BWT for repetitive texts, part 1: ***runs***!

Ben Langmead

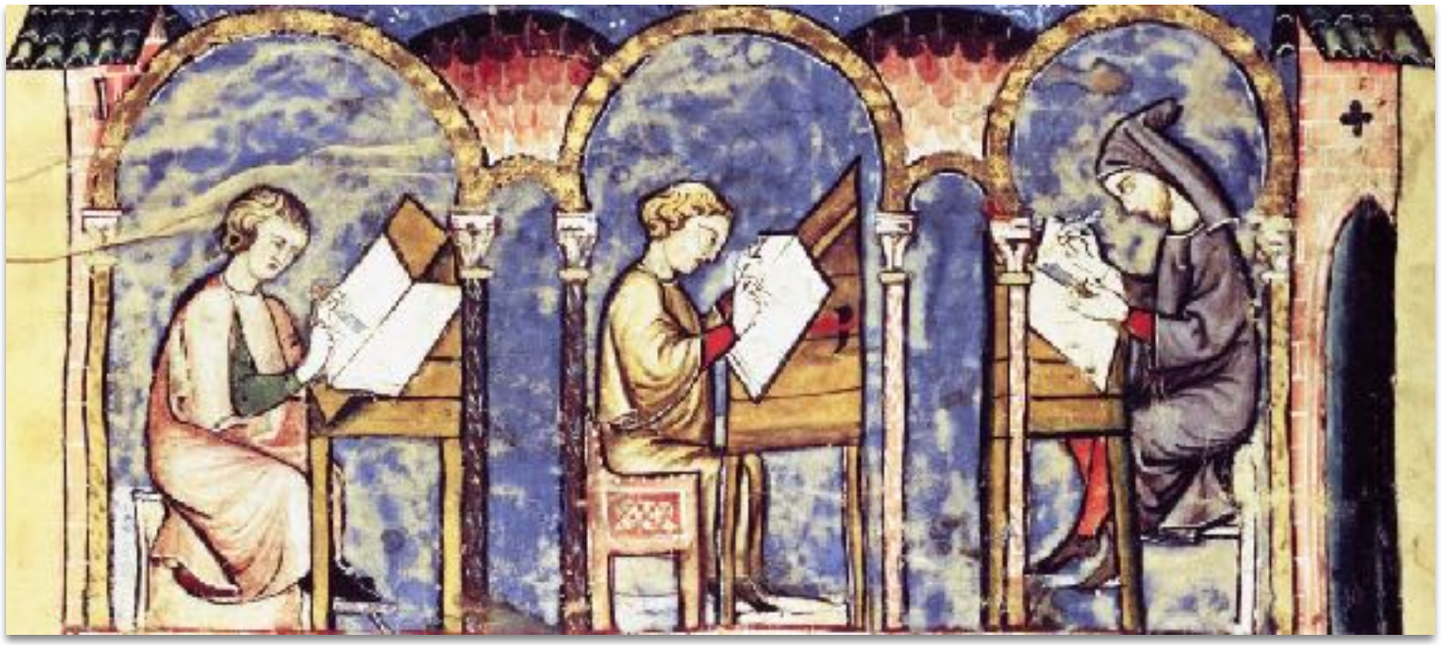
JOHNS HOPKINS
WHITING SCHOOL
of ENGINEERING

