UNIVERSITÀ
di VERONA

A theoretical and experimental analysis of BWT variants for string collections

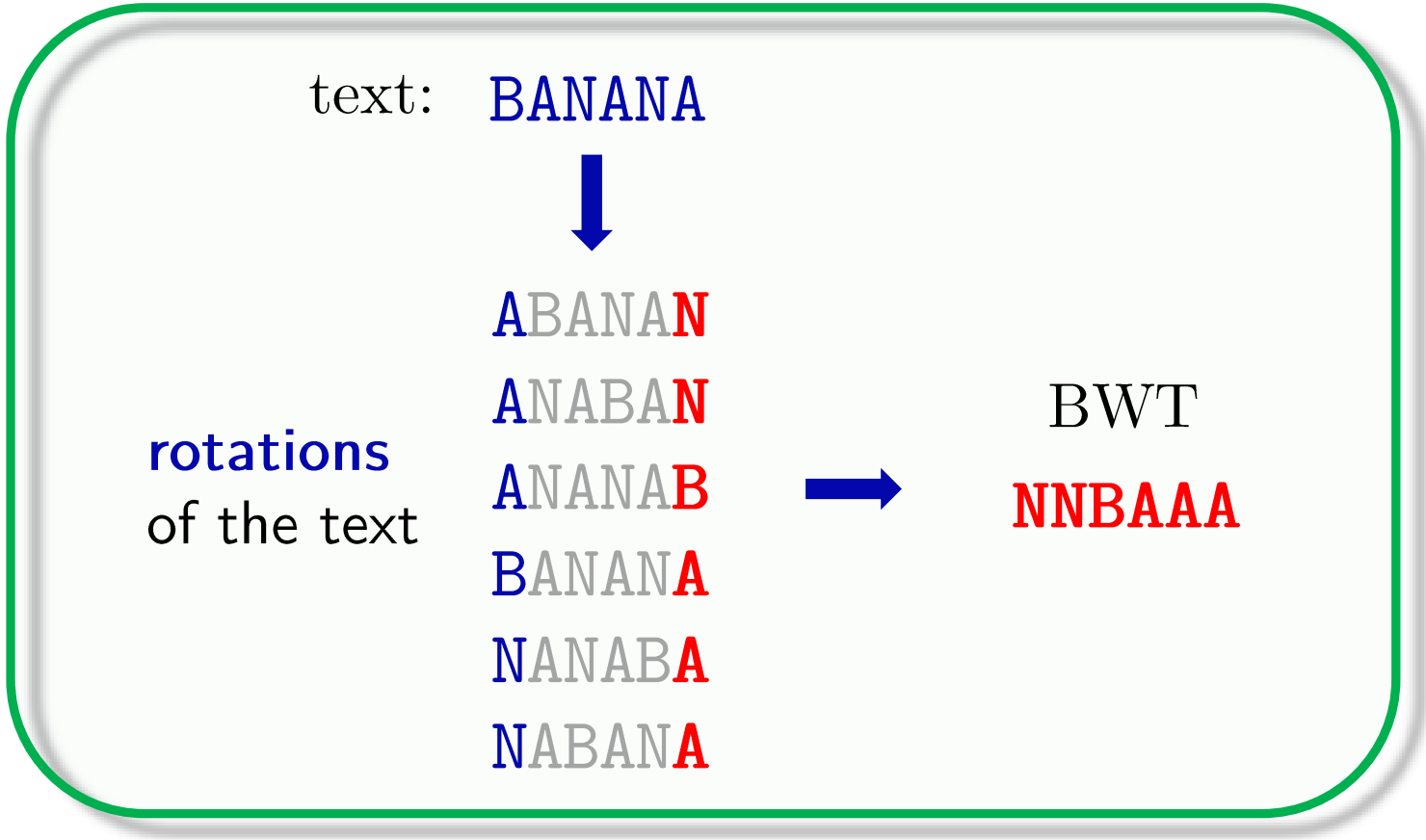
Davide Cenzato and Zsuzsanna Lipták

University of Verona, Department of Computer Science

CPM 2022, June 27 – 29, 2022 - Prague, Czech Republic

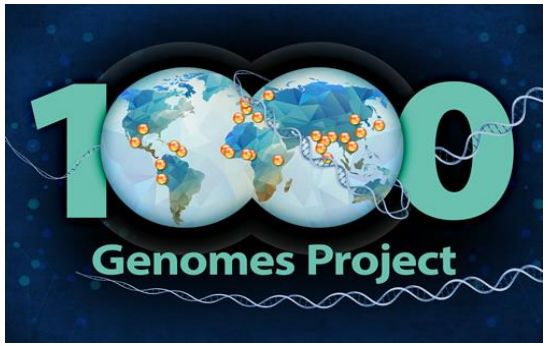
The Burrows-Wheeler-Transform (BWT)

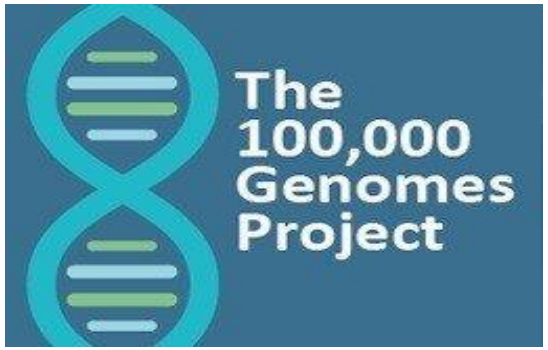
- basis of several compressed data structures for strings



Large string collections are highly abundant

- focus has moved from single strings to collections of strings
- need for compressed data structures for storing and processing large datasets





1001 Genomes

The Burrows-Wheeler-Transform for string collections

- basis of several compressed data structures for strings
- originally defined for single sequences
- several tools in literature computing different variants

