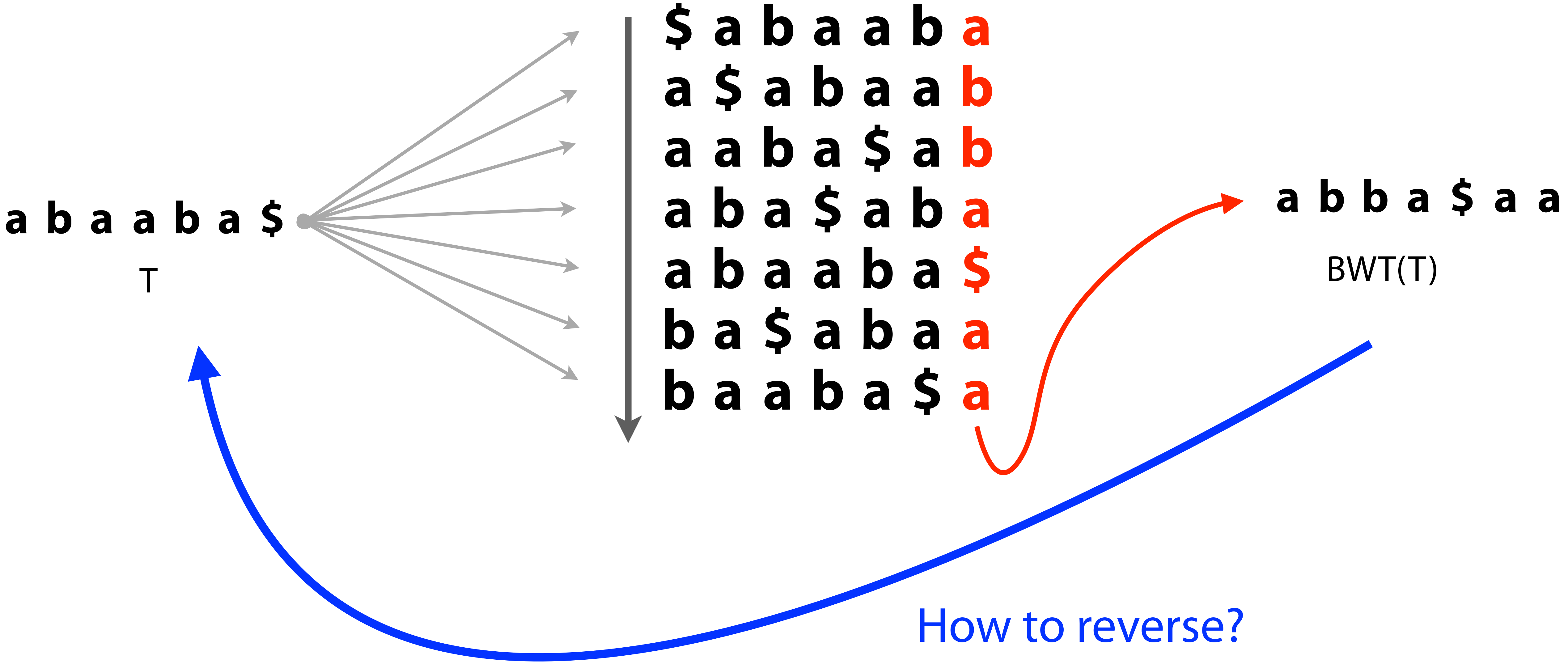
Reversing the BWT

Ben Langmead

JOHNS HOPKINS
WHITING SCHOOL
of ENGINEERING

Department of Computer Science

Burrows-Wheeler Transform



a b a a b a $

T

```
$ a b a a b a
a $ a b a a b
a a b a $ a b
a b a $ a b a
a b a a b a $
b a $ a b a a
b a a b a $ a
```

a b b a $ a a

BWT(T)

How to reverse?

Burrows M, Wheeler DJ: A block sorting lossless data compression algorithm.
Digital Equipment Corporation, Palo Alto, CA 1994, Technical Report 124; 1994

Burrows-Wheeler Transform

Give each character in *T* a ***rank:***

a b a a b a $

Burrows-Wheeler Transform

Give each character in *T* a ***rank:***

$$\mathbf{a}_{0}\ \mathbf{b}_{0}\ \mathbf{a}_{1}\ \mathbf{a}_{2}\ \mathbf{b}_{1}\ \mathbf{a}_{3}\ \mathbf{\$}$$

Ranks aren’t explicitly stored; we use them to distinguish occurrences

Let’s re-write the BWM with ranks...