Department of Computer Science

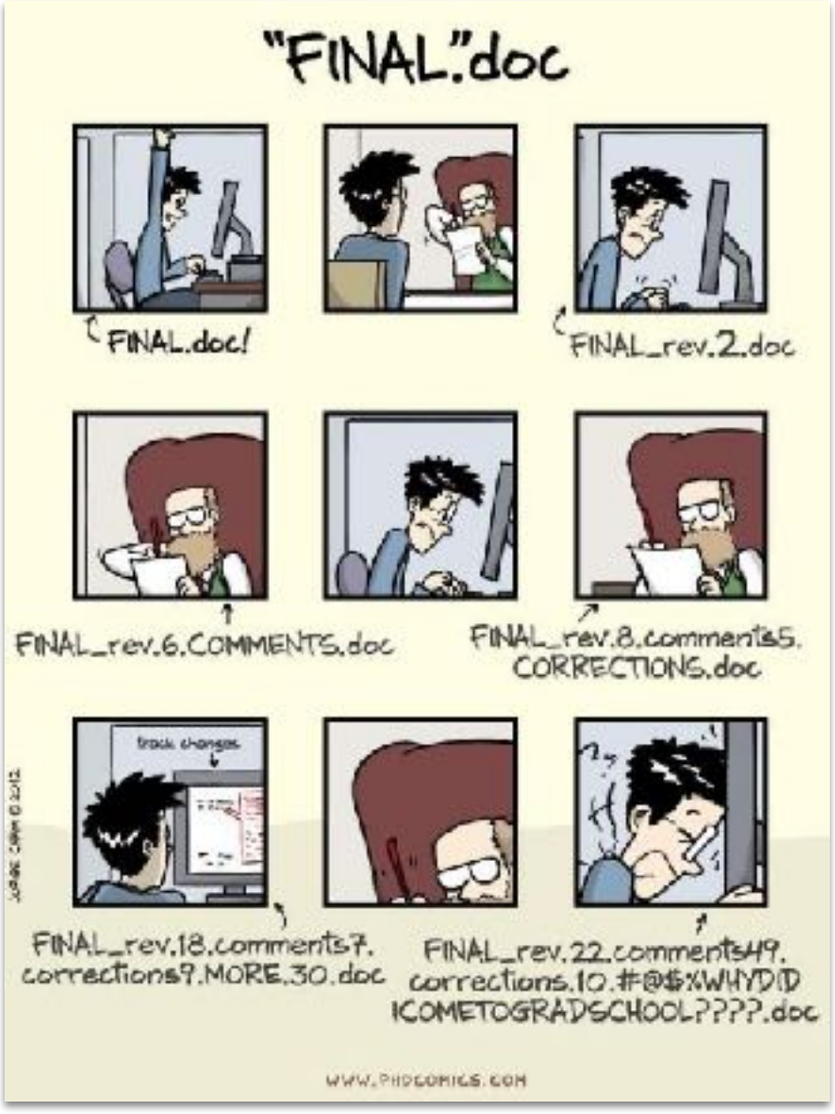
Texts as copies

Real-world large text datasets frequently come from ***make-a-copy-make-a-change*** processes