text

BANANA

ABANAN
ANABAN
ANANAB
BANANA
NANABA
NABANA

text	BWT
ATATGTGAACGATCAGGA	GGACAGTTATGCATAAGA

string collection	?
{ATATG,TGA,ACG,ATCA,GGA}	

The BWT variants of string collections

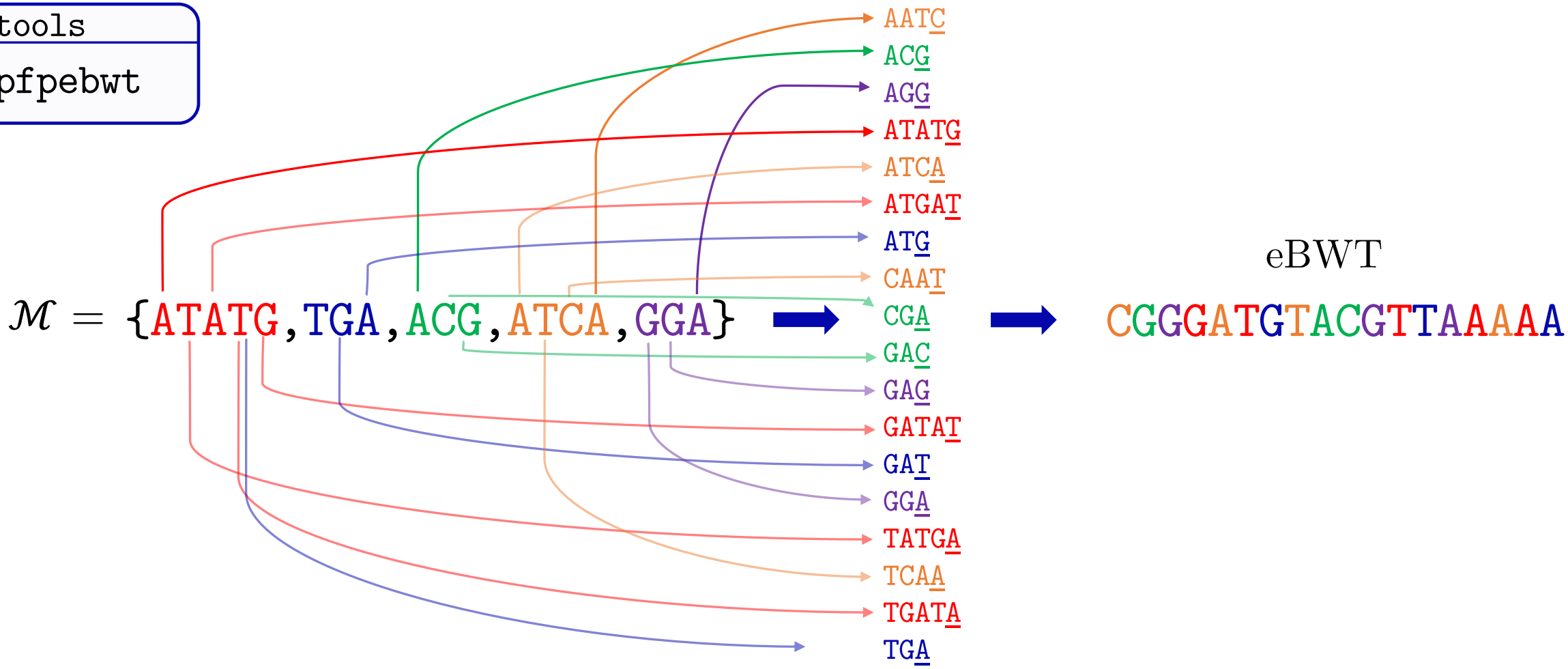
variant (our terminology)	result on example	tools
eBWT	CGGGATGTACGTTAAAAA	pfpebwt
dolEBWT	GGAAACGG$$$TTACTGTAAA	G2BWT, pfpebwt, msbwt
mdolBWT	GAGAAGCG$$$TTATCTGAAA	BCR, ropebwt2, nvSetBWT Merge-BWT, eGSA, eGAP, bwt-lcp-parallel, gsufsort
concBWT	AAGAGGGC$$$TTACTGTAAA	BigBWT, tools for single-string BWT
colexBWT	AAAGGCGG$$$TTACTGTAAA	ropebwt2

$\mathcal{M} = \{\texttt{ATATG}, \texttt{TGA}, \texttt{ACG}, \texttt{ATCA}, \texttt{GGA}\}$

The eBWT

The extended BWT (eBWT) of Mantaci et al. (2007) is a reversible transformation that takes in input a string collection $\mathcal{M}$ and produces a permutation of characters in $\mathcal{M}$.

tools
pfpebwt



The omega order

AATC
ACG
AGG
ATATG
ATCA
ATGAT
ATG
CAAT
CGA
GAC
GAG
GATAT
GAT
GGA
TATGA
TCAA
TGATA
TGA

Let $U^\omega = UUU\ldots$ (infinite concatenation). Then for U, V primitive strings: $U <_\omega V$ if $U^\omega <_{lex} V^\omega$ (can be extended to non-primitive strings).

ATGATATGAT...
ATGATG...

$ATG <_{lex} ATGAT$

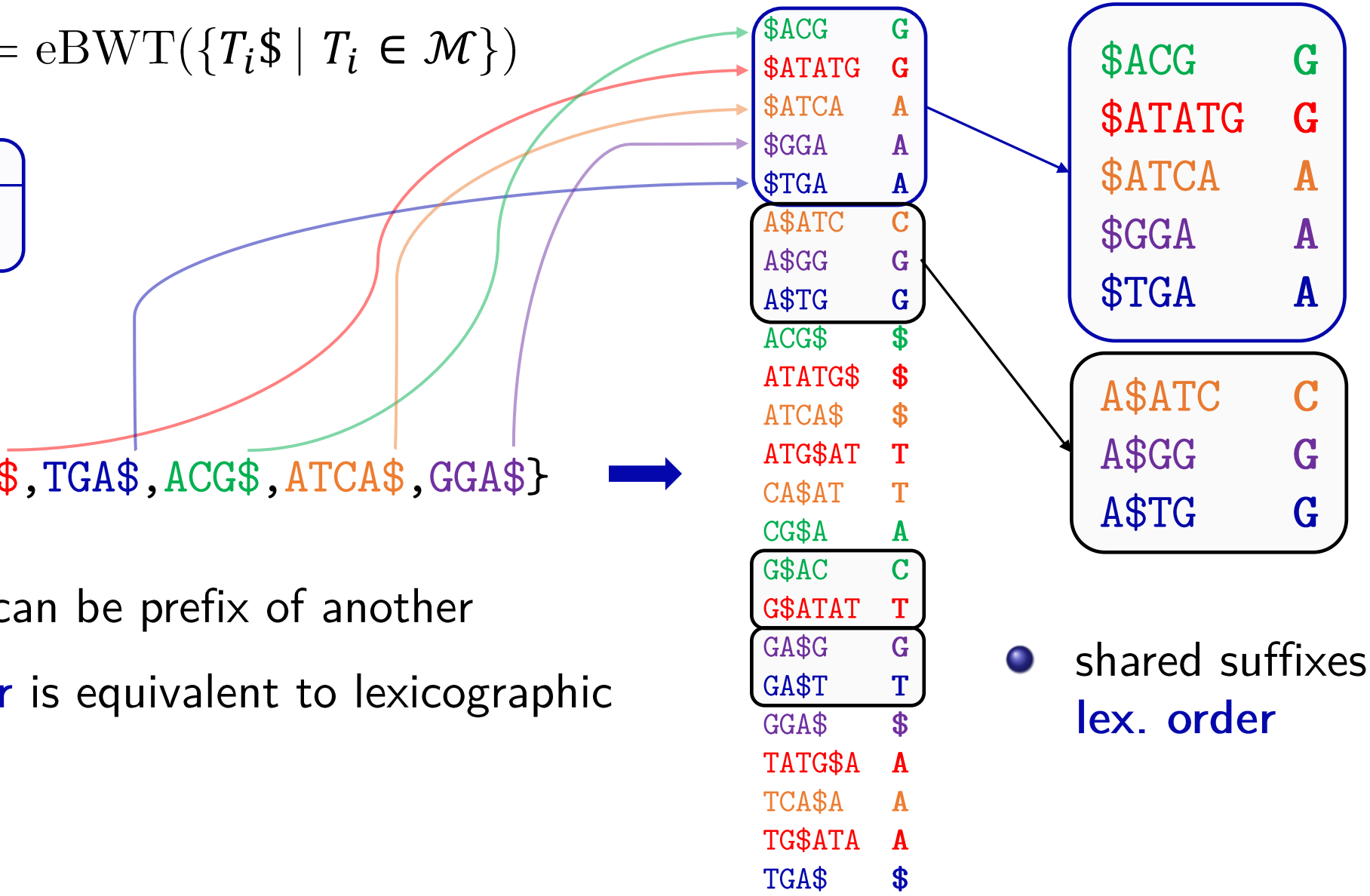
$ATG >_\omega ATGAT$

The dollar-eBWT

$$\text{dolEBWT}(\mathcal{M}) = \text{eBWT}(\{T_i\$ \mid T_i \in \mathcal{M}\})$$

tools
G2BWT, pfpebwt

$$\mathcal{M} = \{\texttt{ATATG\$}, \texttt{TGA\$}, \texttt{ACG\$}, \texttt{ATCA\$}, \texttt{GGA\$}\}$$

