



Posgrado en
**Biología
Integrativa**



Transcriptómica

Clase 4 - Alineamiento y cuantificación

Biología Computacional 2017

Selene L. Fernández-Valverde

[regRNAlab.github.io](https://github.com/regRNAlab/regRNAlab)

@Selfdz

Objetivos de aprendizaje

En esta clase aprenderemos:

- A a alinear datos de RNA-Seq a una referencia
 - Lecturas crudas
 - Transcritos generados *de novo*
- Entender los formatos SAM y BAM.

Ensamblando transcriptomas

- Tophat

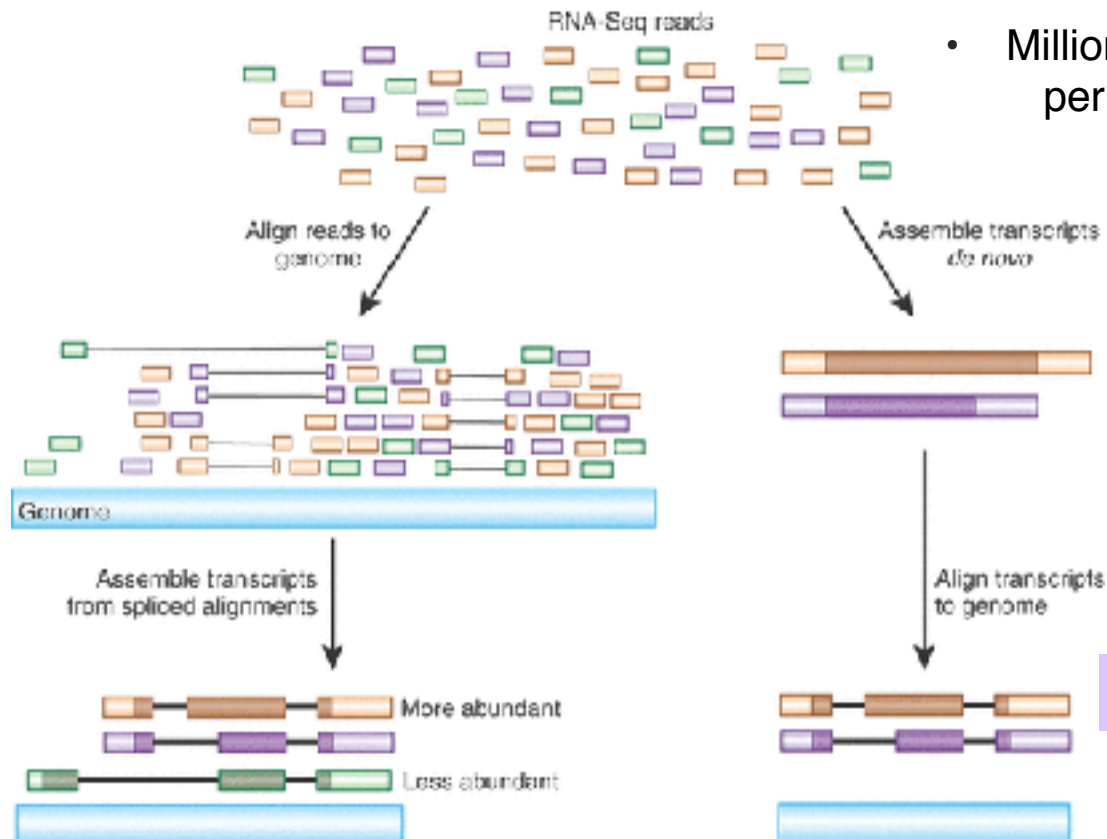
- Bowtie

- BWA

- STAR

- Cufflinks

- Scripture



- Millions of reads per sample !

- Trinity

- Oases de Novo

- trans-ABYSS

- GMAP

- Blat

- Exonerate

Ensamblando transcriptomas

- Tophat

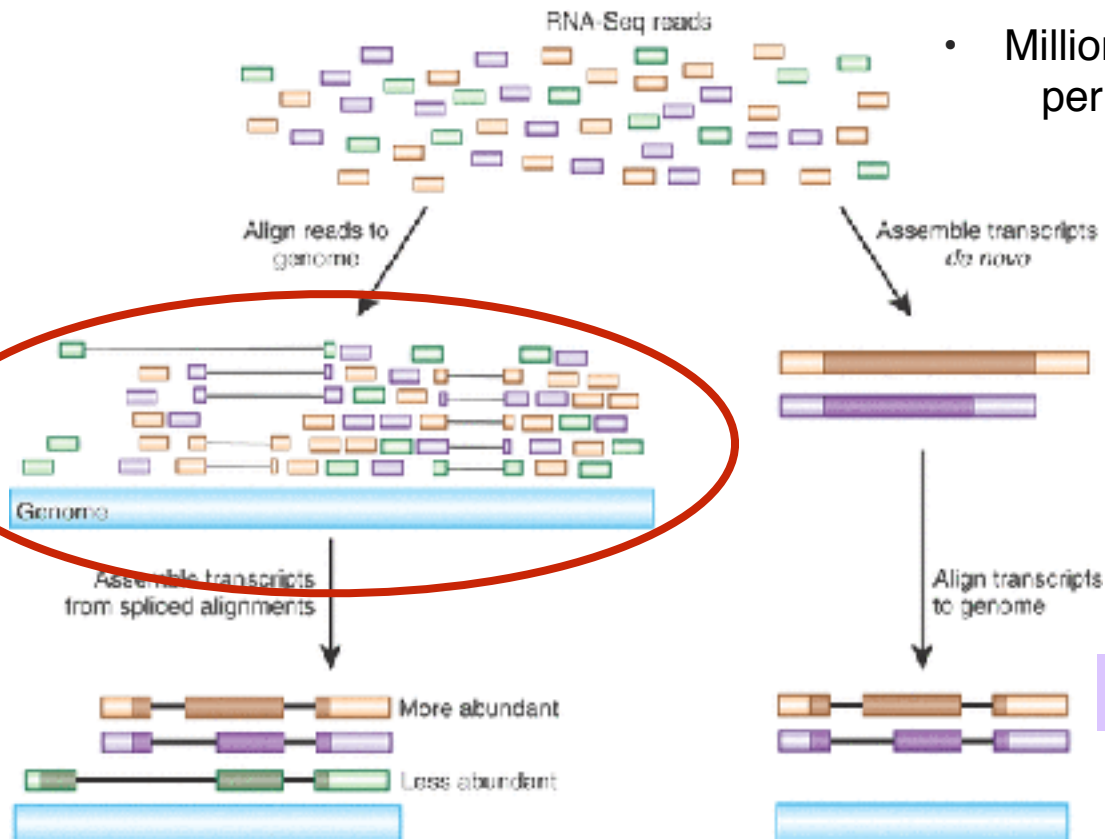
- Bowtie

- BWA

- STAR

- Cufflinks

- Scripture



- Millions of reads per sample !

- Trinity

- Oases de Novo

- trans-ABYSS

- GMAP

- Blat

- Exonerate

Ensamblando transcriptomas

- Tophat

- Bowtie

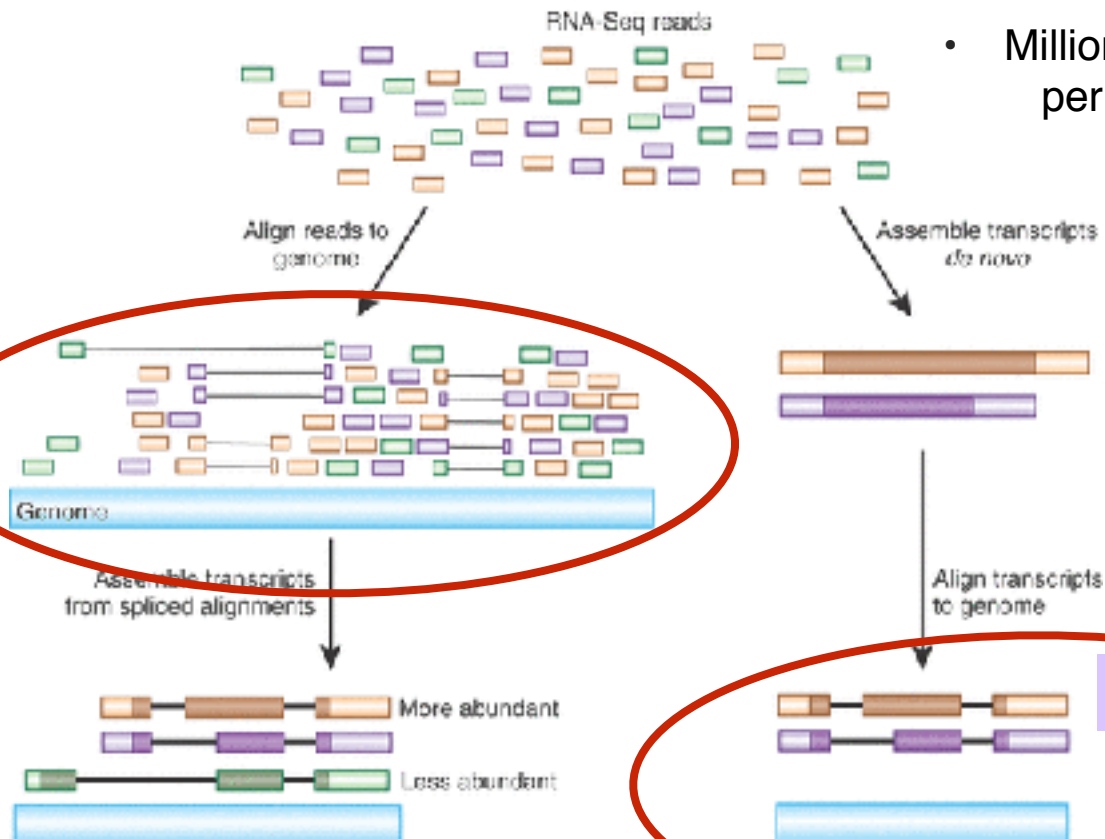
- BWA

- STAR

- Cufflinks

- Scripture

- Millions of reads per sample !