Texts as copies

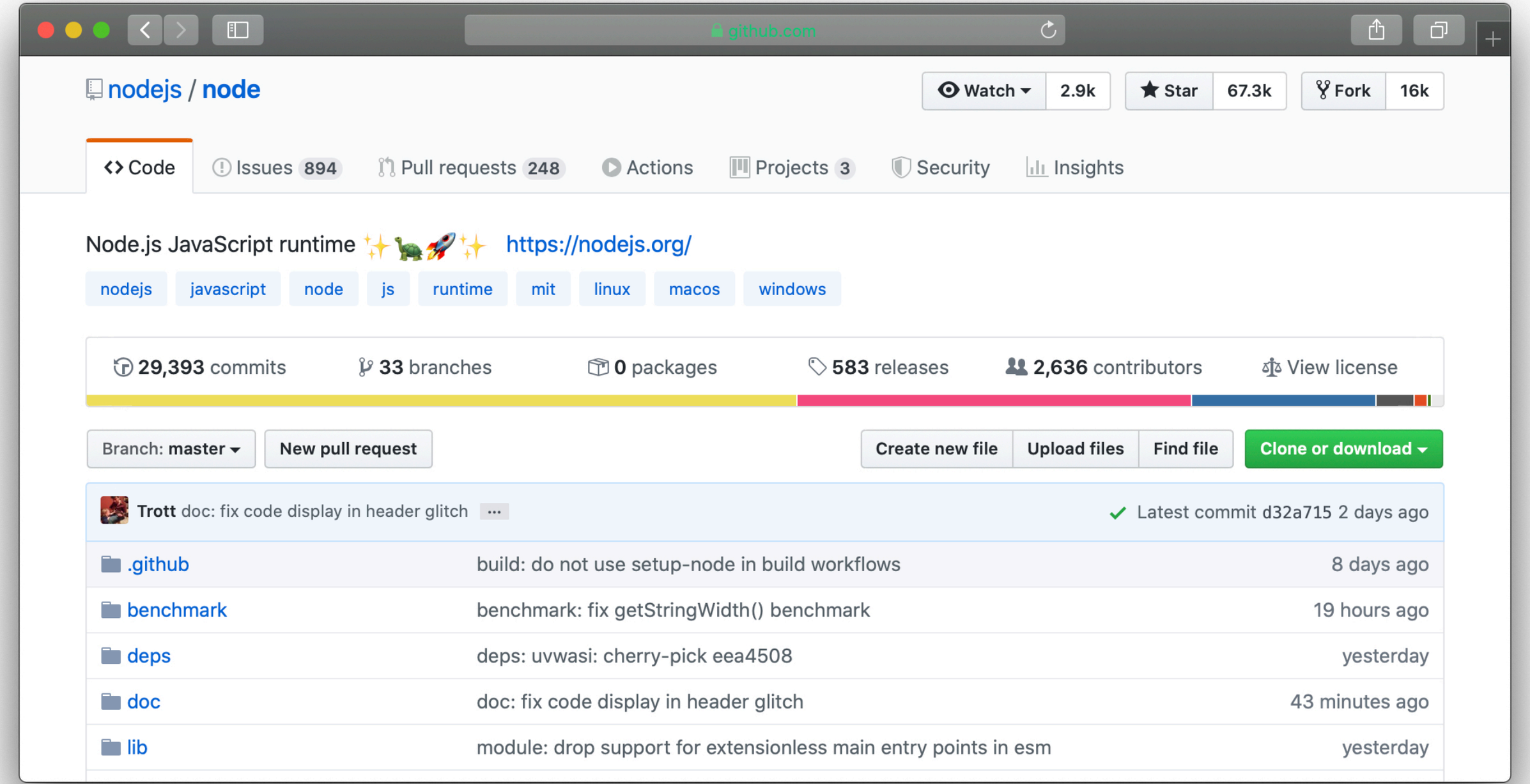


Scriptorium, from manuscript in the Biblioteca de San Lorenzo de El Escorial, Madrid, Spain, c. 14th century AD
(c/o medievalfragments.wordpress.com)



