 150 T's. With $ < A < C < T < G.
How do we then get G 122.

1. Skip $
2. Skip 120 A's
3. Skip 200 C's
4. Skip 122 G's

Row = 1+120+200+122 = 443

Reconstruction:



Query:

$$C = F\ compressed = \{\$: 0; a: 1; b: 5\}$$

$$S' = C_c + Trank(S, c)$$

$$E' = C_c + Trank(E, c)$$

Query:

Q = bbaba

S = 1

E = 4

F	L	SA
$	a_0	6
a_0	b_0	5
a_1	b_1	2
a_2	a_1	3
a_3	$	0
b_0	a_2	4
b_1	a_3	1

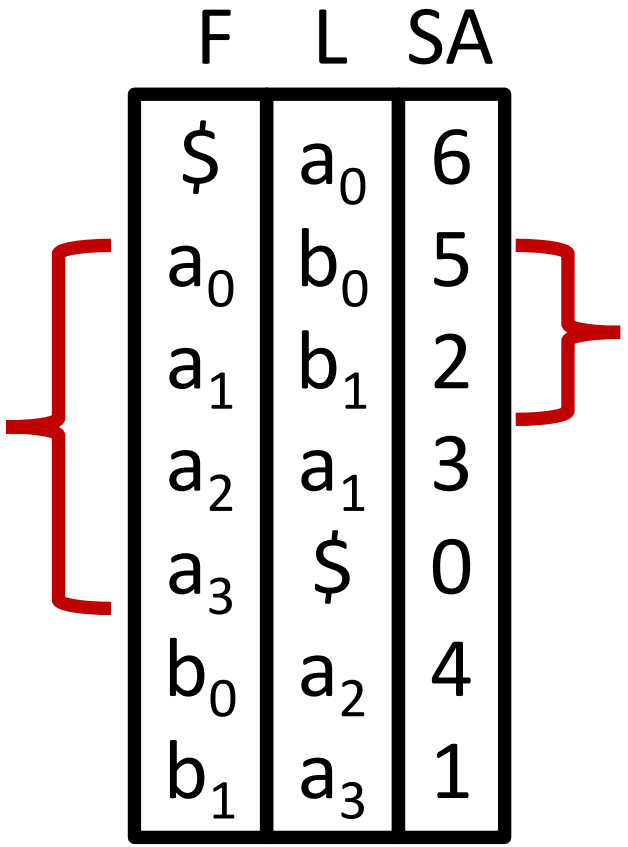
Query:

Q = bbaba

c = a, C = 1

S’ = 5 + 0 = 5

E’ = 5 + 1 = 6



Query:

Q = bbaba

c = b, C = 5

S’ = 1 + 2 = 3

E’ = 1 + 3 = 4

F	L	SA
$	a_0	6
a_0	b_0	5
a_1	b_1	2
a_2	a_1	3
a_3	$	0
b_0	a_2	4
b_1	a_3	1

Possible match positions:
0

Query:

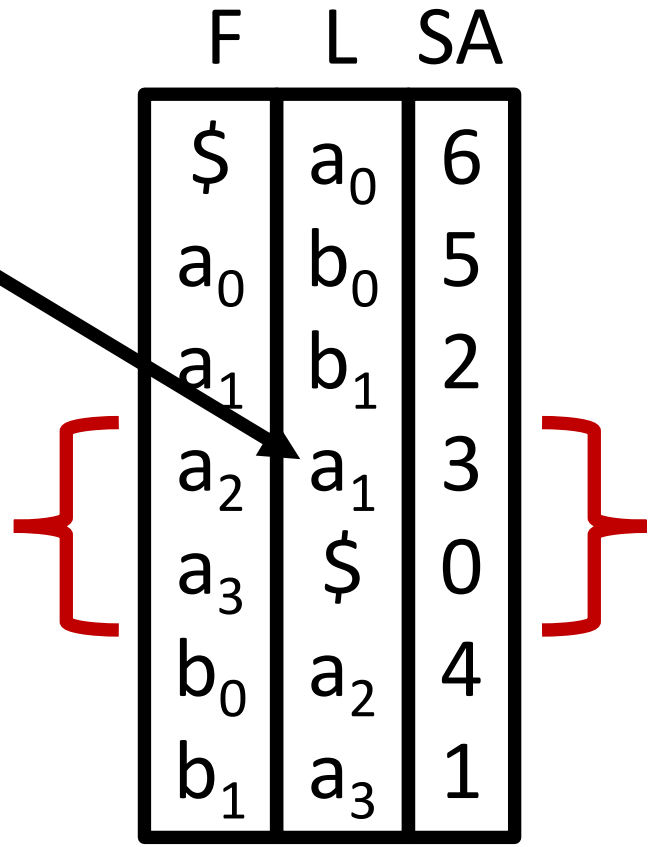
Q = bbaba

c = a, C = 1

S’ = ?

E’ = ?

Ignore mismatch and treat it as an ‘a’.



Possible match positions:
0

Query:

Q = bbaba

c = a, C = 1

S’ = 1 + 1 = 2

E’ = 1 + 1 = 2

Ignore mismatch and treat it as an ‘a’.

F	L	SA
$	a_0	6
a_0	b_0	5
a_1	b_1	2
a_2	a_1	3
a_3	$	0
b_0	a_2	4
b_1	a_3	1

Possible match positions:
0

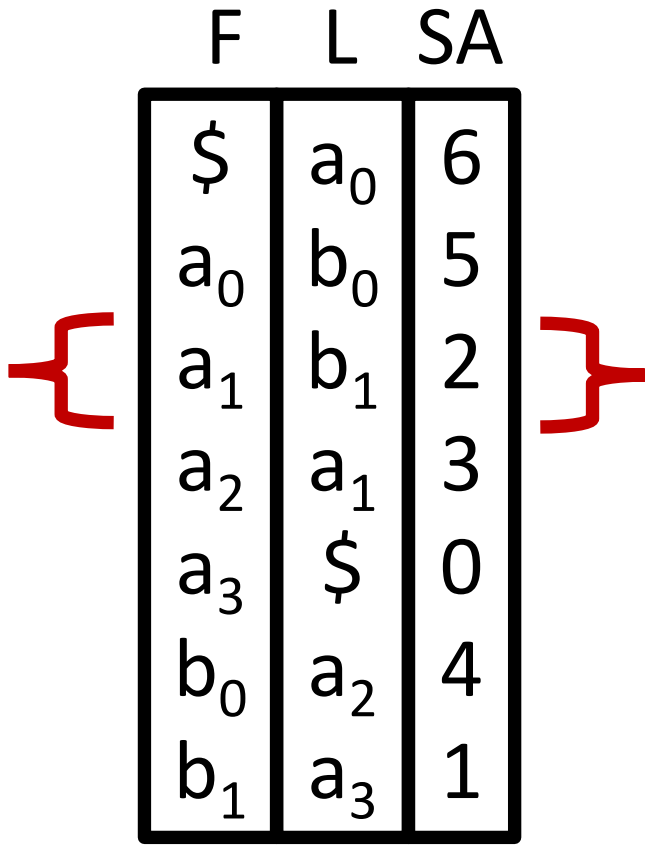
Query:

Q = bbaba

c = a, C = 1

S’ = 5 + 1 = 6

E’ = 5 + 1 = 6



Query:

Q = bbaba

c = a, C = 1

S’ = 5 + 1 = 6

E’ = 5 + 1 = 6

Possible match positions:
0
1

F	L	SA
$	a_0	6
a_0	b_0	5
a_1	b_1	2
a_2	a_1	3
a_3	$	0
b_0	a_2	4
b_1	a_3	1

Our achieved matches:

```
T :          abaaba
SA:          012345

M0:        bbaba
M1:            bbaba
```

The best match depends on various scoring penalties for:

1. Match
2. Mismatch
3. Gap opening
4. Gap extension
5. Not able to match end to end

Wait a minute !?

What about lookup time, O(1) ?