- Trinity

- Oases de Novo

- trans-ABYSS

- GMAP

- Blat

- Exonerate

¿Qué significa alinear (mapear) una secuencia?

- Es identificar la posición de origen (alta similitud) de **lecturas** o transcritos secuenciados en una **secuencia de referencia** (genomas o transcritos)

Secuencia de referencia

GATCACGAAAGCACTTTACTGGGTAAATAAAGTAC

|||||

CGAAAGCACTTTATTGG

Lectura



Error o Mismatch

No podemos usar BLAST

- BLAST hace un alineamiento local, lo cual lo hace muy útil para buscar alineamientos parciales y/o divergentes en bases de datos grandes.
- BLAST es muy lento para alinear secuencias, lo que lo hace poco práctico alinear millones de secuencias.
- Dado que generalmente esperamos un alto nivel de similitud con la referencia en un experimento de secuenciación masiva necesitamos un algoritmo de alineamiento semi-global y muy rápido.

Burrows-Wheeler transform (BWT)

- Descubierta por David Wheeler en 1983.
- Permutación reversible de los caracteres en una cadena - usada originalmente para comprimir datos.
- En 2005 se encontró que era extremadamente útil para encontrar subcadenas.
- En 2009 se comenzó a usar para alinear lecturas resultado de experimentos de secuenciación masiva.
- En conjunto con índices comprimidos (e.g. FM index) permite que el tiempo de alineamiento crece de manera lineal con la cantidad de secuencias.
- Permite alinear ~100 millones de lecturas por hora (Bowtie - 1 solo thread)

Generando una BWT

ATCTTATC\$

\$ - Caracter que indica el final de una cadena

Generando una BWT

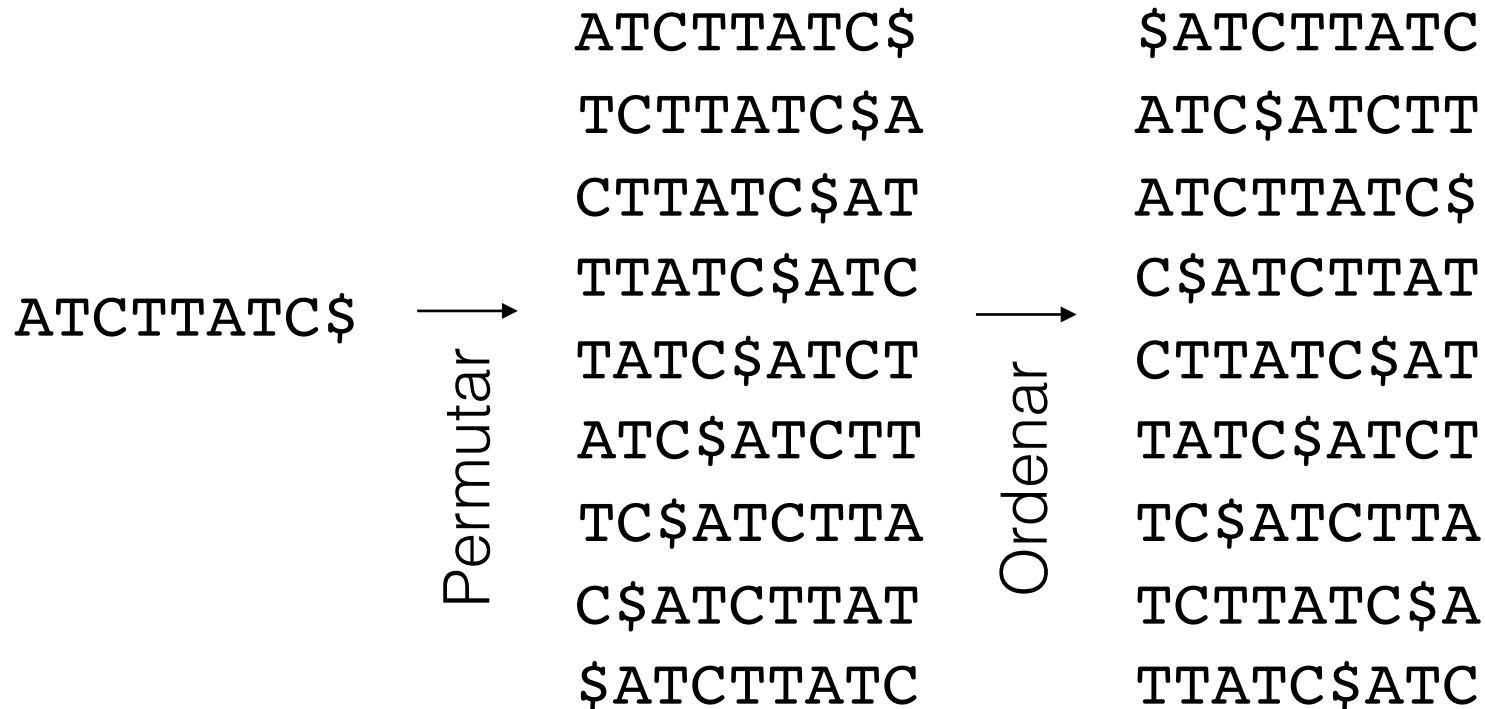
ATCTTATC\$

↓
Permutar

ATCTTATC\$
TCTTATC\$A
CTTATC\$AT
TTATC\$ATC
TATC\$ATCT
ATC\$ATCTT
TC\$ATCTTA
C\$ATCTTAT
\$ATCTTATC

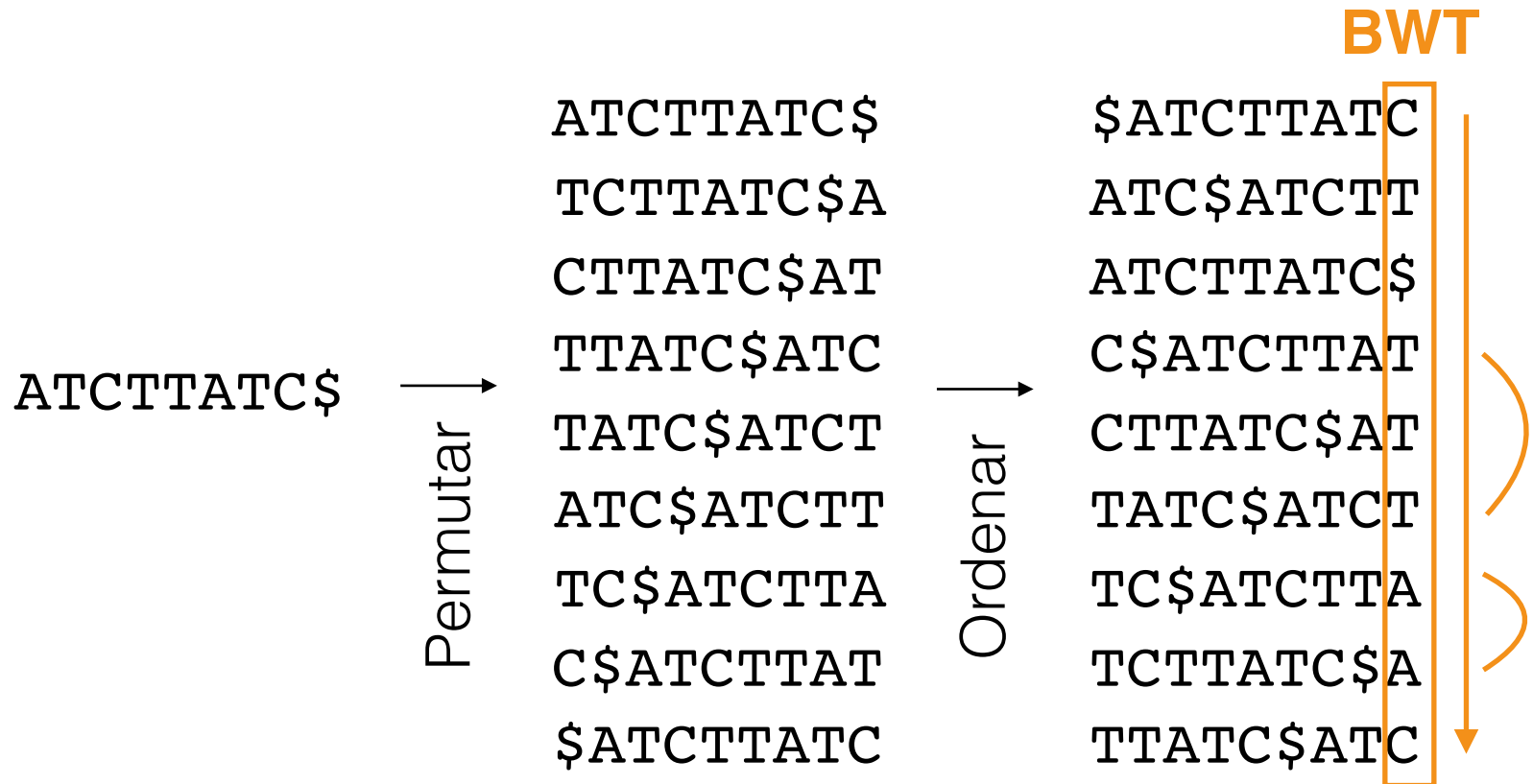
\$ - Caracter que indica el final de una cadena