http://phdcomics.com/comics/archive.php?comicid=1531

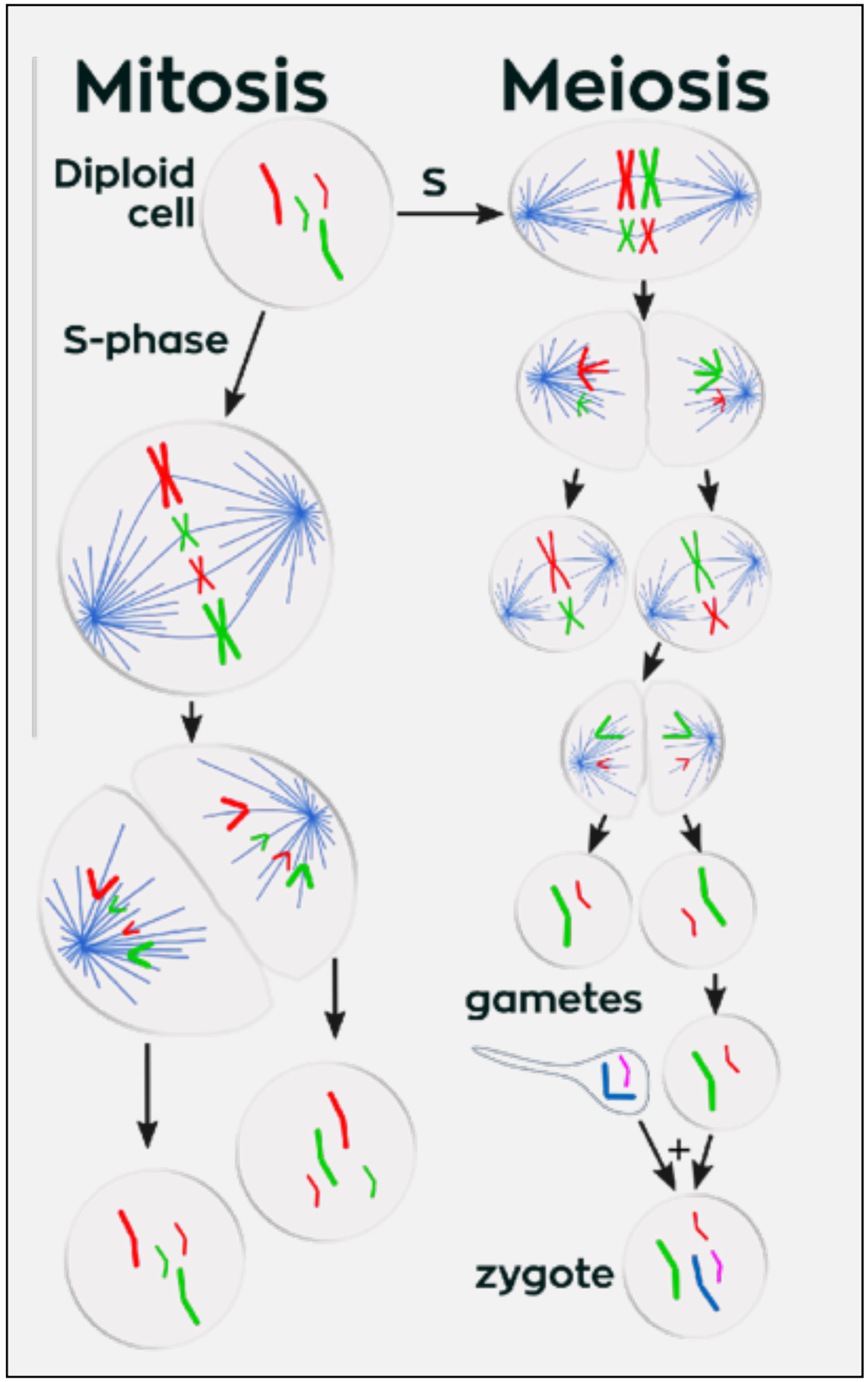
Texts as copies



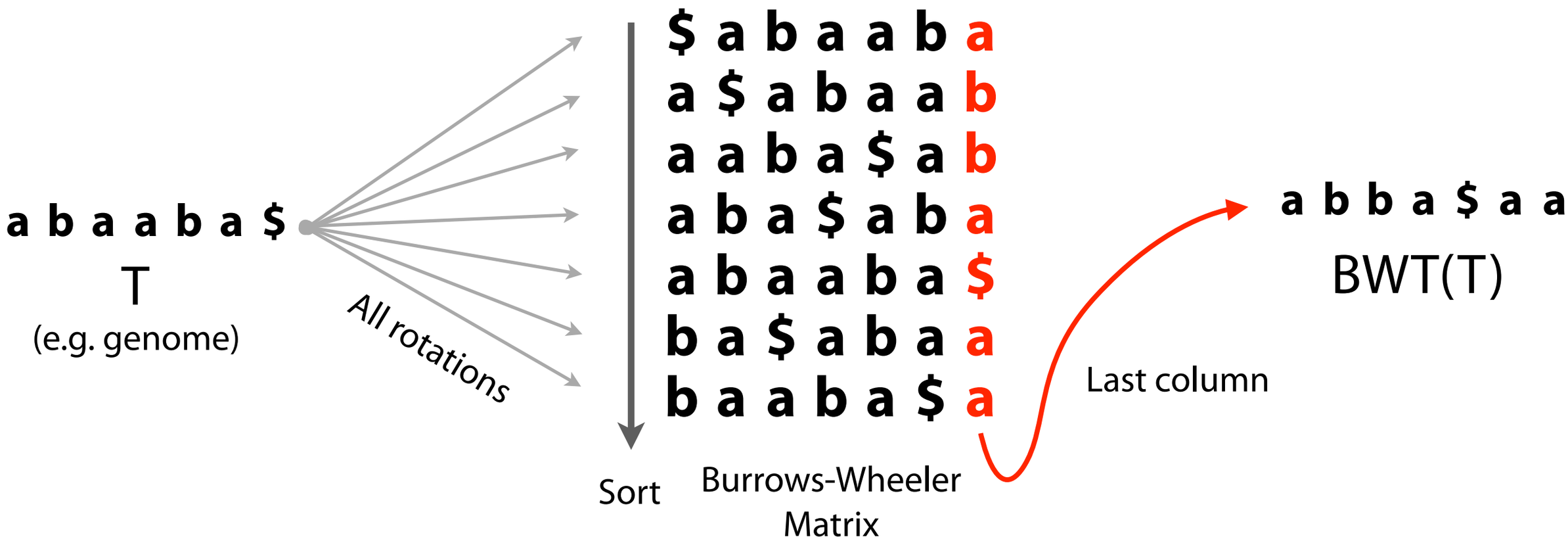
Texts as copies

The DNA in one of your cells comes from a chain of copying (**mitosis**) & mixing (**meiosis**) events

https://upload.wikimedia.org/wikipedia/commons/thumb/d/df/Three_cell_growth_types.svg/1920px-Three_cell_growth_types.svg.png



FM Index



BWT reorders the letters according to alphabetical order of their **right contexts** in T...