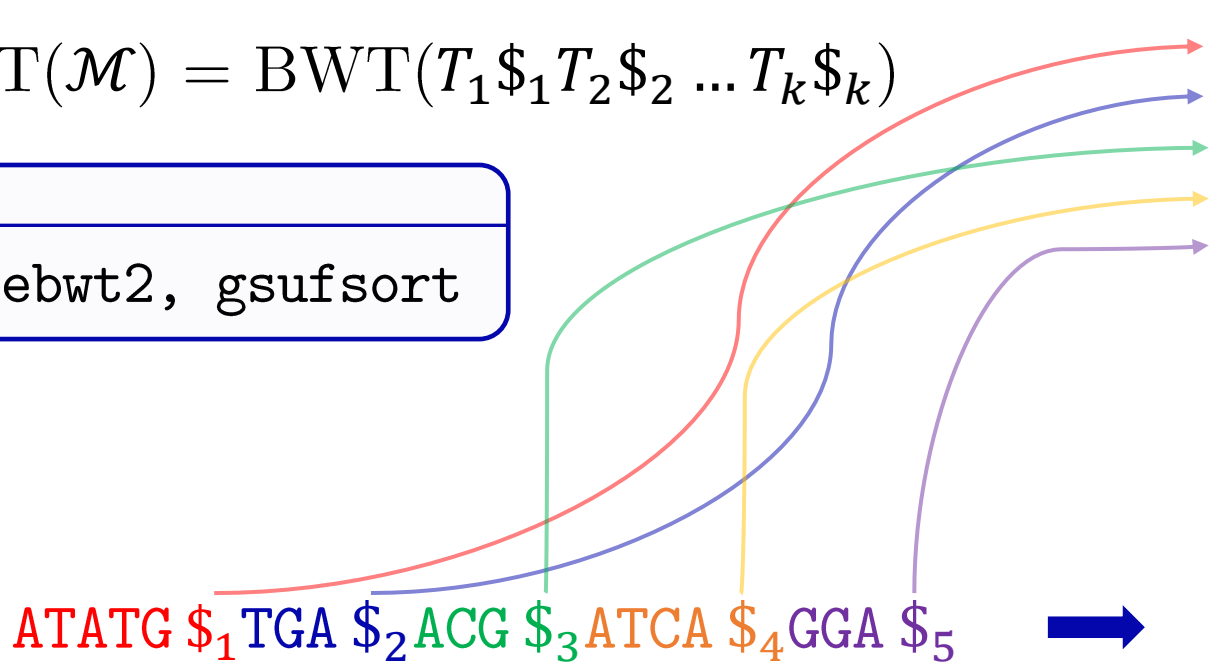


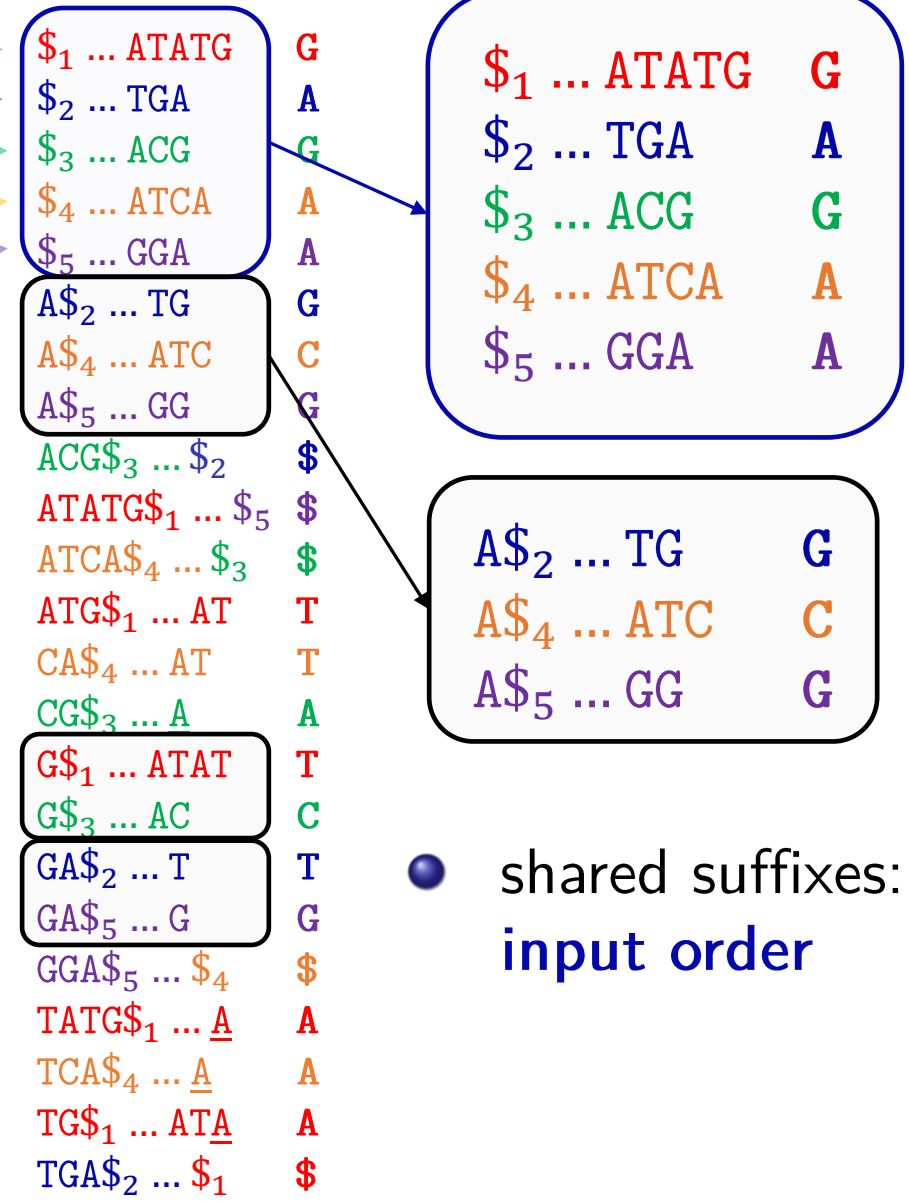
- no rotation can be prefix of another
- **omega order** is equivalent to lexicographic order
- shared suffixes: **lex. order**