Generando una BWT



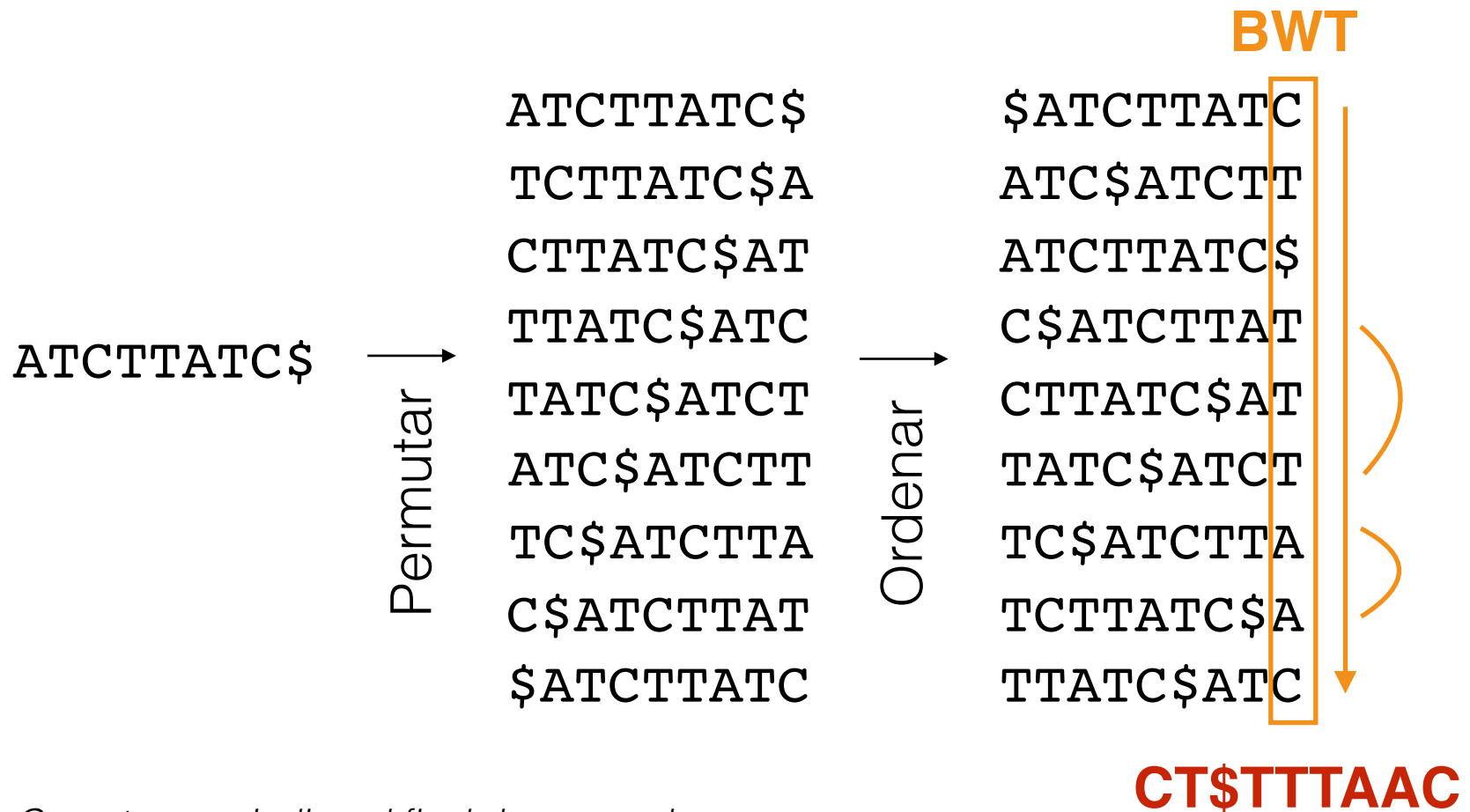
\$ - Caracter que indica el final de una cadena

Generando una BWT



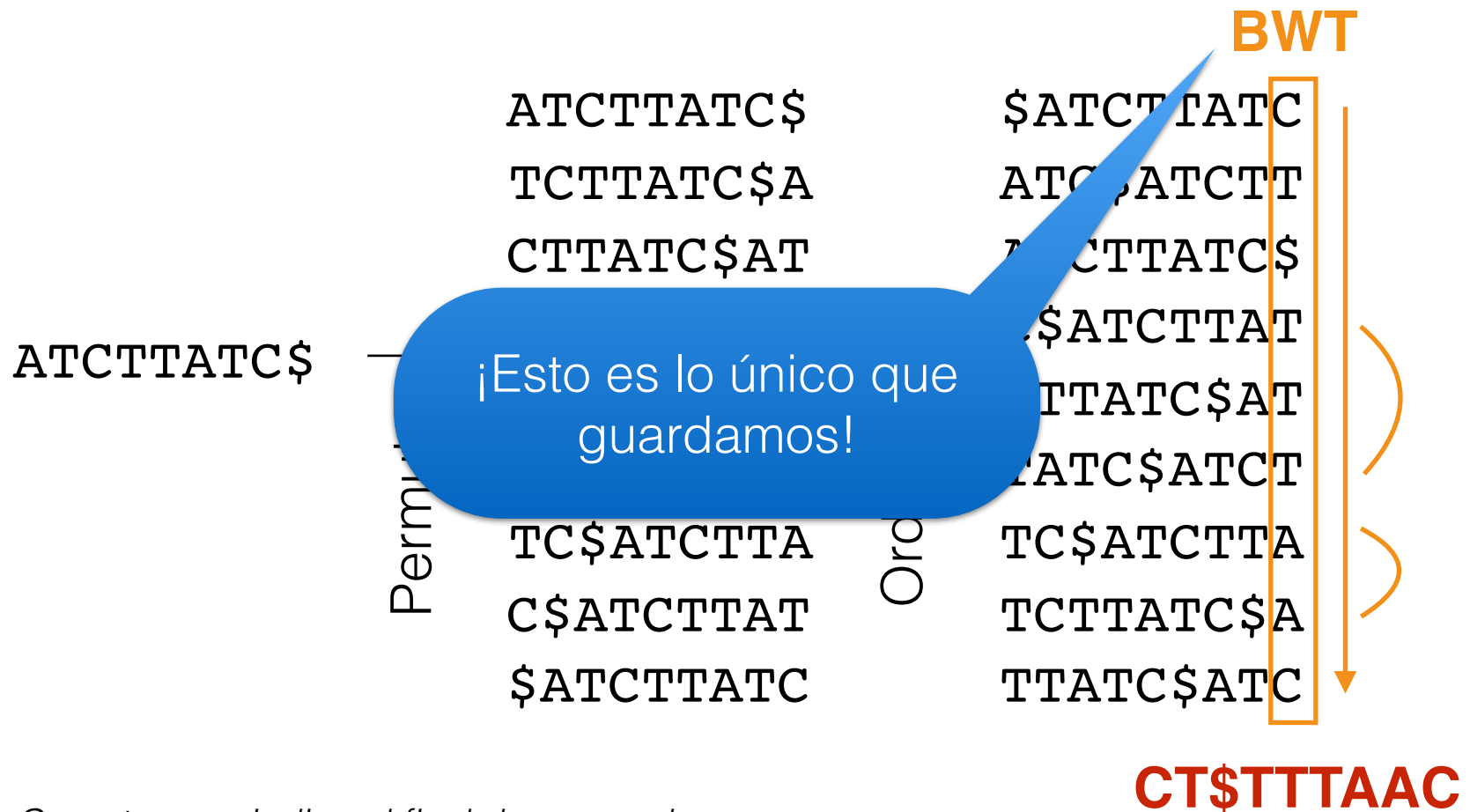
\$ - Caracter que indica el final de una cadena

Generando una BWT



\$ - Caracter que indica el final de una cadena

Generando una BWT



\$ - Caracter que indica el final de una cadena

Propiedad FT

Renglón

0	$\$0$	ATCTTAT	C_0
1	A_0	TC\$ATCT	T_0
2	A_1	TCTTATC	$\$0$
3	C_0	\$ATCTTA	T_1
4	C_1	TTATC\$A	T_2
5	T_0	ATC\$ATC	T_3
6	T_1	C\$ATCTT	A_0
7	T_2	CTTATC\$	A_1
8	T_3	TATC\$AT	C_1

Propiedad FT

Renglón

BWT

0	\$ ₀	ATCTTAT	C ₀
1	A ₀	TC\$ATCT	T ₀
2	A ₁	TCTTATC	\$ ₀
3	C ₀	\$ATCTTA	T ₁
4	C ₁	TTATC\$A	T ₂
5	T ₀	ATC\$ATC	T ₃
6	T ₁	C\$ATCTT	A ₀
7	T ₂	CTTATC\$	A ₁
8	T ₃	TATC\$AT	C ₁

F- First

L- Last

Propiedad FT

Renglón

BWT

0	\$ ₀	ATCTTAT	C ₀
1	A ₀	TC\$ATCT	T ₀
2	A ₁	TCTTATC	\$ ₀
3	C ₀	\$ATCTTA	T ₁
4	C ₁	TTATC\$A	T ₂
5	T ₀	ATC\$ATC	T ₃
6	T ₁	C\$ATCTT	A ₀
7	T ₂	CTTATC\$	A ₁
8	T ₃	TATC\$AT	C ₁

F- First

L- Last

El rango de los caracteres se mantiene en la primera (F) y última (L) columna.