Burrows-Wheeler Transform

Ordered by *right-context*

...bringing characters with similar contexts together in **runs**

final char (L)	sorted rotations
a	n to decompress. It achieves compression
o	n to perform only comparisons to a depth
o	n transformation} This section describes
o	n transformation} We use the example and
o	n treats the right-hand side as the most
a	n tree for each 16 kbyte input block, enc
a	n tree in the output stream, then encodes
i	n turn, set $L[i]$ to be the
i	n turn, set $R[i]$ to the
o	n unusual data. Like the algorithm of Man
a	n use a single set of probabilities table
e	n using the positions of the suffixes in
i	n value at a given point in the vector $R
e	n we present modifications that improve t
e	n when the block size is quite large. Ho
i	n which codes that have not been seen in
i	n with ch appear in the {\em same order
i	n with ch. In our exam
o	n with Huffman or arithmetic coding. Bri
o	n with figures given by Bell~\cite{bell}.

Figure 1: Example of sorted rotations. Twenty consecutive rotations from the sorted list of rotations of a version of this paper are shown, together with the final character of each rotation.

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

BWT runs

E.g. for a text where `rectangle` appears many times, `ectangle` tends to be preceded by `r`

```
T  rectangular_rectangle_divided_into_rectangles$
```

BWT runs

E.g. for a text where `rectangle` appears many times, `ectangle` tends to be preceded by `r`

These `r`s come together in a BWT **run**

T	rectangular_rectangle_divided_into_rectangles$
BWT(T)	sedrotttleeeei_lrrrdlnnnv_duggaaaita__$ecccngi

BWT runs

When T is more repetitive, BWT runs are longer & fewer

	Avg. run length
T Tomorrow_and_tomorrow_and_tomorrow$	1.09

BWT runs

When T is more repetitive, BWT runs are longer & fewer

		Avg. run length
T	`Tomorrow_and_tomorrow_and_tomorrow$`	1.09
BWT(T)	`w$wwdd__nnoooaattTmmmrrrrrrooo__ooo`	2.33

BWT runs

When T is more repetitive, BWT runs are longer & fewer

		Avg. run length
T	`Tomorrow_and_tomorrow_and_tomorrow$`	1.09
BWT(T)	`w$wwdd__nnoooaattTmmmrrrrrrooo__ooo`	2.33
T	`It_was_the_best_of_times_it_was_the_worst_of_times$`	1.00
BWT(T)	`s$esttssfftteww_hhmmbootttt_ii__woeeaaressIi_____`	1.76

BWT runs

When T is more repetitive, BWT runs are longer & fewer

		Avg. run length
T	Tomorrow_and_tomorrow_and_tomorrow$	1.09
BWT(T)	w$wwdd__nnoooaattTmmmrrrrrrooo__ooo	2.33
T	It_was_the_best_of_times_it_was_the_worst_of_times$	1.00
BWT(T)	s$esttssfftteww_hhmmbootttt_ii__woeeaaressIi_____	1.76
T	in_the_jingle_jangle_morning_Ill_come_following_you$	1.04
BWT(T)	u_gleeeengj_mlhl_nnnnt$nwj__lggIolo_iiiiarfcmylo_oo_	1.30

BWT runs

```
row_row_row_your_boat
row_row_row_your_boat
row_row_row_your_boat
```

BWT

BWT runs

```
row_row_row_your_boat
row_row_row_your_boat
row_row_row_your_boat$
```

BWT

↓

```
trrrwwwwwwwww...
```

BWT runs

```
row_row_row_your_boat
row_row_row_your_boat
row_row_row_your_boat$
```

↓ BWT

`trrrwwwwwwwwwooo___bbbyyyrrrrrrrrruuutt$_____aaaoooooooooooo___`

↓ RLE

(**t**, 1), (**r**, 3), (**w**, 9), (**o**, 3), (**_**, 3), (**b**, 3), (**y**, 3), (**r**, 9), (**u**, 3), (**t**, 2), (**$**, 1), (**_**, 6), (**a**, 3), (**o**, 12), (**_**, 3)