The multidollar BWT

$$\text{mdolBWT}(\mathcal{M}) = \text{BWT}(T_1\$_1T_2\$_2 \ldots T_k\$_k)$$

tools
BCR, ropebwt2, gsufsort





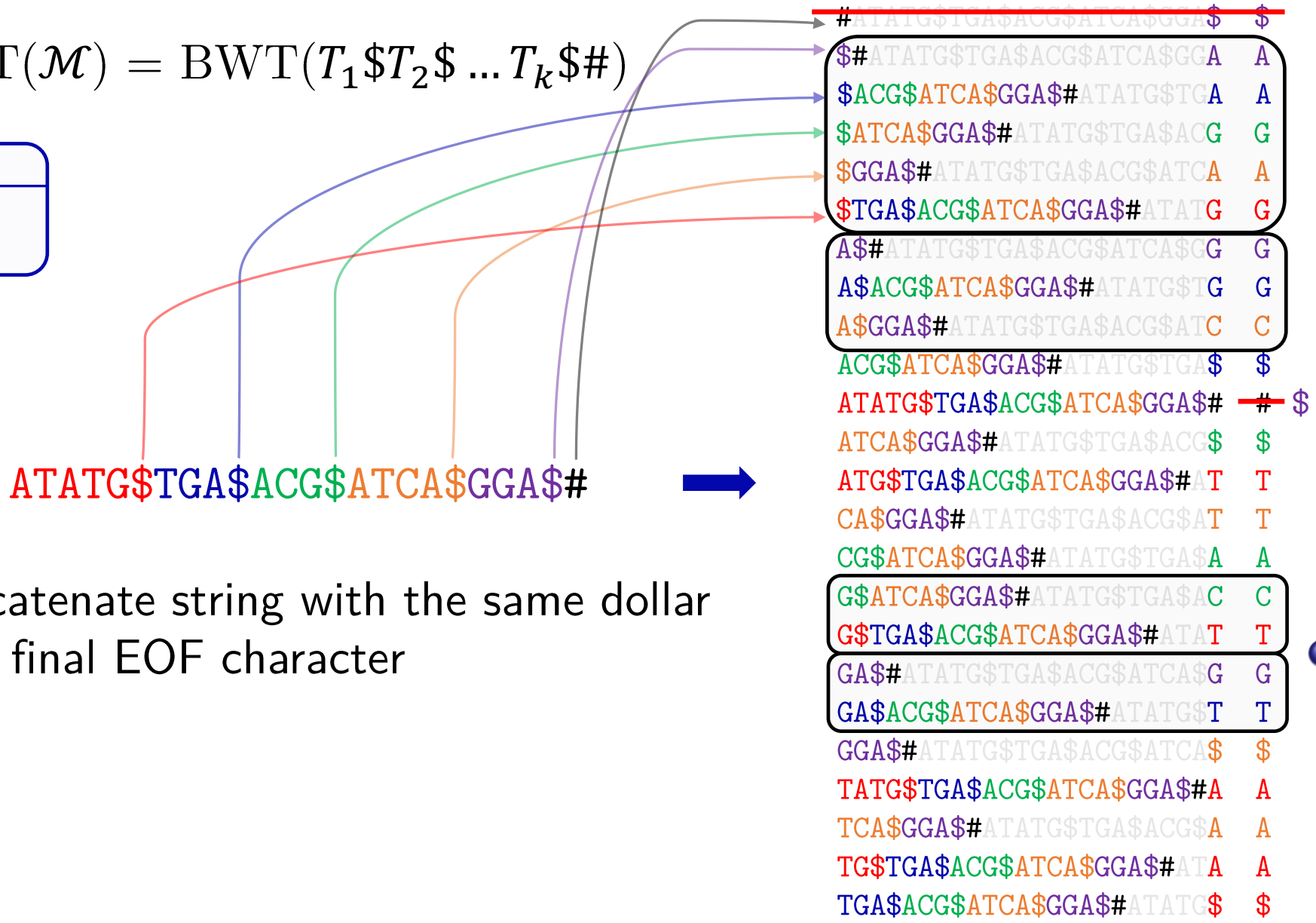
- concatenate strings with **different** dollars (implicitly or explicitly)
- **traditionally** used for generating the suffix tree and suffix array of multiple strings
- shared suffixes: **input order**

The concatenated BWT

$$\mathrm{concBWT}(\mathcal{M}) = \mathrm{BWT}(T_1\$T_2\$\ldots T_k\$\#)$$

tools

BigBWT

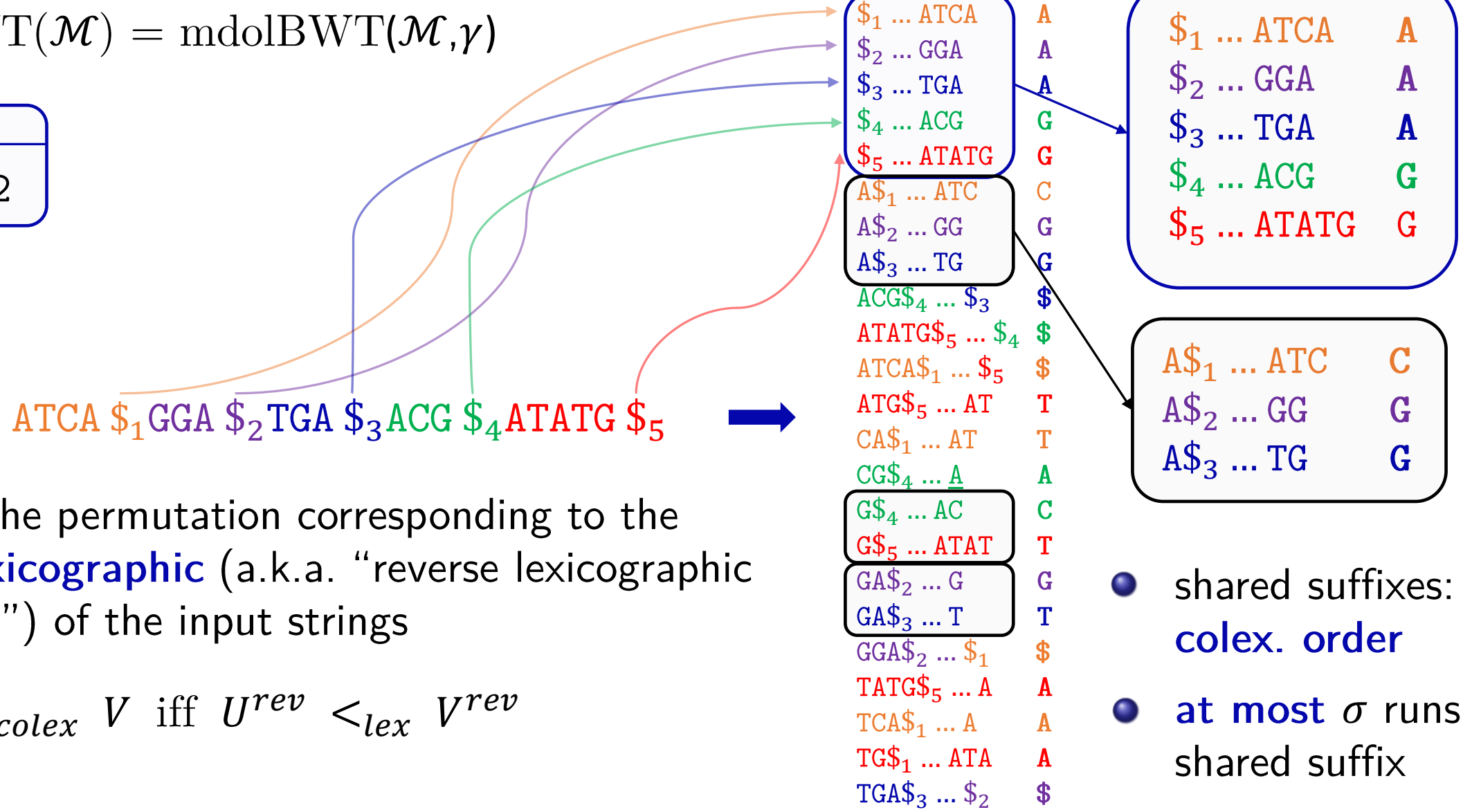


- concatenate string with the same dollar plus final EOF character
- shared suffixes: lex. order of the **next** string

The colexBWT

$$\text{colexBWT}(\mathcal{M}) = \text{mdolBWT}(\mathcal{M},\gamma)$$

tools
ropebwt2



- γ is the permutation corresponding to the **colexicographic** (a.k.a. "reverse lexicographic order") of the input strings

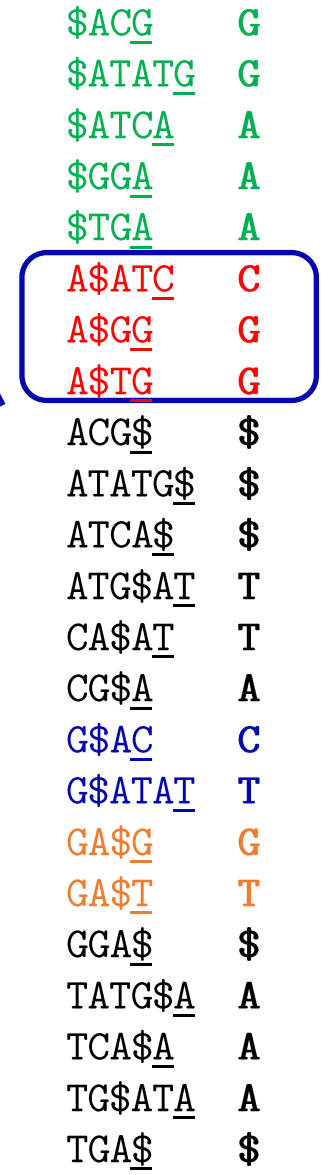
$$U <_{colex} V \text{ iff } U^{rev} <_{lex} V^{rev}$$

- shared suffixes: **colex. order**
- **at most** σ runs per shared suffix

Interesting intervals

We call $[b..e]$ an **interesting interval** if it is the SA-interval of a left-maximal suffix U.