Burrows-Wheeler Transform

BWM with ranks:

F						*L*
$	a_0	b_0	a_1	a_2	b_1	$\mathbf{a}_3$
$\mathbf{a}_3$	$	a_0	b_0	a_1	a_2	b_1
$\mathbf{a}_1$	a_2	b_1	a_3	$	a_0	b_0
$\mathbf{a}_2$	b_1	a_3	$	a_0	b_0	$\mathbf{a}_1$
$\mathbf{a}_0$	b_0	a_1	a_2	b_1	a_3	$
b_1	a_3	$	a_0	b_0	a_1	$\mathbf{a}_2$
b_0	a_1	a_2	b_1	a_3	$	$\mathbf{a}_0$

Look at first and last columns, called *F* and *L*

And look at just the **a**s

as occur in the same order in *F* and *L*. As we look down columns, in both cases we see: $\mathbf{a}_3$, $\mathbf{a}_1$, $\mathbf{a}_2$, $\mathbf{a}_0$

Burrows-Wheeler Transform

BWM with ranks:

F						*L*
$\$$	a_0	b_0	a_1	a_2	b_1	a_3
a_3	$\$$	a_0	b_0	a_1	a_2	**b_1**
a_1	a_2	b_1	a_3	$\$$	a_0	**b_0**
a_2	b_1	a_3	$\$$	a_0	b_0	a_1
a_0	b_0	a_1	a_2	b_1	a_3	$\$$
b_1	a_3	$\$$	a_0	b_0	a_1	a_2
b_0	a_1	a_2	b_1	a_3	$\$$	a_0

Same with **b**s: **b_1**, **b_0**

Burrows-Wheeler Transform

BWM with ranks:

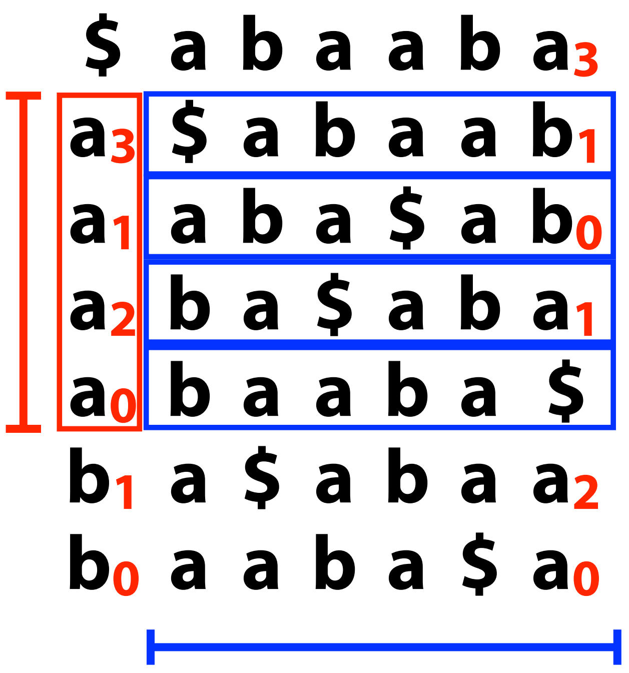
F						L
\$	a_0	b_0	a_1	a_2	b_1	a_3
a_3	\$	a_0	b_0	a_1	a_2	b_1
a_1	a_2	b_1	a_0	\$	a_0	b_0
a_2	b_1	a_3	\$	a_0	b_0	a_1
a_0	b_0	a_1	a_2	b_1	a_3	\$
b_1	a_3	\$	a_0	b_0	a_1	a_2
b_0	a_1	a_2	b_1	a_3	\$	a_0

LF Mapping: The i^{th} occurrence of a character $c \in \Sigma$ in L and the i^{th} occurrence of c in F correspond to the *same* occurrence in T (i.e. have ***same rank***)

Burrows-Wheeler Transform

Why does the LF Mapping hold?

Why are these **a**s in this order relative to each other?