Avg run length = 4.27

BWT runs

Runs in $BWT(S \times 1)$ = 14

Runs in $BWT(S \times 2)$ = 15

Runs in $BWT(S \times 3)$ = 15

Runs in $BWT(S \times 4)$ = 15

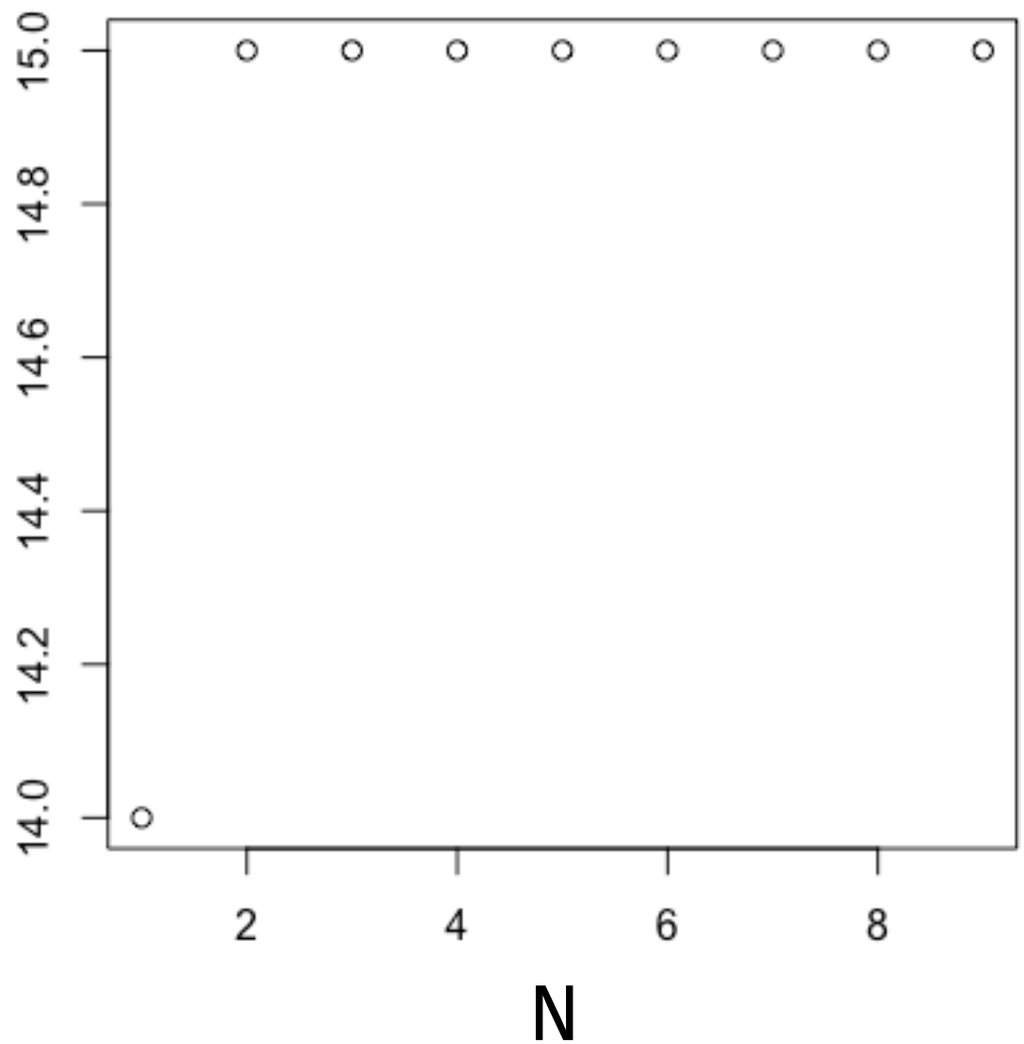
$\vdots$

N

$S =$ "row_row_row_your_boat" $\times$ N

BWT runs

Runs in
$BWT(S \times N)$



$S =$ "row_row_row_your_boat" $\times$ N

High-order entropy

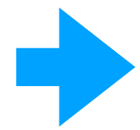
H_k is a weighted sum over all contexts of the zero order empirical entropy of symbols with that context

$$|S|H_k(S) = |S| \sum_{t \in \Sigma^k} \frac{|S_t|}{|S|} \cdot H_0(S_t) \quad \text{for } k > 0$$

S_t is the concatenation of symbols having context t

High-order entropy

```
row_row_row_your_boat
```



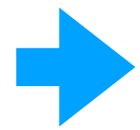
```
row_row_row_your_boat
row_row_row_your_boat
row_row_row_your_boat
```

$\times N$

$$|S|H_k(S) = |S| \sum_{t \in \Sigma^k} \frac{|S_t|}{|S|} \cdot H_0(S_t)$$

High-order entropy

```
row_row_row_your_boat
```



```
row_row_row_your_boat
row_row_row_your_boat
row_row_row_your_boat
```

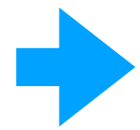
$\times N$

Increases by factor of N

$$|S|H_k(S) = |S| \sum_{t \in \Sigma^k} \frac{\boxed{|S_t|}}{|S|} \cdot H_0(S_t)$$