La primera columna se puede reconstruir ordenando la última

Revirtiendo la transformación BWT

Renglón

0	S_0	C_0
1	A_0	T_0
2	A_1	S_0
3	C_0	T_1
4	C_1	T_2
5	T_0	T_3
6	T_1	A_0
7	T_2	A_1
8	T_3	C_1

Secuencia original

Revirtiendo la transformación BWT

Renglón

0	\$ ₀	C ₀
1	A ₀	T ₀
2	A ₁	\$ ₀
3	C ₀	T ₁
4	C ₁	T ₂
5	T ₀	T ₃
6	T ₁	A ₀
7	T ₂	A ₁
8	T ₃	C ₁

Secuencia original

Revirtiendo la transformación BWT

Renglón

0	\$ ₀	C ₀
1	A ₀	T ₀
2	A ₁	\$ ₀
3	C ₀	T ₁
4	C ₁	T ₂
5	T ₀	T ₃
6	T ₁	A ₀
7	T ₂	A ₁
8	T ₃	C ₁

\$₀
Secuencia original

Revirtiendo la transformación BWT

Renglón

0	\$ ₀	→	C ₀
1	A ₀		T ₀
2	A ₁		\$ ₀
3	C ₀		T ₁
4	C ₁		T ₂
5	T ₀		T ₃
6	T ₁		A ₀
7	T ₂		A ₁
8	T ₃		C ₁

C₀ \$₀

Secuencia original

Revirtiendo la transformación BWT

Renglón

0	\$ ₀	→	C ₀
1	A ₀		T ₀
2	A ₁		\$ ₀
3	C ₀	↙	T ₁
4	C ₁		T ₂
5	T ₀		T ₃
6	T ₁		A ₀
7	T ₂		A ₁
8	T ₃		C ₁

C₀ \$₀

Secuencia original

Revirtiendo la transformación BWT

Renglón

0	\$ ₀	→	C ₀
1	A ₀		T ₀
2	A ₁		\$ ₀
3	C ₀	→	T ₁
4	C ₁		T ₂
5	T ₀		T ₃
6	T ₁		A ₀
7	T ₂		A ₁
8	T ₃		C ₁

T₁ C₀ \$₀

Secuencia original

Revirtiendo la transformación BWT

Renglón

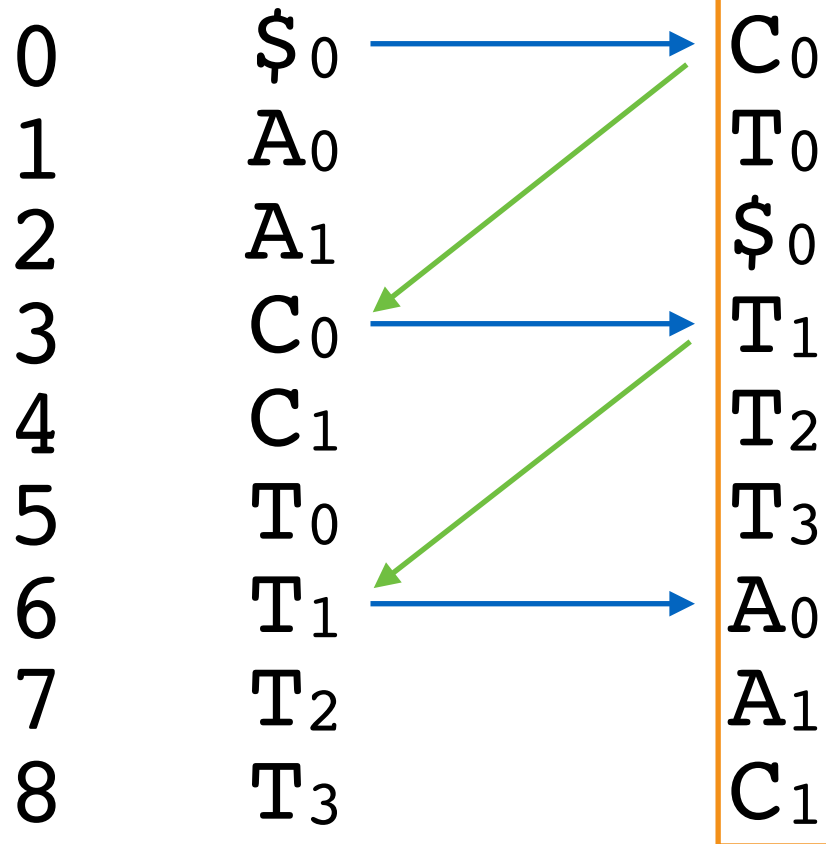
0	\$ ₀	→	C ₀
1	A ₀	↘	T ₀
2	A ₁	↘	\$ ₀
3	C ₀	→	T ₁
4	C ₁	↘	T ₂
5	T ₀	↘	T ₃
6	T ₁	↘	A ₀
7	T ₂	↘	A ₁
8	T ₃	↘	C ₁