- differences among separator-based BWTs **only in the interesting intervals**



A$#	G
A$...	G
A$...	C

U = A$

concBWT

A$$_2$...TG	G
A$$_4$...ATC	C
A$$_5$...GG	G

U = A$

mdolBWT

A$$_1$ATC	C
A$$_2$GG	G
A$$_3$TG	G

U = A$

dolEBWT

Input order dependence

The mdolBWT and the concBWT are **dependent** on the input order of the strings.

- different **permutations** of the input strings lead to different outputs

$\mathcal{M}_1$ = [ATATG, TGA, ACG, ATCA, GGA] mdolBWT($\mathcal{M}_1$) = GAGAAGCG$$$TTATCTGAAA

$\mathcal{M}_2$ = [ACG, ATATG, GGA, TGA, ATCA] mdolBWT($\mathcal{M}_2$) = GGAAAGGC$$$TTACTGTAAA

$\mathcal{M}_1$ = [ATATG, TGA, ACG, ATCA, GGA] concBWT($\mathcal{M}_1$) = AAGAGGGC$$$TTACTGTAAA

$\mathcal{M}_2$ = [ACG, ATATG, GGA, TGA, ATCA] concBWT($\mathcal{M}_2$) = AGAGACGG$$$TTACTTGAAA

- with concBWT we cannot reach all possible permutations of the BWT characters

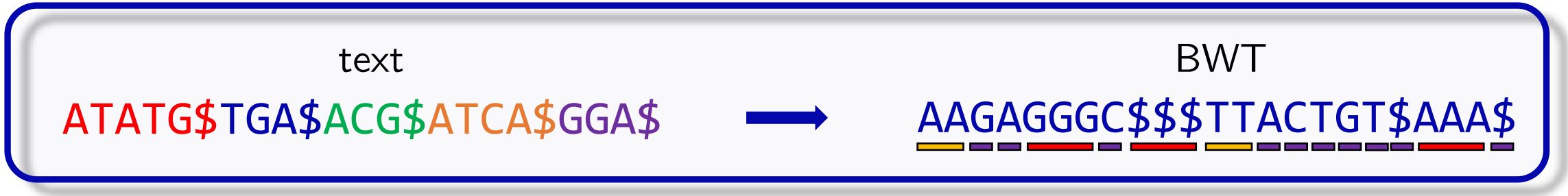
The BWT variants of string collections

variant	result on example	order of shared suffixes	independent of input order?
non-sep. based			
eBWT	CGGGATGTACGTTAAAAA	omega order	yes
separator based			
dolEBWT	GGAAACGG$$$TTACTGTAAA	lexicographic order	yes
mdolBWT	GAGAAGCG$$$TTATCTGAAA	input order	no
concBWT	AAGAGGGC$$$TTACTGTAAA	lex. order of subsequent string	no
colexBWT	AAAGGCGG$$$TTACTGTAAA	colexicographic order	yes

The effect on the r parameter

The number of runs (r) of the BWT

- r(w) = number of single-letter runs of BWT(w)



Increasing interest in the r parameter

- performance of BWT based data structures often measured in terms of r [Gagie et al. SODA 2018]
- measure of repetitiveness

Optimal BWT

- Bentley et al. gave a **linear-time algorithm** for computing the order that minimizes the number of runs

  [Bentley et al. ESA 2020]

- We called this permutation **optBWT** (opt)

- We implemented a variant of this algorithm and compared the different BWT variants to the optBWT

Experimental results

We conducted experiments on 8 real-life datasets with different characteristics.

average runlength

dataset	no. seq	total length	avg	min	max	n/r (opt)
SARS-CoV-2 short	500,000	25,000,000	50	50	50	35.125
Simons Diversity reads	500,000	50,000,000	100	100	100	8.133
16S rRNA short	500,000	75,929,833	152	69	301	44.873
Influenza A reads	500,000	115,692,842	231	60	251	50.275
SARS-CoV-2 long	50,000	53,726,351	1,075	265	3,355	74.498
16S rRNA long	16,741	25,142,323	1,502	1,430	1,549	47.140
Candida auris reads	50,000	124,150,880	2,483	214	8,791	1.732
SARS-CoV-2 genomes	2,000	59,610,692	29,805	22,871	29,920	523.240

Experimental results

We conducted experiments on 8 real-life datasets with different characteristics.

- computed **statistics** for each dataset
- Hamming **distance**
- comparison of the ***r* values** against the optBWT

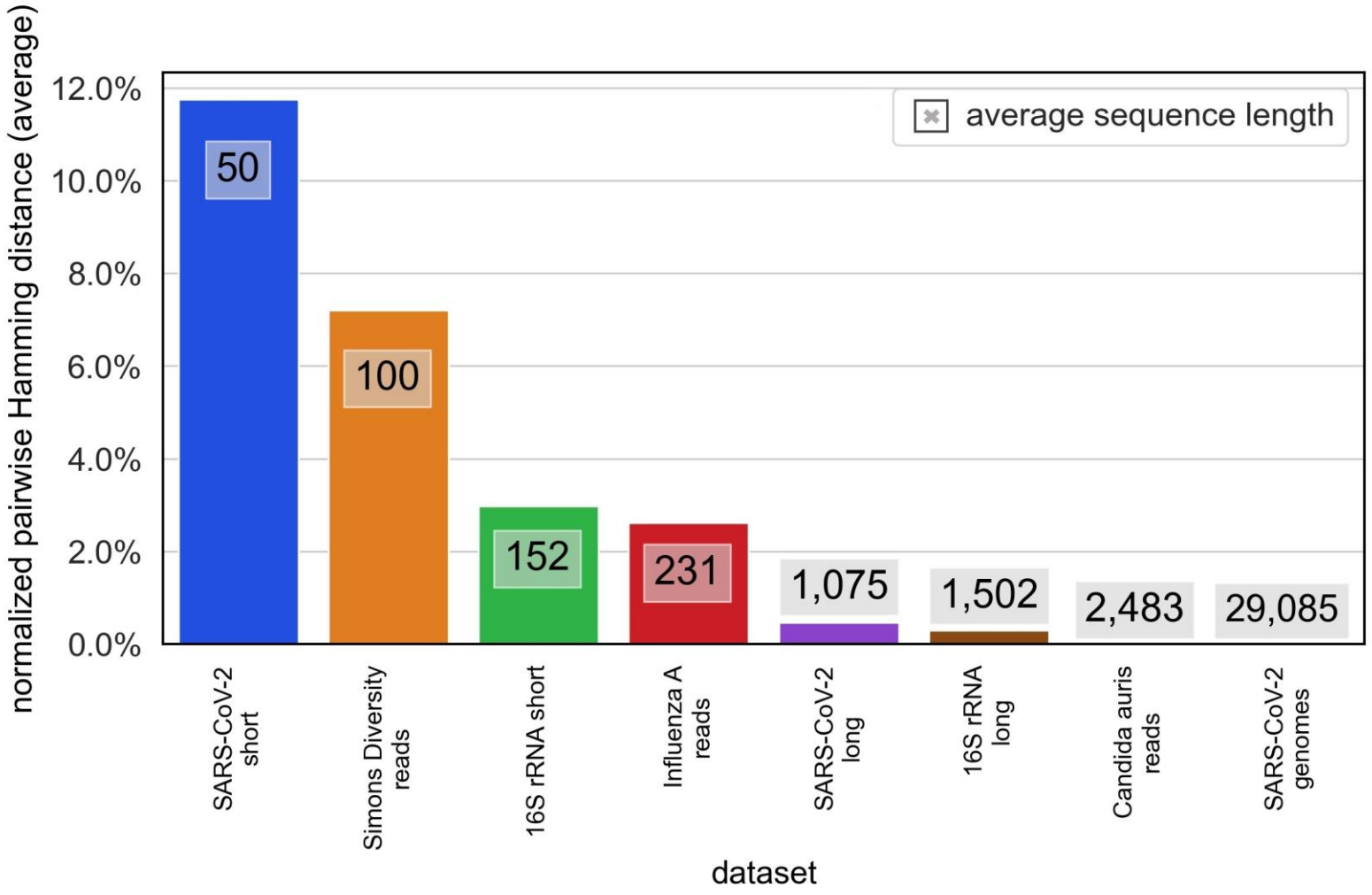
$$var(\mathcal{M}) = \frac{\sum_{[b,e] \text{ interesting interval}} var([b,e])}{\sum_{[b,e] \text{ interesting interval}} (e-b+1)}.$$