High-order entropy

```
row_row_row_your_boat
```



```
row_row_row_your_boat
row_row_row_your_boat
row_row_row_your_boat
```

$\times N$

Increases by factor of N

$$|S|\,H_k(S) = |S| \sum_{t \in \Sigma^k} \frac{\boxed{|S_t|}}{\boxed{|S|}} \cdot H_0(S_t)$$

Increases by factor of N

High-order entropy

`row_row_row_your_boat` → `row_row_row_your_boat`
`row_row_row_your_boat`
`row_row_row_your_boat`
$\times N$

Increases by factor of N

$$|S|H_k(S) = |S| \sum_{t \in \Sigma^k} \frac{\boxed{|S_t|}}{\boxed{|S|}} \cdot \boxed{H_0(S_t)}$$

Stays the same

Increases by factor of N

High-order entropy

`row_row_row_your_boat` → `row_row_row_your_boat`
`row_row_row_your_boat`
`row_row_row_your_boat`
$\times N$

Increases by factor of N

$$|S|H_k(S) = |S| \sum_{t \in \Sigma^k} \frac{|S_t|}{|S|} \cdot H_0(S_t)$$

Stays the same

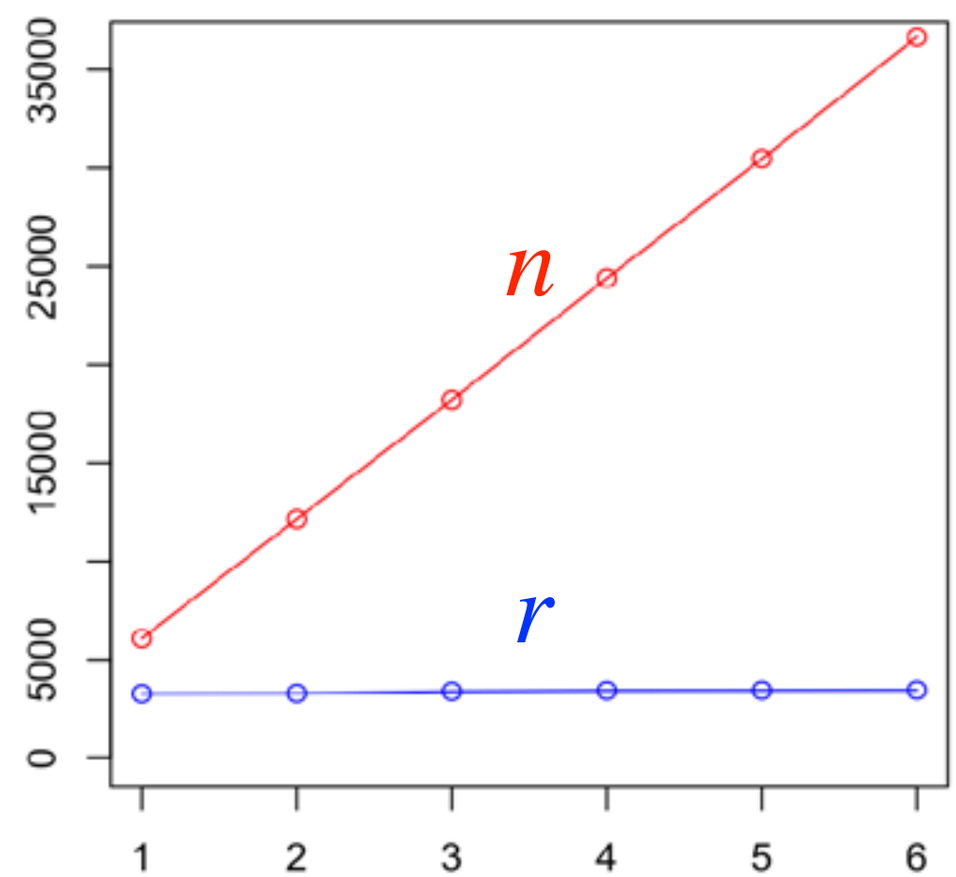
Increases by factor of N

e.g.

$$H_0(\text{rwww}) = H_0(\text{rrrwwwwwwwwwwwww})$$

FM Index

# human genomes	n	r
1	6,072 M	3,264 M
2	12,144 M	3,282 M
3	18,217 M	3,386 M
4	24,408 M	3,423 M
5	30,480 M	3,436 M
6	36,671 M	3,449 M



Kuhnle A, Mun T, Boucher C, Gagie T, Langmead B, Manzini G. Efficient Construction of a Complete Index for Pan-Genomics Read Alignment. J Comput Biol. 2020 Apr;27(4):500-513.