T₁ C₀ \$₀

Secuencia original

Revirtiendo la transformación BWT

Renglón

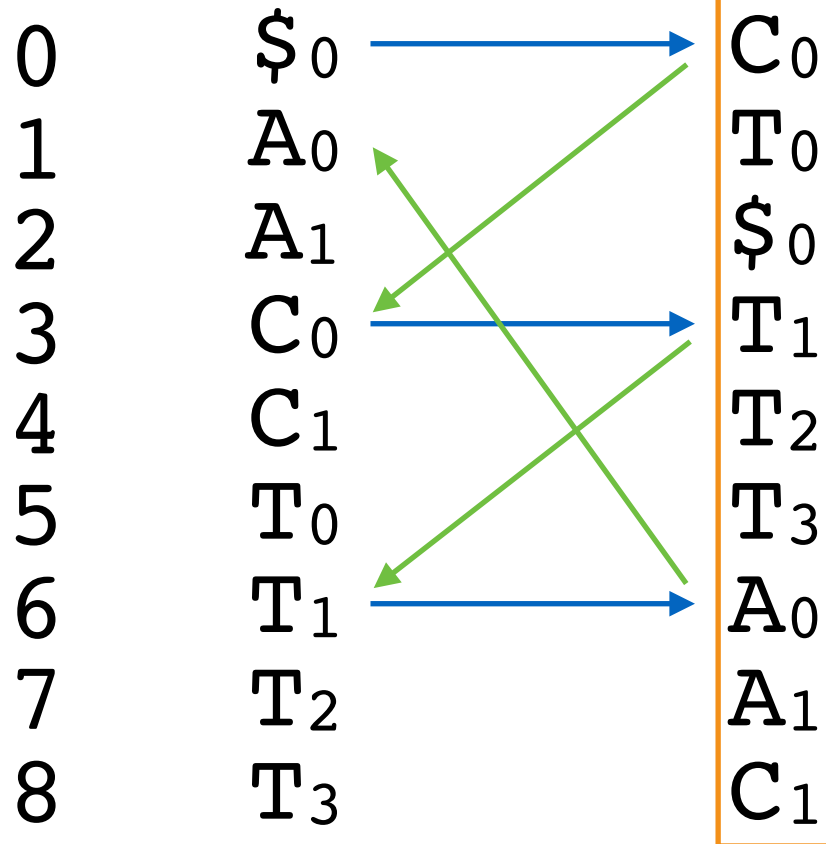


A₀ T₁ C₀ \$₀

Secuencia original

Revirtiendo la transformación BWT

Renglón

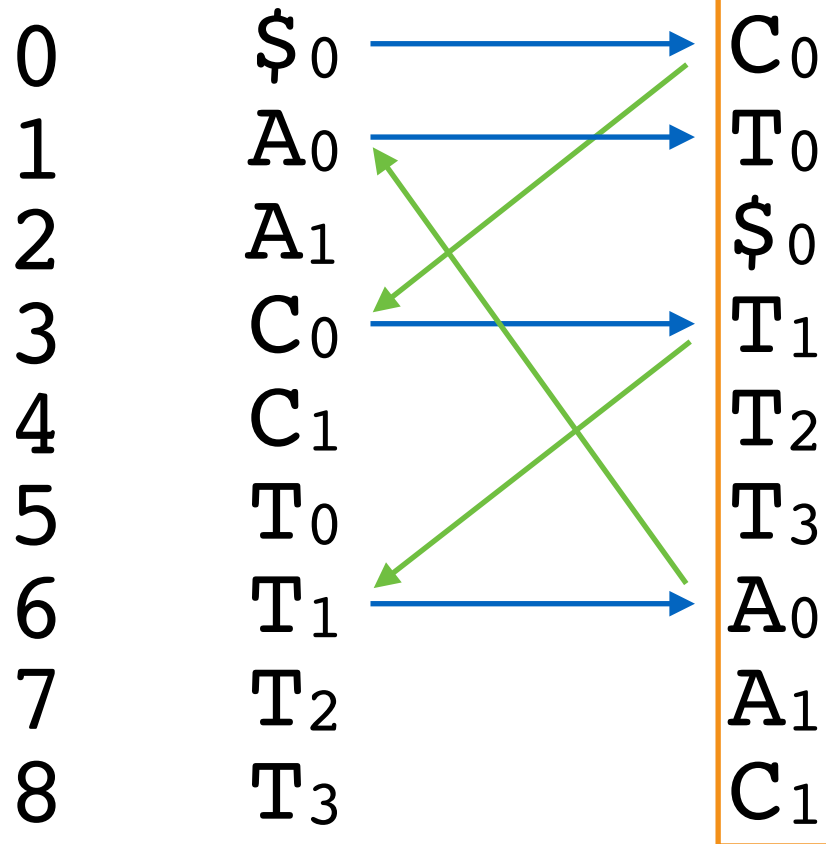


A₀ T₁ C₀ \$₀

Secuencia original

Revirtiendo la transformación BWT

Renglón

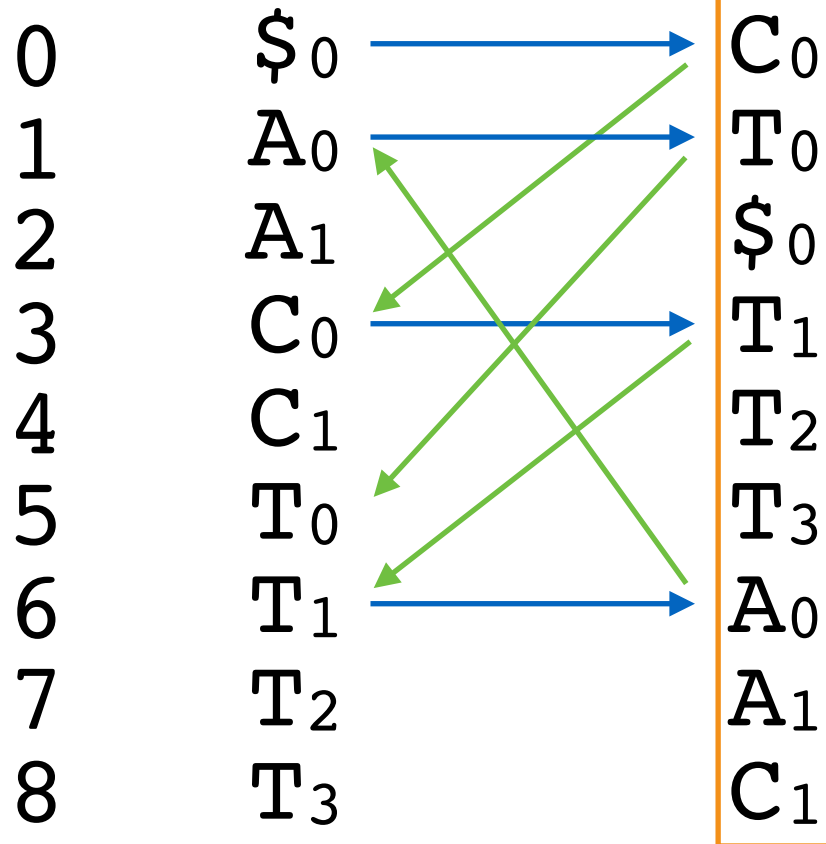


T₀ A₀ T₁ C₀ \$₀

Secuencia original

Revirtiendo la transformación BWT

Renglón

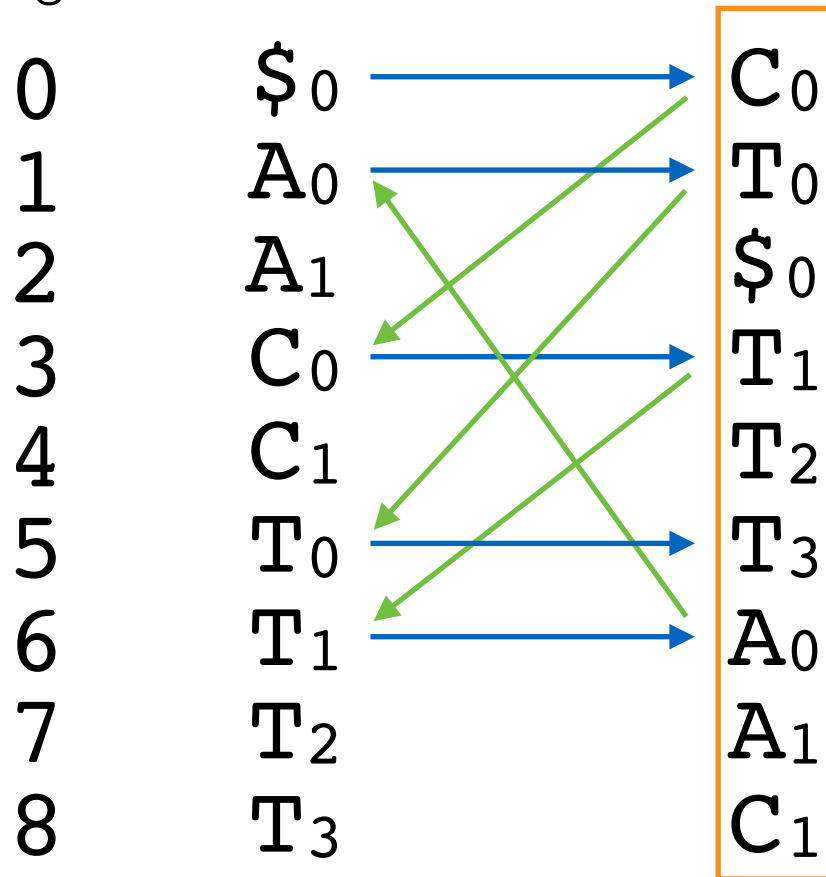


T₀ A₀ T₁ C₀ \$₀

Secuencia original

Revirtiendo la transformación BWT

Renglón

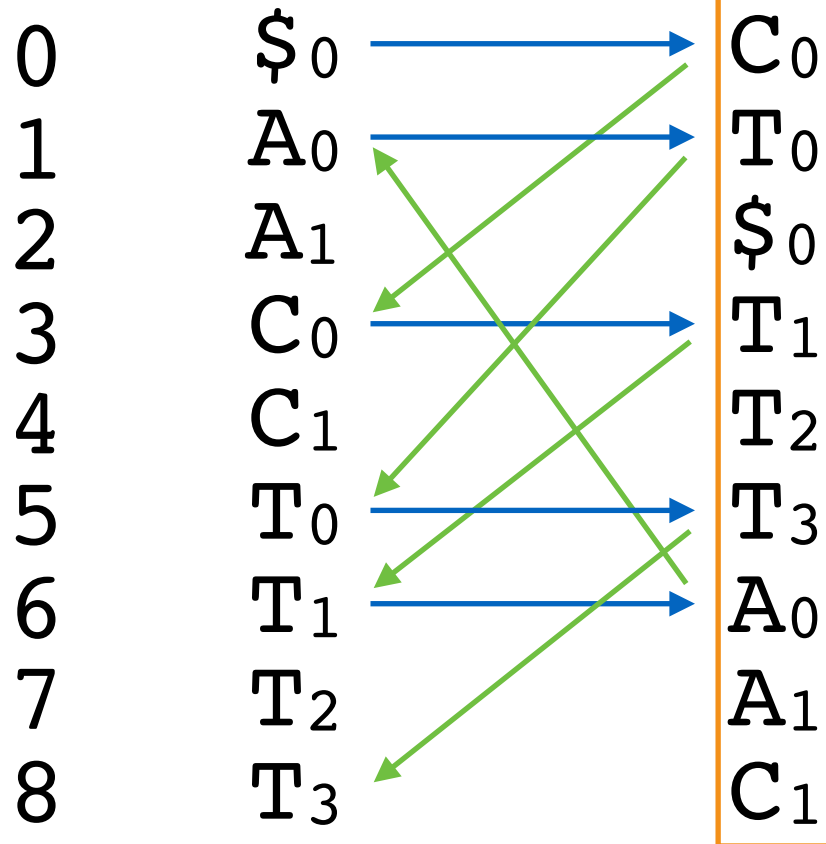


T₃ T₀ A₀ T₁ C₀ \$₀

Secuencia original

Revirtiendo la transformación BWT

Renglón

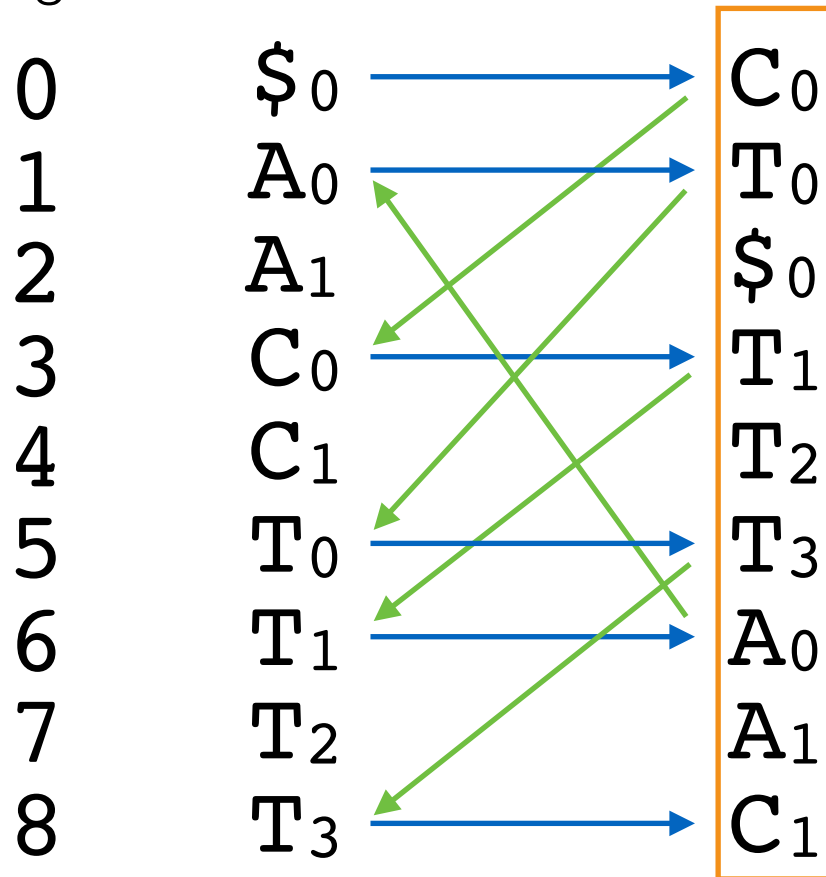


T₃ T₀ A₀ T₁ C₀ \$₀

Secuencia original