dataset	fraction pos.s in interesting intervals	variability	avg. Hamming d. betw. $-sep. BWTs	max n/r (avg. runlength)	min n/r (avg. runlength)
SARS-CoV-2 short	0.792	0.210	0.11754	31.524	7.494
Simons Diversity reads	0.107	0.976	0.07195	7.873	5.299
16S rRNA short	0.741	0.058	0.02982	44.253	18.836
Influenza A reads	0.103	0.363	0.02609	49.172	23.100
SARS-CoV-2 long	0.175	0.037	0.00464	73.204	57.568
16S rRNA long	0.047	0.104	0.00289	46.879	45.015
Candida auris reads	0.007	0.497	0.00246	1.732	1.726
SARS-CoV-2 genomes	0.001	0.148	0.00012	521.610	499.549

Experimental results: Hamming distance

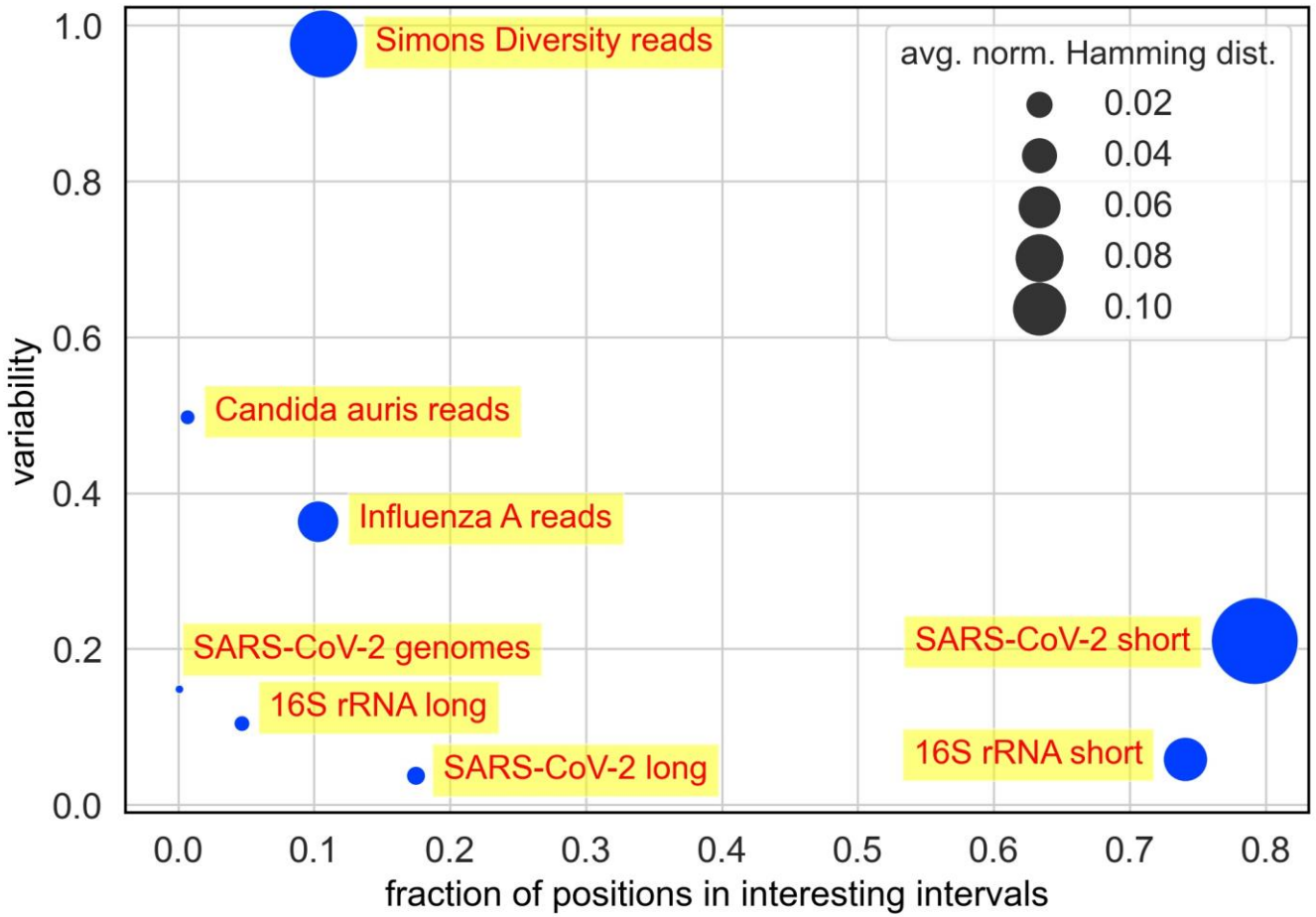
Hamming distance among separator-based BWT variants.

- strongly depends on **sequence length**
- on SARS-CoV-2 short: 500,000 sequences of length 50, on average **almost 12%** different BWT positions



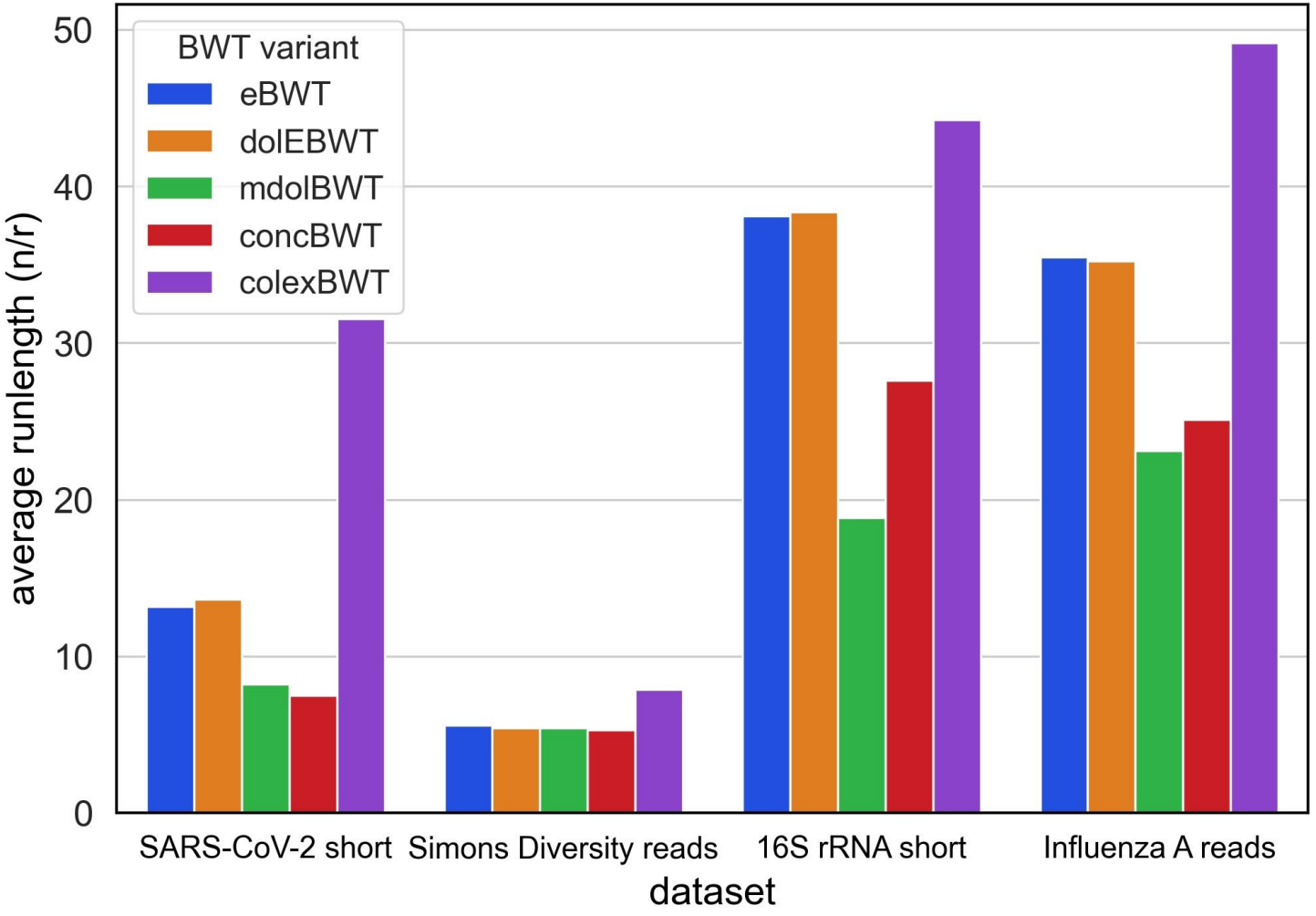
Experimental results: Hamming distance variation