FM Index to r-index

	Count		Locate	
	Space	Time	Space	Time
FM Index (2000)	$O(n)$	$O(m)$	$O(n)$	$O(m + \text{occ})$

Where n is total reference length, m is query-string length, r is total # BWT runs

(log factors omitted)

FM: Ferragina P, and Manzini M. Opportunistic data structures with applications. Proceedings of 41st FOCS. IEEE, 2000.

FM Index to r-index

	Count		Locate	
	Space	Time	Space	Time
FM Index (2000)	$O(n)$	$O(m)$	$O(n)$	$O(m + \text{occ})$
RLFM Index (2005)	$O(r)$	$O(m)$	$O(n)$	$O(m + \text{occ})$

Where n is total reference length, m is query-string length, r is total # BWT runs

(log factors omitted)

FM: Ferragina P, and Manzini M. Opportunistic data structures with applications. Proceedings of 41st FOCS. IEEE, 2000.

RLFM: Mäkinen V, and Navarro G. Succinct suffix arrays based on run-length encoding. Annual Symposium on CPM. Springer, Berlin, Heidelberg. 2005. pp45–56.

FM Index to r-index

	Count		Locate	
	Space	Time	Space	Time
FM Index (2000)	$O(n)$	$O(m)$	$O(n)$	$O(m + \text{occ})$
RLFM Index (2005)	$O(r)$	$O(m)$	$O(n)$	$O(m + \text{occ})$
r-index (2018)	$O(r)$	$O(m)$	$O(r)$	$O(m + \text{occ})$

Where n is total reference length, m is query-string length, r is total # BWT runs

(log factors omitted)

FM: Ferragina P, and Manzini M. Opportunistic data structures with applications. Proceedings of 41st FOCS. IEEE, 2000.

RLFM: Mäkinen V, and Navarro G. Succinct suffix arrays based on run-length encoding. Annual Symposium on CPM. Springer, Berlin, Heidelberg. 2005. pp45–56.

r-index: Gagie T, Navarro G, and Prezza P. Optimal-time text indexing in BWT-runs bounded space. Proceedings of 29th SODA, ACM-SIAM. 2018. pp1459—1477.

FM Index to r-index

Next: How?

	Count		Locate	
	Space	Time	Space	Time
FM Index (2000)	$O(n)$	$O(m)$	$O(n)$	$O(m + \text{occ})$
RLFM Index (2005)	$\boxed{O(r)}$	$O(m)$	$O(n)$	$O(m + \text{occ})$
r-index (2018)	$O(r)$	$O(m)$	$O(r)$	$O(m + \text{occ})$

Where n is total reference length, m is query-string length, r is total # BWT runs

(log factors omitted)

FM: Ferragina P, and Manzini M. Opportunistic data structures with applications. Proceedings of 41st FOCS. IEEE, 2000.

RLFM: Mäkinen V, and Navarro G. Succinct suffix arrays based on run-length encoding. Annual Symposium on CPM. Springer, Berlin, Heidelberg. 2005. pp45–56.

r-index: Gagie T, Navarro G, and Prezza P. Optimal-time text indexing in BWT-runs bounded space. Proceedings of 29th SODA, ACM-SIAM. 2018. pp1459—1477.

FM Index to r-index

	Count		Locate	
	Space	Time	Space	Time
FM Index (2000)	$O(n)$	$O(m)$	$O(n)$	$O(m + \text{occ})$
RLFM Index (2005)	$\boxed{O(r)}$	$O(m)$	$O(n)$	$O(m + \text{occ})$
r-index (2018)	$O(r)$	$O(m)$	$\boxed{O(r)}$	$O(m + \text{occ})$

Next: How?

Later: How?

Where n is total reference length, m is query-string length, r is total # BWT runs

(log factors omitted)

FM: Ferragina P, and Manzini M. Opportunistic data structures with applications. Proceedings of 41st FOCS. IEEE, 2000.

RLFM: Mäkinen V, and Navarro G. Succinct suffix arrays based on run-length encoding. Annual Symposium on CPM. Springer, Berlin, Heidelberg. 2005. pp45–56.

r-index: Gagie T, Navarro G, and Prezza P. Optimal-time text indexing in BWT-runs bounded space. Proceedings of 29th SODA, ACM-SIAM. 2018. pp1459—1477.