Revirtiendo la transformación BWT

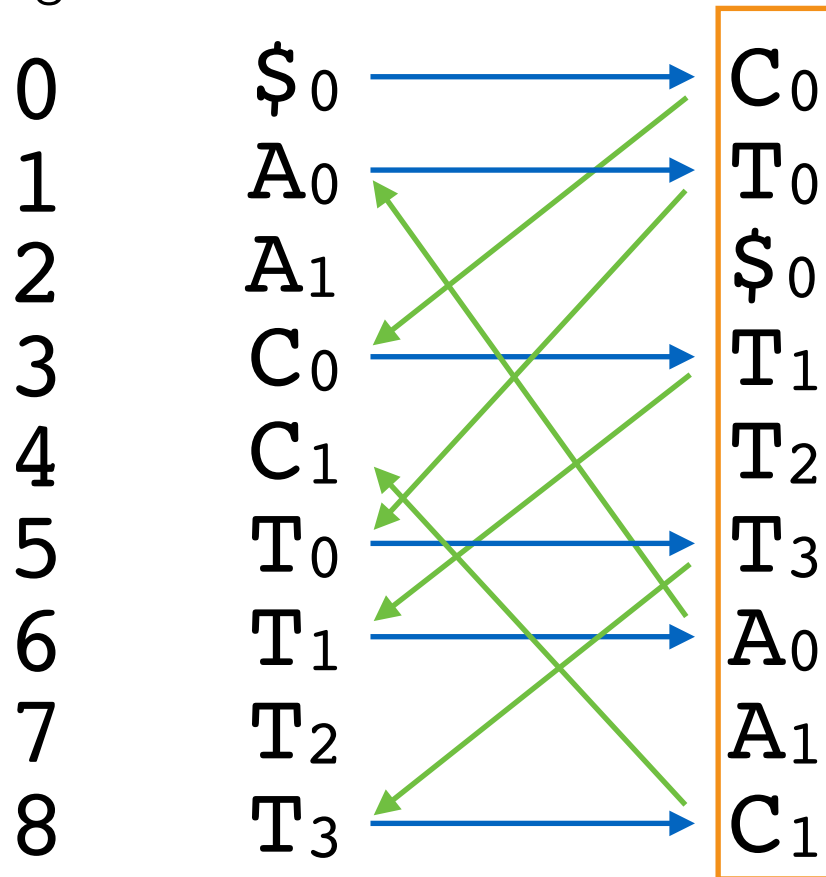
Renglón



C₁ T₃ T₀ A₀ T₁ C₀ \$₀
Secuencia original

Revirtiendo la transformación BWT

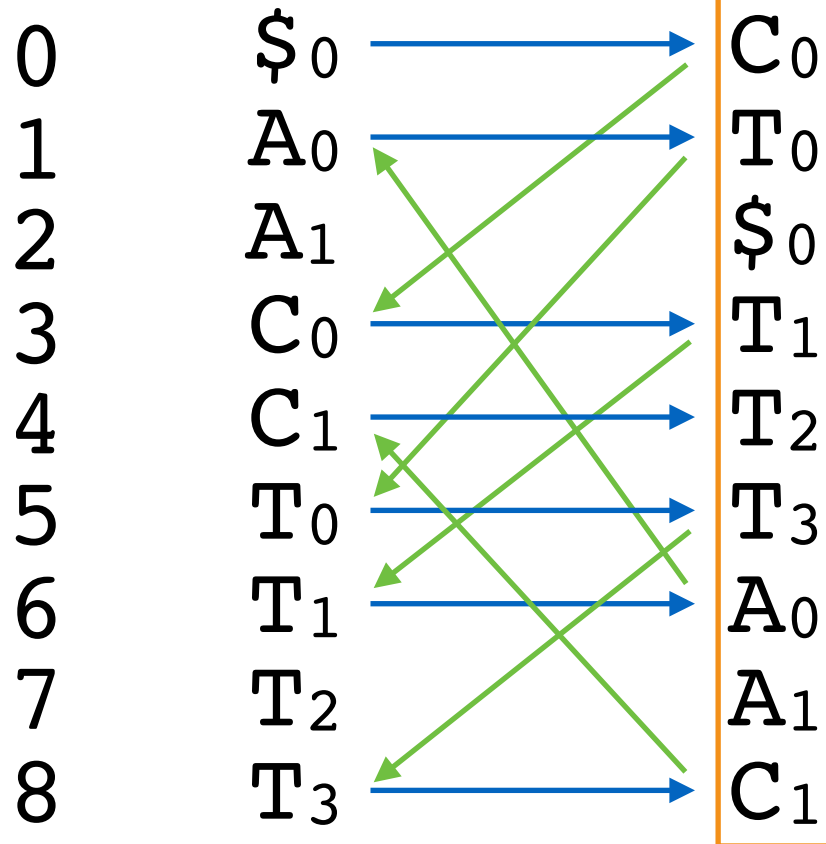
Renglón



C₁ T₃ T₀ A₀ T₁ C₀ \$₀
Secuencia original

Revirtiendo la transformación BWT

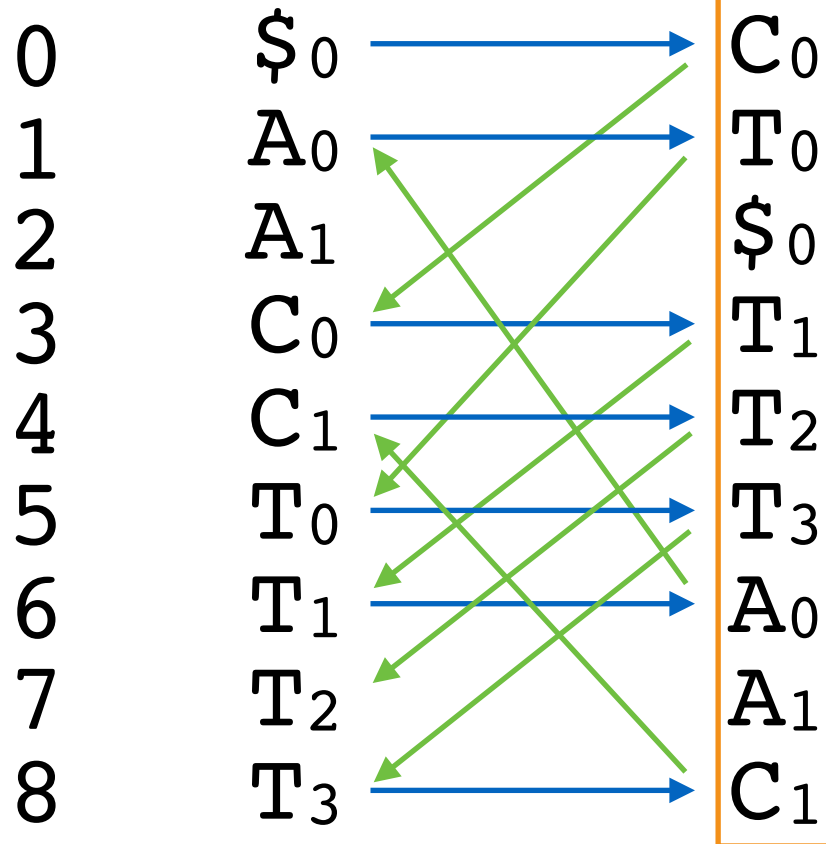
Renglón



T₂ C₁ T₃ T₀ A₀ T₁ C₀ \$₀
Secuencia original

Revirtiendo la transformación BWT

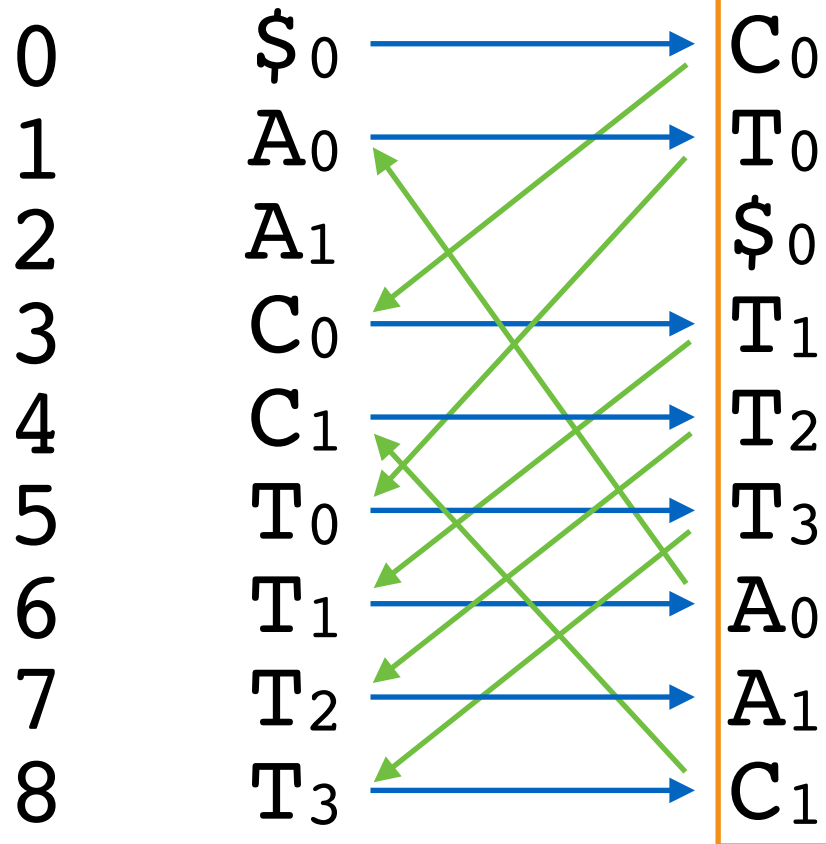
Renglón



T₂ C₁ T₃ T₀ A₀ T₁ C₀ \$₀
Secuencia original

Revirtiendo la transformación BWT

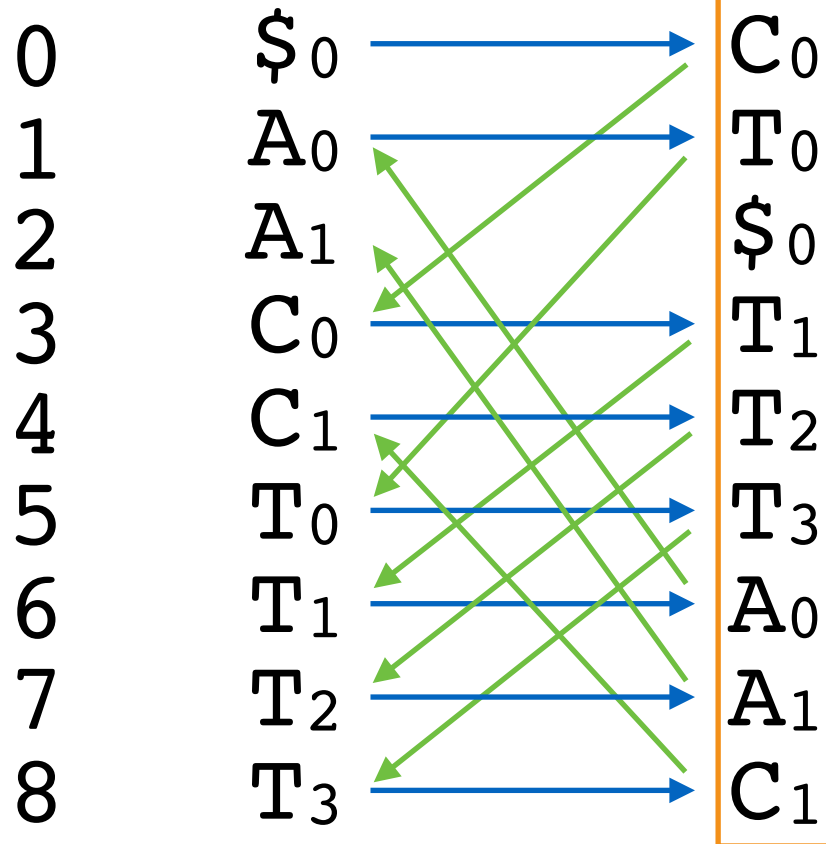
Renglón



A₁ T₂ C₁ T₃ T₀ A₀ T₁ C₀ \$₀
Secuencia original

Revirtiendo la transformación BWT

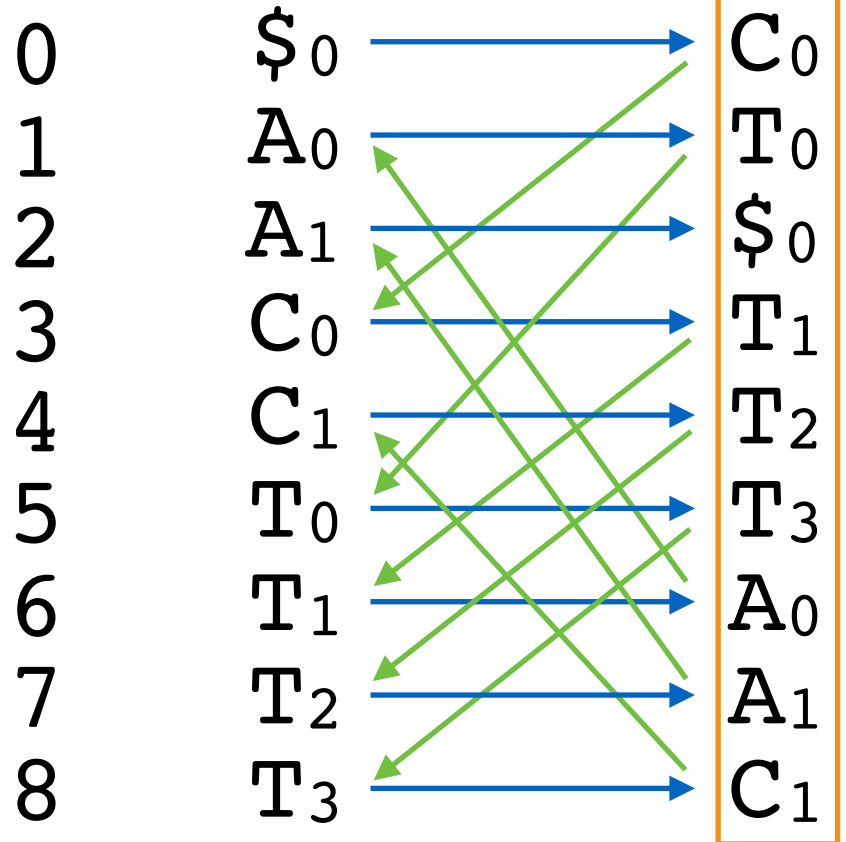
Renglón



A₁ T₂ C₁ T₃ T₀ A₀ T₁ C₀ \$₀
Secuencia original