$\$\ a\ b\ a\ a\ b\ a_3$

$a_3\ \$\ a\ b\ a\ a\ b_1$

$a_1\ a\ b\ a\ \$\ a\ b_0$

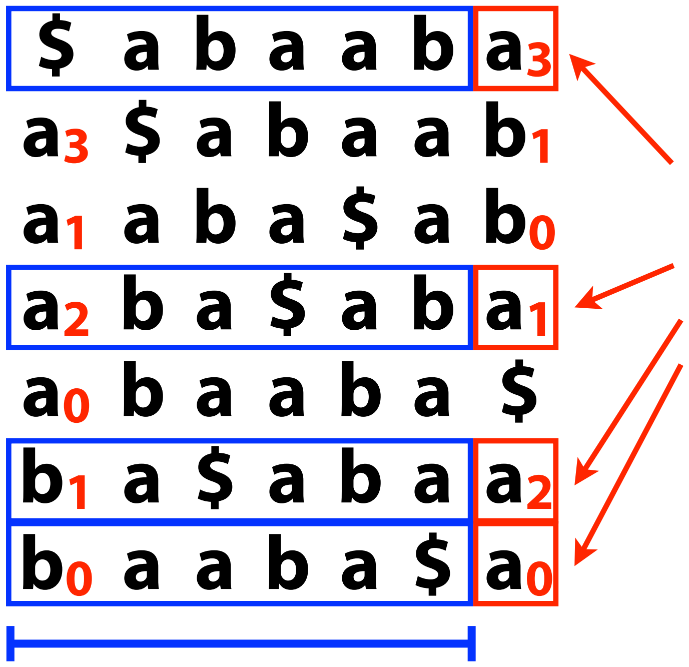
$a_2\ b\ a\ \$\ a\ b\ a_1$

$a_0\ b\ a\ a\ b\ a\ \$$

$b_1\ a\ \$\ a\ b\ a\ a_2$

$b_0\ a\ a\ b\ a\ \$\ a_0$

They're sorted by right-context



$\$\ a\ b\ a\ a\ b\ a_3$

$a_3\ \$\ a\ b\ a\ a\ b_1$

$a_1\ a\ b\ a\ \$\ a\ b_0$

$a_2\ b\ a\ \$\ a\ b\ a_1$

$a_0\ b\ a\ a\ b\ a\ \$$

$b_1\ a\ \$\ a\ b\ a\ a_2$

$b_0\ a\ a\ b\ a\ \$\ a_0$

Why are these **a**s in this order relative to each other?

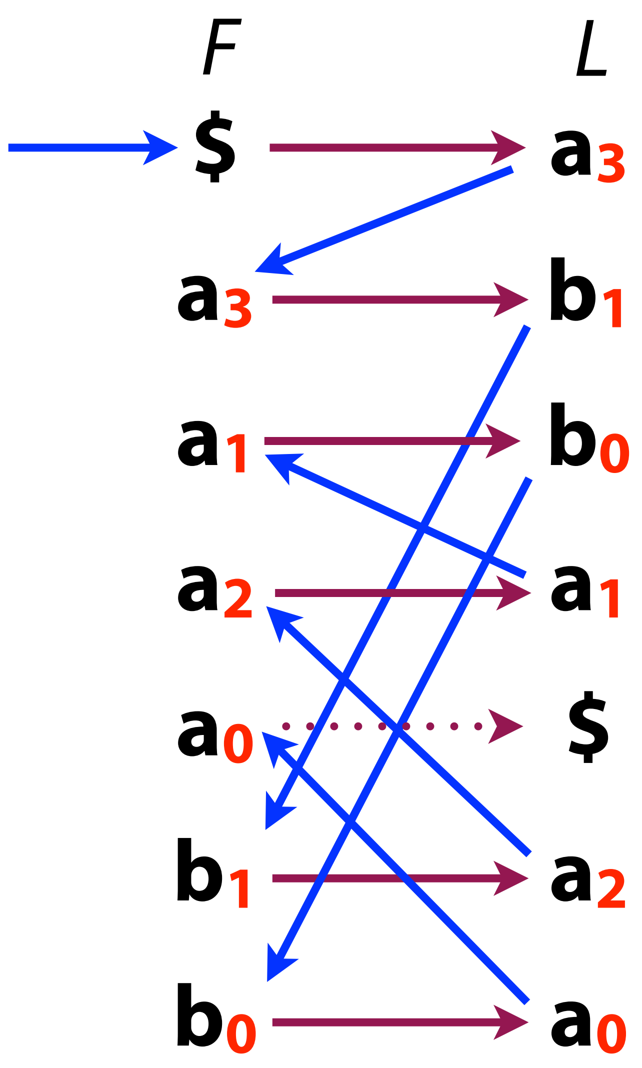
They're sorted by right-context

Occurrences of c in F are sorted by right-context; same for L

Burrows-Wheeler Transform

Reverse BWT(T) starting at right end of *T*, moving left

T: **a**$_0$ **b**$_0$ **a**$_1$ **a**$_2$ **b**$_1$ **a**$_3$ **$**



Start in first row. *F* must have **$**.

L contains character prior: **a**$_3$

Jump to row *beginning* with **a**$_3$.

L contains character just prior: **b**$_1$.

Repeat for **b**$_1$, get **a**$_2$

Repeat for **a**$_2$, get **a**$_1$

Repeat for **a**$_1$, get **b**$_0$

Repeat for **b**$_0$, get **a**$_0$

Repeat for **a**$_0$, get **$** (done)

We visited (backwards) T's chars:

a$_0$ **b**$_0$ **a**$_1$ **a**$_2$ **b**$_1$ **a**$_3$ **$**