Average Hamming distance variation with respect to the **variability** and **fraction of positions in interesting intervals**.



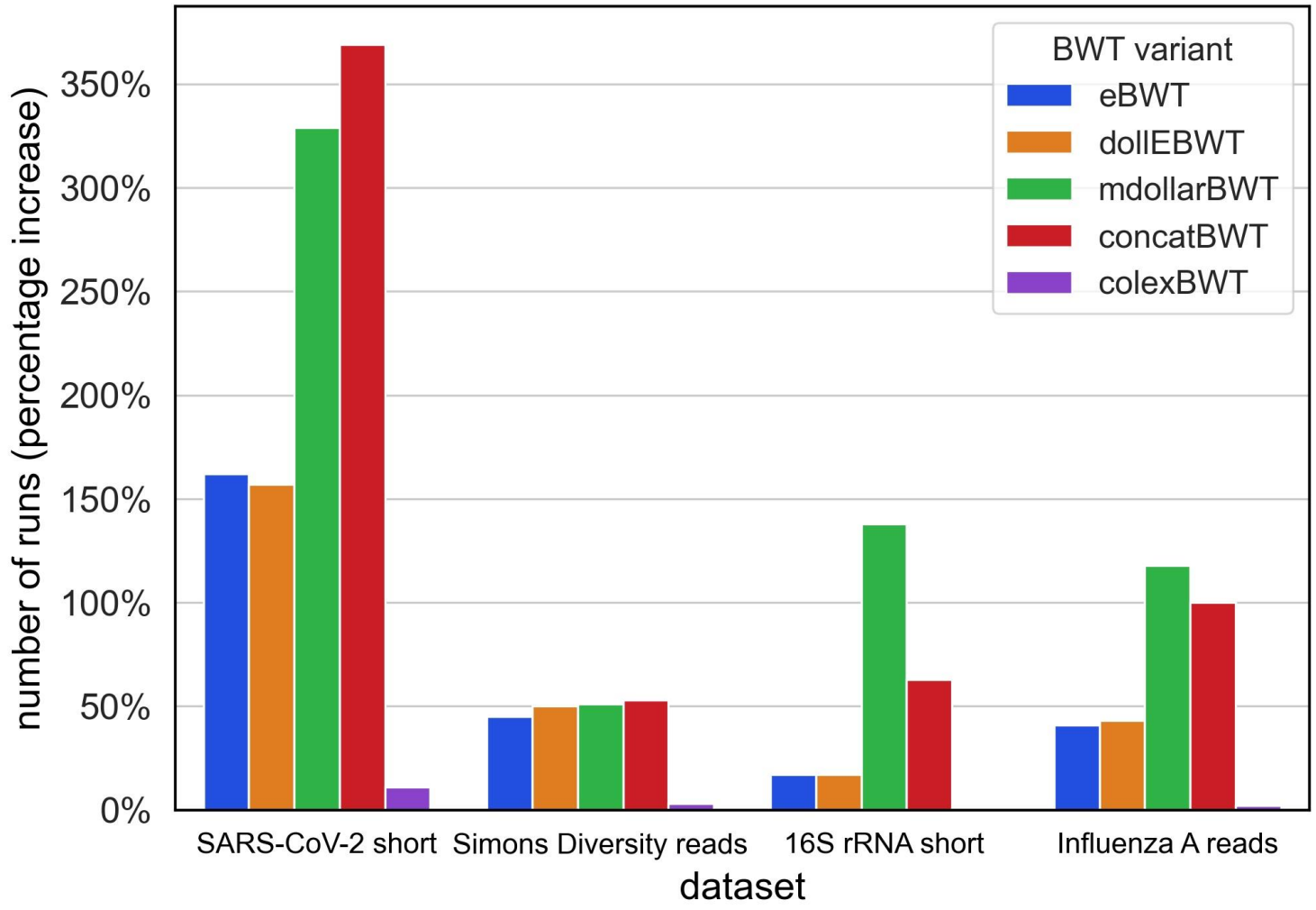
Experimental results: number of runs

Average runlength (n/r) on all BWT variants.



Experimental results: number of runs

Number of runs **percentage increase** with respect to the optBWT.



Experimental results: number of runs

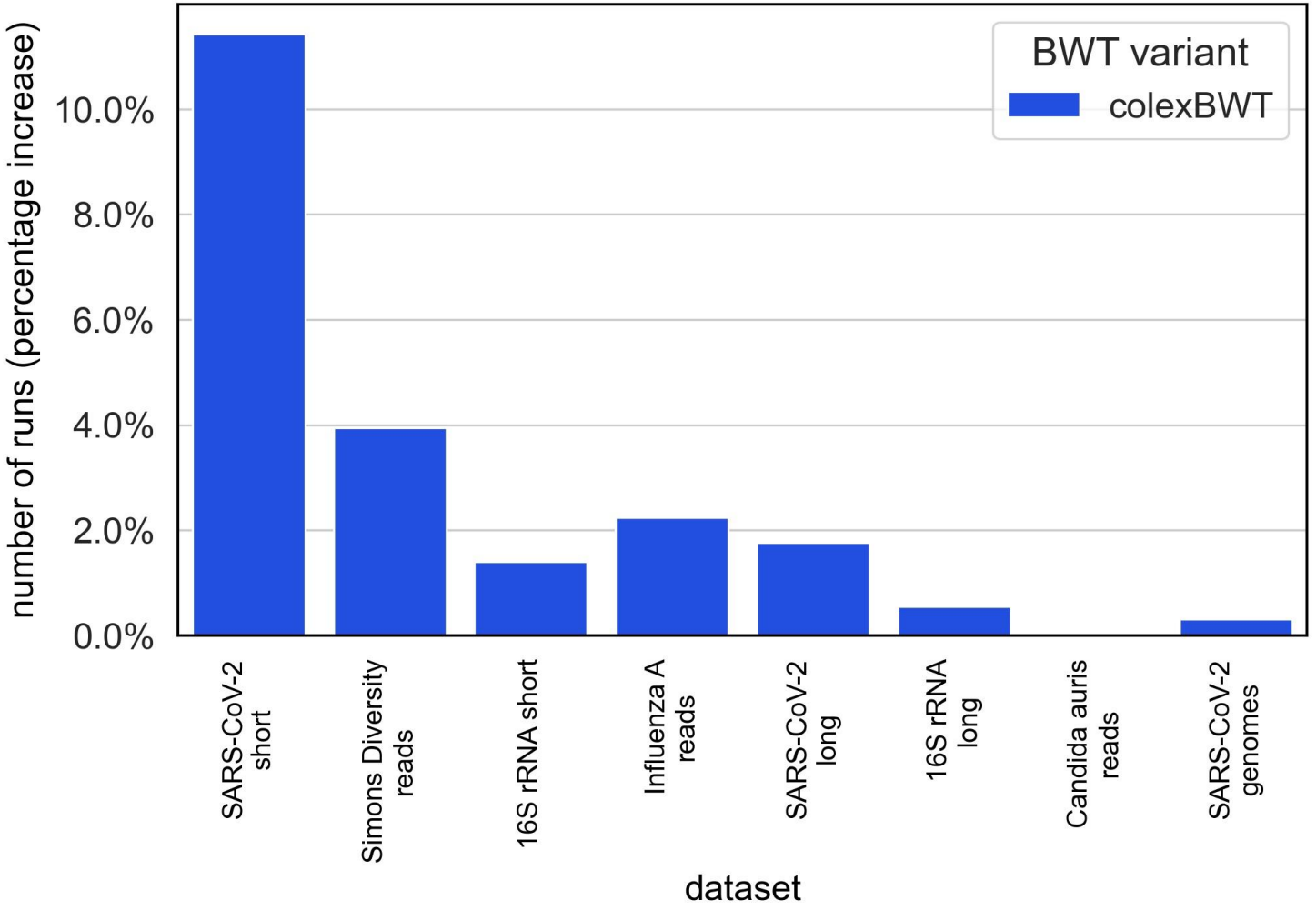
The experiments showed a high variation of the number of runs on datasets containing **short sequences**