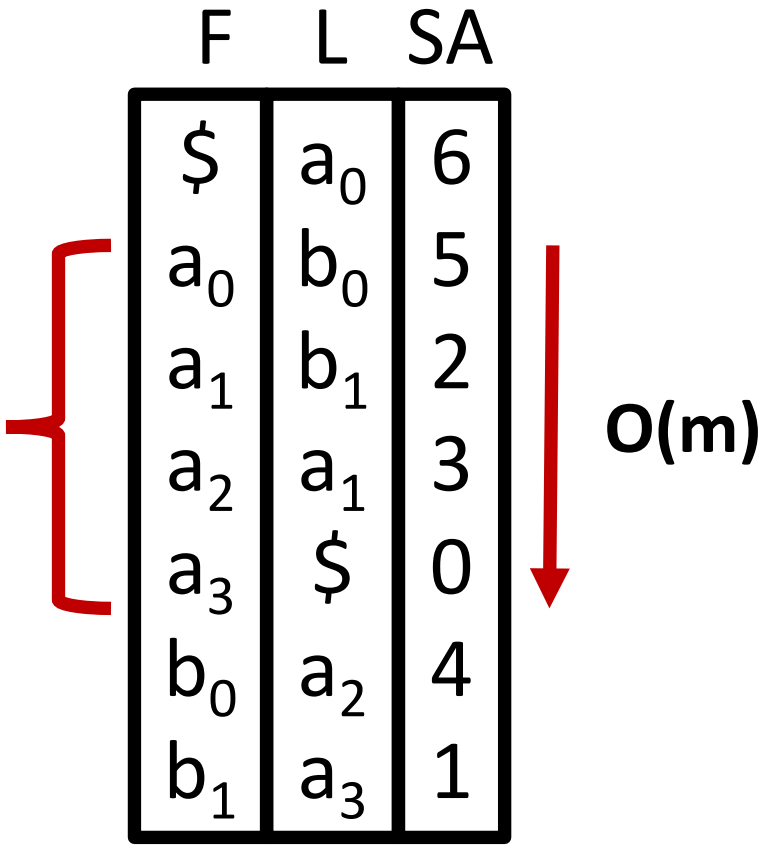
Query:

Q = bbaba

c = a, C = 1

S’ = 5 + 0 = 5

E’ = 5 + 1 = 6



What do we do?

Examples:

1. Save the position of each character. (Memory heavy)

2. Save some positions of each character. (Back to time)

3. Make constant spaces between positions, O(1) lookup.

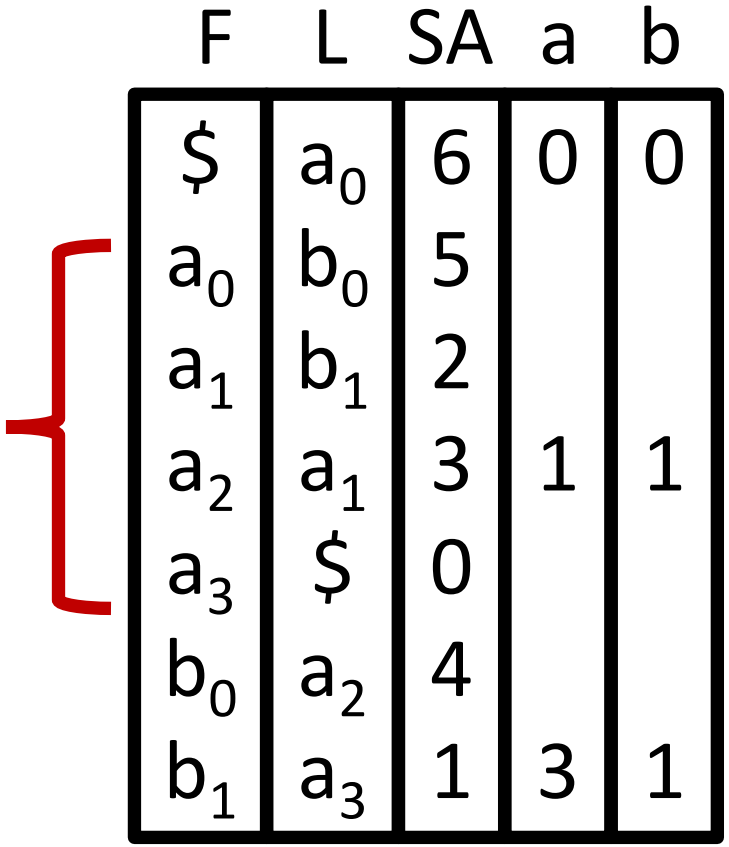
Query:

Q = bbaba

c = a, C = 1

S’ = 5 + 0 = 5

E’ = 5 + 1 = 6



F	L	SA	a	b
$	a_0	6	0	0
a_0	b_0	5		
a_1	b_1	2		
a_2	a_1	3	1	1
a_3	$	0		
b_0	a_2	4		
b_1	a_3	1	3	1

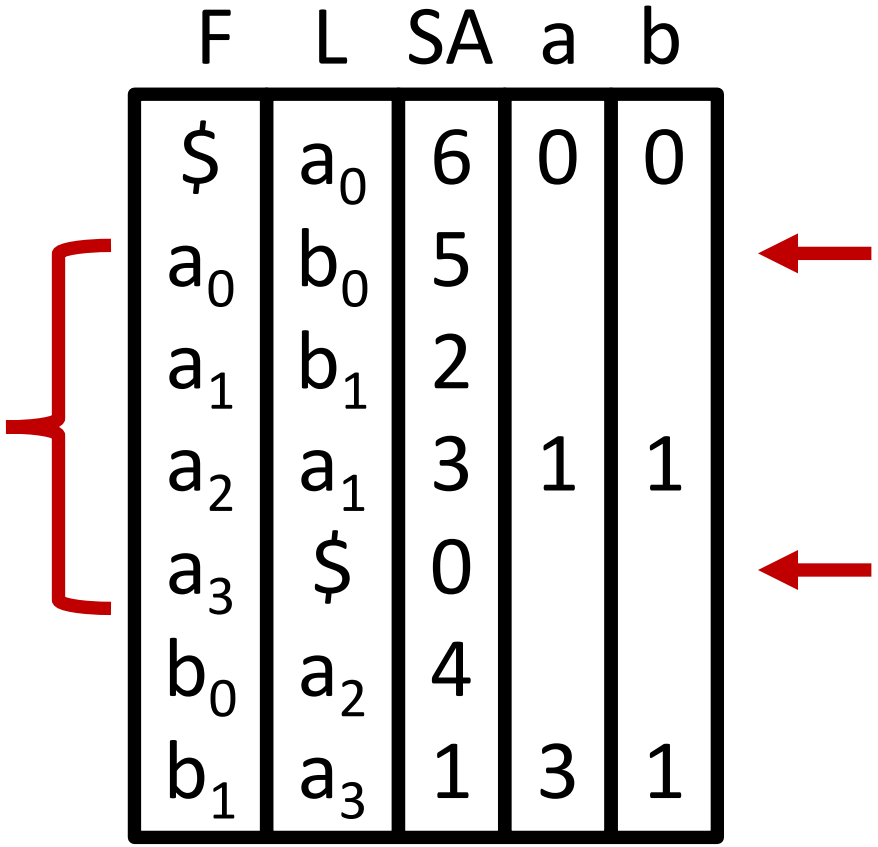
Query:

Q = bbaba

c = a, C = 1

S’ = 5 + 0 = 5

E’ = 5 + 1 = 6



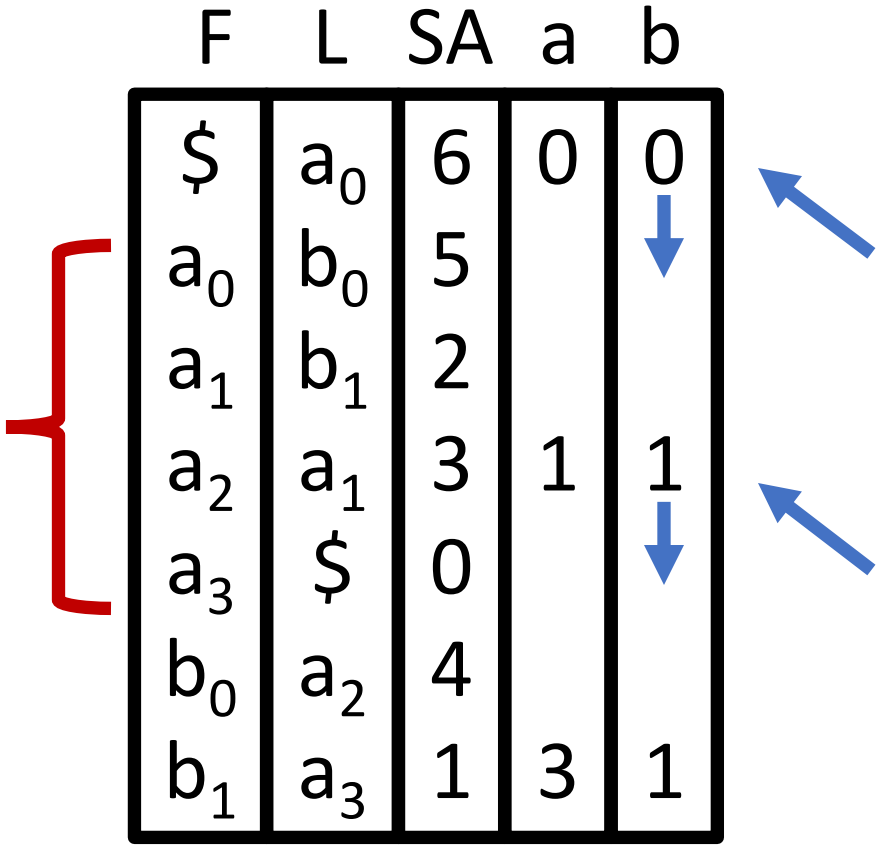
Query:

Q = bbaba

c = a, C = 1

S’ = 5 + 0 = 5

E’ = 5 + 1 = 6



Compression of genomic data:

Alphabet composed of 4 characters,
each character can be encoded with 2 bit’s.

F	->	4 integers.
L	->	encoded in 2-bit arrays.
SA	->	Save only a fraction.
Positions	->	Save only a fraction.

Compression example

8 K complete bacterial genomes ~ 30 billion nucleotides.

F	->	4 * 4 bytes = 16 bytes
L	->	30 billion * 2 bits = 7.5 GB
SA	->	30 billion * 4 bytes / 32 = 3.75 GB
Positions	->	30 billion * 4 * 4 bytes / 32 = 15 GB
Total	->	26.25 GB

Implementations and assumptions

1. Bowtie (not used today)
2. Bowtie2
3. BWA

1. Bowtie

1. Map against large reference genome.
2. Short high quality reads.
3. Reference and query must be highly similar.
4. No gaps.
5. No non-uniformly distributed redundancy.

2. Bowtie

1. Map against large reference genome.
2. Short to semi-long high quality reads.
3. Reference and query must be highly similar.
4. No non-uniformly distributed redundancy.

3. BWA

1. Map against large reference genome.
2. Short to semi-long high quality reads.
3. Reference and query must be highly similar.
4. No non-uniformly distributed redundancy.

Summary:

1. Memory efficient.
2. Relatively fast alignment, O(n).
3. Outputs in sam-format, which is standardized for human mapping.
4. All three works well, **if the assumptions are met!!!**

General pitfalls:

1. Not designed for database mapping.
2. Only designed to find highly similar templates.
3. Computation time rises exponentially with the number of gaps (to a user specified maximum).
4. Uses a minimum seed length of 19 (user specified).
5. Poor handle of redundancy.

$	S	9
E	U	2
I	T	5
N	O	7
O	I	6
S	N	8
S	E	3
T	S	4
U	Q	1
Q	$	0

$QUESTIONS	9
ESTIONS$QU	2
IONS$QUEST	5
NS$QUESTIO	7
ONS$QUESTI	6
S$QUESTION	8
STIONS$QUE	3
TIONS$QUES	4
UESTIONS$Q	1
QUESTIONS$	0