Revirtiendo la transformación BWT

Renglón



A₁ T₂ C₁ T₃ T₀ A₀ T₁ C₀ \$₀
Secuencia original

Usando BWT para mapear

Renglón

0	\$ ₀	C ₀
1	A ₀	T ₀
2	A ₁	\$ ₀
3	C ₀	T ₁
4	C ₁	T ₂
5	T ₀	T ₃
6	T ₁	A ₀
7	T ₂	A ₁
8	T ₃	C ₁

Usando BWT para mapear

Renglón

0	\$ ₀
1	A ₀
2	A ₁
3	C ₀
4	C ₁
5	T ₀
6	T ₁
7	T ₂
8	T ₃

BWT

C ₀
T ₀
\$ ₀
T ₁
T ₂
T ₃
A ₀
A ₁
C ₁

Lectura: TTATC

Usando BWT para mapear

Renglón

0	\$ ₀
1	A ₀
2	A ₁
3	C ₀
4	C ₁
5	T ₀
6	T ₁
7	T ₂
8	T ₃

BWT

C ₀
T ₀
\$ ₀
T ₁
T ₂
T ₃
A ₀
A ₁
C ₁

Usando BWT para mapear

Renglón

0	\$ ₀	C ₀
1	A ₀	T ₀
2	A ₁	\$ ₀
3	C ₀	T ₁
4	C ₁	T ₂
5	T ₀	T ₃
6	T ₁	A ₀
7	T ₂	A ₁
8	T ₃	C ₁

Lectura: TTATC

Usando BWT para mapear

Renglón

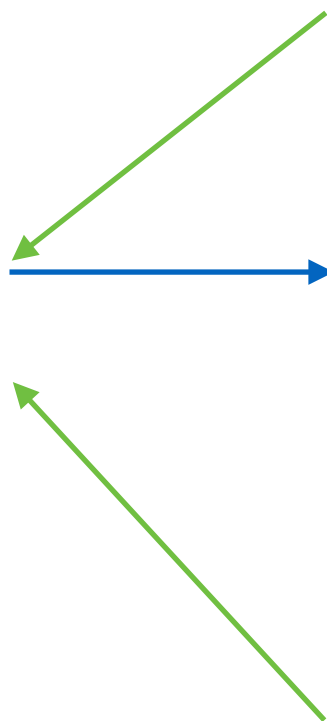
0	\$ ₀	C₀
1	A ₀	T ₀
2	A ₁	\$ ₀
3	C₀	T₁
4	C₁	T₂
5	T ₀	T ₃
6	T ₁	A ₀
7	T ₂	A ₁
8	T ₃	C₁