- highest variation: **multiplicative factor of over 4.2**
- specific input permutation: for concBWT and mdolBWT, depends on the input order

no. runs **SARSCov2short** dataset		
	no. runs r	avg. runlength n/r
eBWT	1,902,148	13.143
dolEBWT	1,868,581	13.647
mdolBWT	3,113,818	8.189
concBWT	3,402,513	7.494
colexBWT	**808,906**	**31.524**

Experimental results: increase in number of runs of colexBWT w.r.t. optimum

Increase in number of runs of **colexBWT** w.r.t. **optBWT**.



Conclusions

We presented the first systematic study of the variants of the Burrows-Wheeler-Transform of string collections.

- BWT variants **differ significantly** among each other
- several BWT variants in use depend on the **input order**
- differences extend to r
  - parameter for **analyzing** BWT-based data structures
  - measure for dataset **repetitiveness**

We **recommend to standardize** the definition of the parameter r for string collections: colexicographic order, or the optimal order of Bentley et al.

UNIVERSITÀ
di VERONA

Thank you for your attention

contact: davide.cenzato@univr.it

GitHub: https://github.com/davidecenzato/BWT-variants-of-string-collections

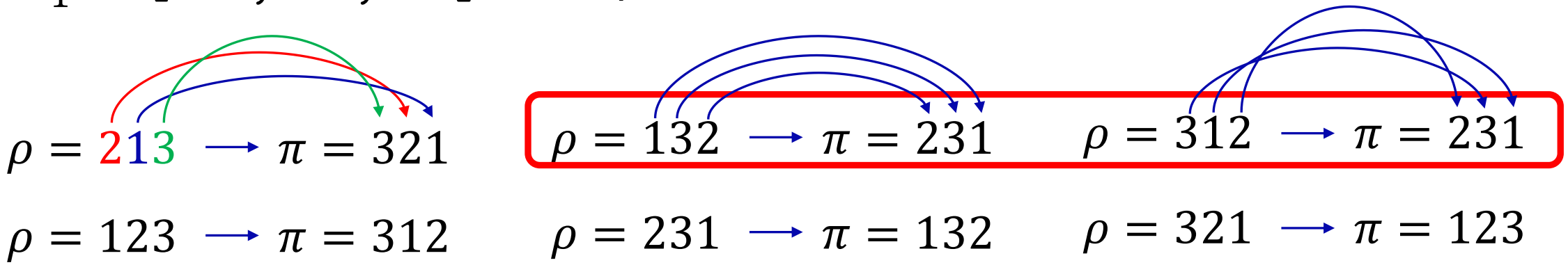
full version: https://arxiv.org/abs/2202.13235

Feasible permutations

With the concBWT we cannot reach all possible permutations of the BWT characters.

$\mathcal{M}_1$ = [GAA, ACA, TGA]

$\rho = 213$



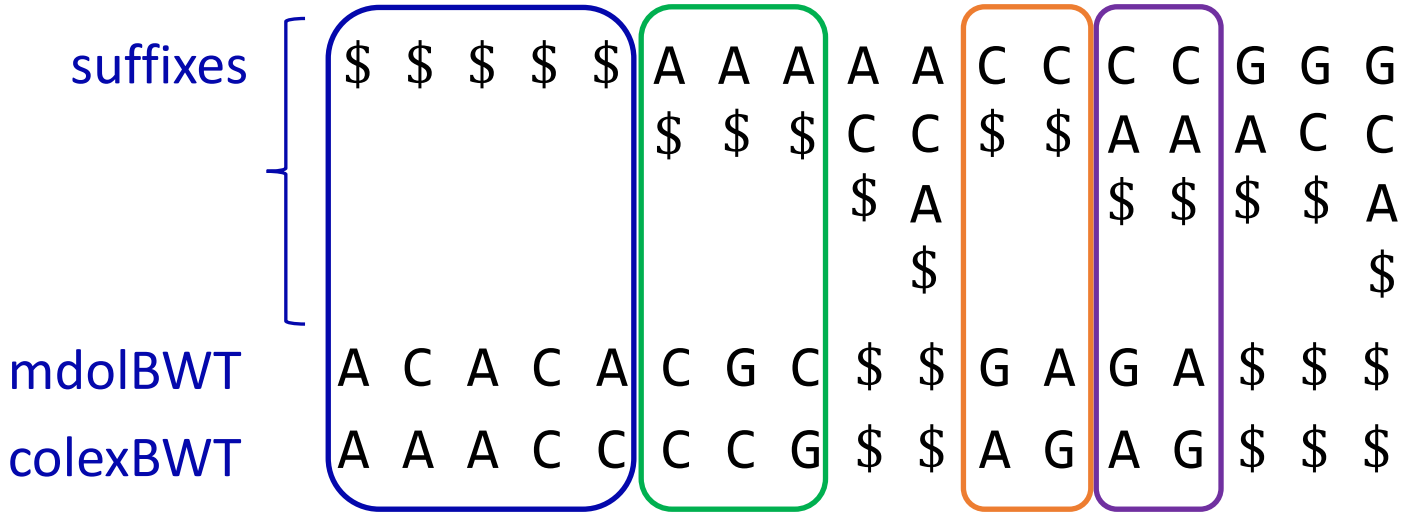
No input permutation maps to $\pi = 213$, so 213 is **not feasible** with concBWT.

We computed the number of feasible permutations for up to 11 strings.

3	4	5	6	7	8	9	10	11
83.33%	75.0%	68.33%	63.89%	60.12%	57.29%	54.8%	52.81%	51.0%

Interesting intervals variability

The number of runs can differ **significantly** between different variants.



$\mathcal{M}$ = [GCA,GC,GA,AC,ACA]

$\mathcal{M}_{colex}$ = [ACA,GCA,GA,CA,GC]

if $n_a - 1 \leq N_a$:
  var = $n_a + N_a$
else:
  var = $2N_a + 1$

N_a = max character frequency
n_a = freq. of the other characters

How much can an interesting interval **vary**?

- depends on the **Parikh vector** of the interval
- the example has maximal variability