Lectura: TTAT**C**

Usando BWT para mapear

Renglón

		BWT
0	\$ ₀	C ₀
1	A ₀	T ₀
2	A ₁	\$ ₀
3	C ₀	T ₁
4	C ₁	T ₂
5	T ₀	T ₃
6	T ₁	A ₀
7	T ₂	A ₁
8	T ₃	C ₁

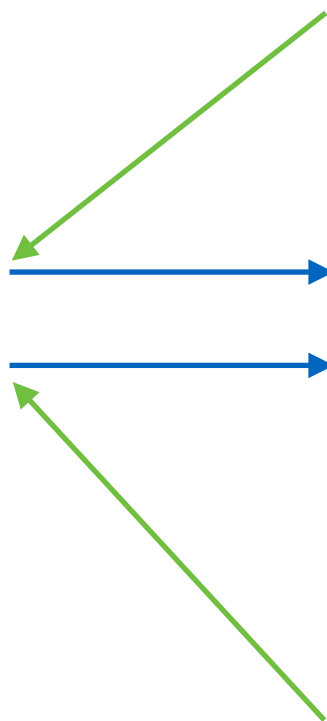


Lectura: TTATC

Usando BWT para mapear

Renglón

		BWT
0	\$ ₀	C ₀
1	A ₀	T ₀
2	A ₁	\$ ₀
3	C ₀	T ₁
4	C ₁	T ₂
5	T ₀	T ₃
6	T ₁	A ₀
7	T ₂	A ₁
8	T ₃	C ₁



Lectura: TTATC

Usando BWT para mapear

Renglón

0	\$ ₀
1	A ₀
2	A ₁
3	C ₀
4	C ₁
5	T ₀
6	T ₁
7	T ₂
8	T ₃

BWT

C ₀
T ₀
\$ ₀
T ₁
T ₂
T ₃
A ₀
A ₁
C ₁

Usando BWT para mapear

Renglón

0	\$ ₀
1	A ₀
2	A ₁
3	C ₀
4	C ₁
5	T ₀
6	T ₁
7	T ₂
8	T ₃

BWT

C ₀
T ₀
\$ ₀
T ₁
T ₂
T ₃
A ₀
A ₁
C ₁

Lectura: TTATC

Usando BWT para mapear

Renglón

0	\$ ₀
1	A ₀
2	A ₁
3	C ₀
4	C ₁
5	T ₀
6	T ₁
7	T ₂
8	T ₃

BWT

C ₀
T ₀
\$ ₀
T ₁
T ₂
T ₃
A ₀
A ₁
C ₁

Lectura: TTATC

Usando BWT para mapear

Renglón

0	\$ ₀
1	A ₀
2	A ₁
3	C ₀
4	C ₁
5	T ₀
6	T ₁
7	T ₂
8	T ₃

BWT

C ₀
T ₀
\$ ₀
T ₁
T ₂
T ₃
A ₀
A ₁
C ₁

Lectura: TTATC

Usando BWT para mapear

Renglón

0	\$ ₀
1	A ₀
2	A ₁
3	C ₀
4	C ₁
5	T ₀
6	T ₁
7	T ₂
8	T ₃

BWT

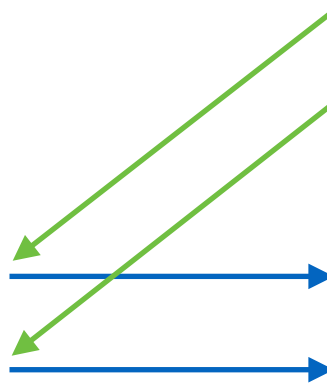
C ₀
T ₀
\$ ₀
T ₁
T ₂
T ₃
A ₀
A ₁
C ₁

Lectura: TTATC

Usando BWT para mapear

Renglón

0	\$ ₀	C ₀
1	A ₀	T ₀
2	A ₁	\$ ₀
3	C ₀	T ₁
4	C ₁	T ₂
5	T ₀	T ₃
6	T ₁	A ₀
7	T ₂	A ₁
8	T ₃	C ₁



Lectura: TTATC

Usando BWT para mapear

Renglón

0	\$ ₀
1	A ₀
2	A ₁
3	C ₀
4	C ₁
5	T ₀
6	T ₁
7	T ₂
8	T ₃

BWT

C ₀
T ₀
\$ ₀
T ₁
T ₂
T ₃
A ₀
A ₁
C ₁

Usando BWT para mapear

Renglón

0	\$ ₀
1	A ₀
2	A ₁
3	C ₀
4	C ₁
5	T ₀
6	T ₁
7	T ₂
8	T ₃

BWT

C ₀
T ₀
\$ ₀
T ₁
T ₂
T ₃
A ₀
A ₁
C ₁

Lectura: TTATC

Usando BWT para mapear

Renglón

0	\$ ₀
1	A ₀
2	A ₁
3	C ₀
4	C ₁
5	T ₀
6	T ₁
7	T ₂
8	T ₃

BWT

C ₀
T ₀
\$ ₀
T ₁
T ₂
T ₃
A ₀
A ₁
C ₁

Lectura: TTATC

Usando BWT para mapear

Renglón

0	\$ ₀		C ₀
1	A ₀	→	T ₀
2	A ₁		\$ ₀
3	C ₀		T ₁
4	C ₁		T ₂
5	T ₀		T ₃
6	T ₁		A ₀
7	T ₂		A ₁
8	T ₃		C ₁

Lectura: TTATC

Usando BWT para mapear

Renglón

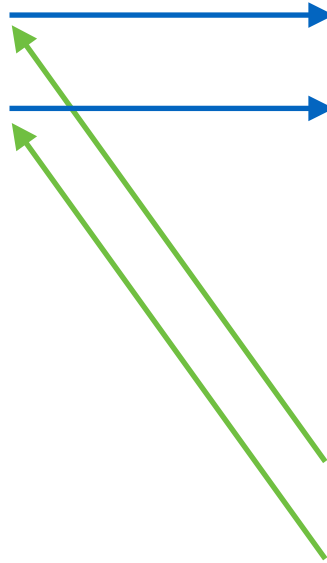
0	\$ ₀		C ₀
1	A ₀	→	T ₀
2	A ₁	↖ ↘	\$ ₀
3	C ₀		T ₁
4	C ₁		T ₂
5	T ₀		T ₃
6	T ₁		A ₀
7	T ₂		A ₁
8	T ₃		C ₁

Lectura: TTATC

Usando BWT para mapear

Renglón

		BWT
0	\$ ₀	C ₀
1	A ₀	T ₀
2	A ₁	\$ ₀
3	C ₀	T ₁
4	C ₁	T ₂
5	T ₀	T ₃
6	T ₁	A ₀
7	T ₂	A ₁
8	T ₃	C ₁



Lectura: TTATC

Usando BWT para mapear

Renglón

0	\$ ₀
1	A ₀
2	A ₁
3	C ₀
4	C ₁
5	T ₀
6	T ₁
7	T ₂
8	T ₃

BWT

C ₀
T ₀
\$ ₀
T ₁
T ₂
T ₃
A ₀
A ₁
C ₁

Usando BWT para mapear

Renglón

0	\$ ₀
1	A ₀
2	A ₁
3	C ₀
4	C ₁
5	T ₀
6	T ₁
7	T ₂
8	T ₃

BWT

C ₀
T ₀
\$ ₀
T ₁
T ₂
T ₃
A ₀
A ₁
C ₁

Lectura: **T**TATC

Usando BWT para mapear

Renglón

0	\$ ₀
1	A ₀
2	A ₁
3	C ₀
4	C ₁
5	T ₀
6	T ₁
7	T ₂
8	T ₃

BWT

C ₀
T ₀
\$ ₀
T ₁
T ₂
T ₃
A ₀
A ₁
C ₁

Lectura: **T**TATC

Usando BWT para mapear

Renglón

0	\$ ₀
1	A ₀
2	A ₁
3	C ₀
4	C ₁
5	T ₀
6	T ₁
7	T ₂
8	T ₃

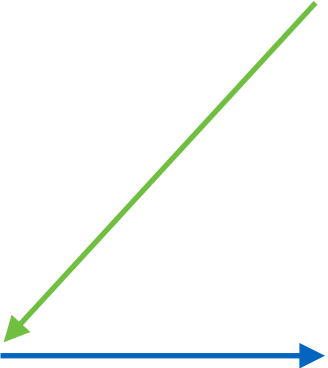
BWT

C ₀
T ₀
\$ ₀
T ₁
T ₂
T ₃
A ₀
A ₁
C ₁

Lectura: **T**TATC

Usando BWT para mapear

Renglón		BWT
0	\$ ₀	C ₀
1	A ₀	T ₀
2	A ₁	\$ ₀
3	C ₀	T ₁
4	C ₁	T ₂
5	T ₀	T ₃
6	T ₁	A ₀
7	T ₂	A ₁
8	T ₃	C ₁



Lectura: **T**TATC

La lectura
mapea a nuestra
secuencia pero ...
¿dónde está en el
genoma?

Usando BWT para mapear

Renglón Suffix array

0	\$ ₀	C ₀	8
1	A ₀	T ₀	5
2	A ₁	\$ ₀	0
3	C ₀	T ₁	7
4	C ₁	T ₂	2
5	T ₀	T ₃	4
6	T ₁	A ₀	6
7	T ₂	A ₁	1
8	T ₃	C ₁	3

BWT

Un sufijo podría indicarnos donde se encuentra en la secuencia original. Usa mucho espacio si tenemos millones de posiciones

Usando BWT para mapear

Renglón

Suffix array

0	\$ ₀	C ₀	8
1	A ₀	T ₀	5
2	A ₁	\$ ₀	0
3	C ₀	T ₁	7
4	C ₁	T ₂	2
5	T ₀	T ₃	4
6	T ₁	A ₀	6
7	T ₂	A ₁	1
8	T ₃	C ₁	3

Lectura: **T**TATC

Un sufijo podría indicarnos donde se encuentra en la secuencia original. Usa mucho espacio si tenemos millones de posiciones

BWT

Usando BWT para mapear

Renglón

Suffix array

0	\$ ₀	C ₀	8
1	A ₀	T ₀	5
2	A ₁	\$ ₀	0
3	C ₀	T ₁	7
4	C ₁	T ₂	2
5	T ₀	T ₃	4
6	T ₁	A ₀	6
7	T ₂	A ₁	1
8	T ₃	C ₁	3

BWT

Lectura: **T**TATC

A₁ T₂ C₁ **T**₃ T₀ A₀ T₁ C₀ \$₀

Un sufijo podría indicarnos donde se encuentra en la secuencia original. Usa mucho espacio si tenemos millones de posiciones

Full-text Minute-size (FM) index

Renglón

0	\$ ₀
1	A ₀
2	A ₁
3	C ₀
4	C ₁
5	T ₀
6	T ₁
7	T ₂
8	T ₃

C ₀
T ₀
\$ ₀
T ₁
T ₂
T ₃
A ₀
A ₁
C ₁

BWT

Checkpoints

[A:0,T:1,C:1,G:0]

[A:2,T:4,C:1,G:0]

Lo que hacemos es utilizar “checkpoints” a lo largo del BWT para indicarnos la posición. Cuando encontramos un match, buscamos el “checkpoint” más cercano para identificar su posición en la referencia (genoma o transcriptoma).

A esto se le conoce como FM index y es muy pequeño.

Full-text Minute-size (FM) index

Renglón

0	\$ ₀
1	A ₀
2	A ₁
3	C ₀
4	C ₁
5	T ₀
6	T ₁
7	T ₂
8	T ₃

Checkpoints

C ₀
T ₀
\$ ₀
T ₁
T ₂
T ₃
A ₀
A ₁
C ₁

[A:0,T:1,C:1,G:0]

[A:2,T:4,C:1,G:0]

BWT

Lectura: **T**TATC

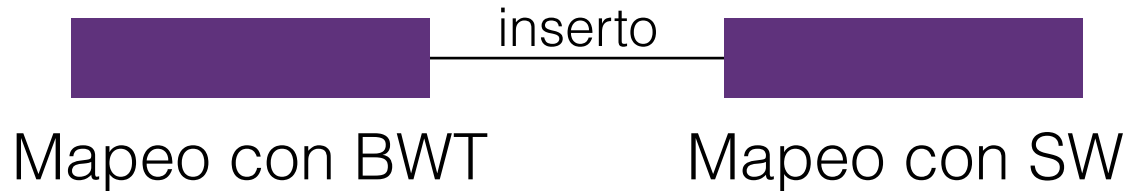
Lo que hacemos es utilizar “checkpoints” a lo largo del BWT para indicarnos la posición. Cuando encontramos un match, buscamos el “checkpoint” más cercano para identificar su posición en la referencia (genoma o transcriptoma).

A esto se le conoce como FM index y es muy pequeño.

Errores o Mismatches

- De no identificarse ningún alineamiento perfecto de la lectura a la secuencia de referencia se toman los alineamientos parciales y se permuta el nucleótido candidato a mismatch (A,T, C,G) y se trata de seguir extendiendo el sitio con similitud a la lectura de interés.
- A esto se le conoce como “backtracking” y generalmente se limita a un número arbitrario de ciclos para evitar incrementar demasiado el tiempo de alineamiento.
- Se hace más backtracking en nucleótidos con baja calidad.
- Dado que el tiempo de cálculo es lineal, no es tan tardado tratar de hacer esto para buscar el lugar de origen de lecturas con errores.

Lecturas en pares (paired-end)



- Muchas veces una sola lectura se encuentra usando alineamiento via BWT. Dado que sabemos el tamaño aproximado del inserto algunos algoritmos utilizan alineamientos Smith-Waterman (SW) para encontrar su par en la región vecina.

Programas para alinear lecturas a una referencia

- bowtie2 - **TopHat** (<https://ccb.jhu.edu/software/tophat/index.shtml>)
- bowtie (<http://bowtie-bio.sourceforge.net/index.shtml>)
- BWA (<http://bio-bwa.sourceforge.net/>)
- STAR (<https://github.com/alexdobin/STAR>)

Programas para alinear **transcritos** a una referencia

- **GMAP** (<http://research-pub.gene.com/gmap/>)
- Blat (<https://genome.ucsc.edu/goldenpath/help/blatSpec.html>)
- Exonerate (<http://www.animalgenome.org/bioinfo/resources/manuals/exonerate/beginner.html>)

Práctica - alineando lecturas usando Bowtie

https://liz-fernandez.github.io/PBI_transcriptomics/