An Extension of the Burrows Wheeler Transform and Applications to Sequence Comparison and Data Compression

Sabrina Mantaci Antonio Restivo Giovanna Rosone Marinella Sciortino

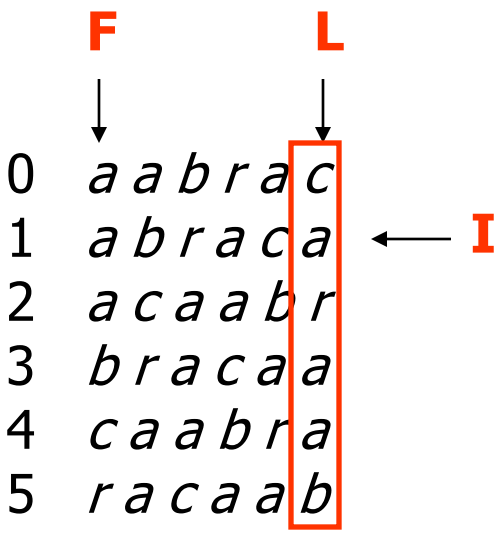
Università di Palermo

Outline

- A short description of the Burrows Wheeler Transform (BWT)
- Extension of the BWT to k words
- Sequences comparison by the extended BWT
- Simultaneous compression of k texts
- Experimental validation
  - of similarity on biological sequences
  - of compression on some files of the Calgary Corpus

How does BWT work?

- INPUT: w = *abraca*
- lexicographically sort all the cyclic shifts or conjugates of w



The following properties hold:

- For all i ≠ I, the character L[i] is followed in w by F[i];
- for any character *ch*, the i-th occurrence of *ch* in F corresponds to the i-th occurrence of *ch* in L.

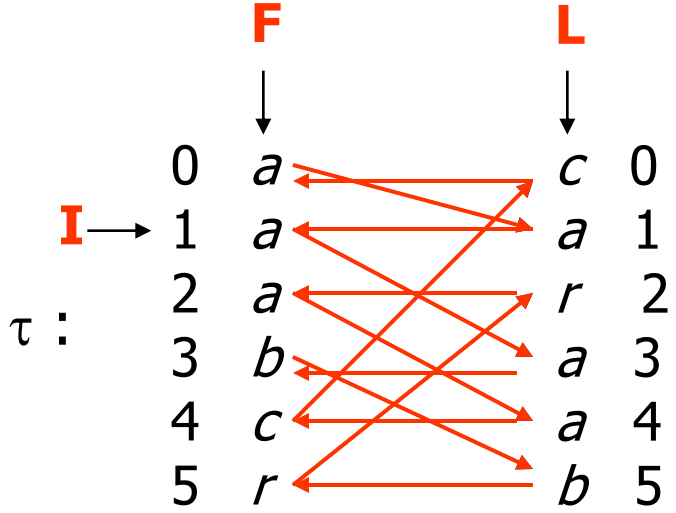
- OUTPUT: BWT(w)=L=*caraab* and the index I=1, that denotes the position of the original word w after the lexicographical sorting of its conjugates.

Reversibility of the BWT

The Burrows-Wheeler Transform is reversible, in the sense that, given BWT(w) and an index I, it is possible to recover the original word w.

Given L=BWT(w)=*caraab* and I=1:

- We construct the first column F, by alphabetically sorting the letters in L.



$$\tau = \begin{pmatrix} 0\ 1\ 2\ 3\ 4\ 5 \\ 1\ 3\ 4\ 5\ 0\ 2 \end{pmatrix} = (1\ 3\ 5\ 2\ 4\ 0)$$

w= *a b r a c a*
1 3 5 2 4 0

- A permutation τ is defined on the set {0,1,…,n-1}, estabilishing a correspondence between the positions of the same letter in F and in L;

- Starting from position I, we can recover w= $F(\tau^0(I))F(\tau^1(I))\ldots F(\tau^{n-1}(I))$ where $\tau^0(x)=x$, $\tau^{i+1}(x)= \tau(\tau^i(x))$

Remarks

If we do not care about the index, the BWT defines a correspondence between the set of conjugacy classes of words over A and the set of words over A. In this sense we have that:

1: The BWT is injective. Every conjugacy class have a different transformation.

2: The Burrows-Wheeler transform is not surjective, in the sense that there exist words in A^* that are not image by the BWT of any conjugacy class. Take for instance *bccaaab*.

3: The permutation defining the BWT is always a cycle with a descent number less than or equal to the size of the alphabet minus one.

The BWT and Combinatorics on words

- Relationship with Standard words.

[S. Mantaci, A. Restivo, M. Sciortino, "Burrows-Wheeler Transform and Sturmian Words", Information Processing Letters, 2003].

- Relationship with combinatorics on permutations.

[M. Crochemore, J. Désarménien, D. Perrin, "A note on the Burrows-Wheeler Transform", Theoretical Computer Science, 2005];
[M. Gessel, C. Reutenauer, "Counting permutations with given cycle structure and descent set", J. Comb. Theory, 1993].

A new order relation

Let A be a finite alphabet and let A^* denote the set of the words over A.
A word u in A^* is primitive if the condition $u=w^n$ implies $u=w$ and $n=1$.

If $u \in A^*$, we denote by $u^\omega=uuuu\ldots$
For every word v, there exists a unique primitive word w and an integer k such that $v=w^k$. By notation, $w=root(v)$ and $k=exp(v)$

Definition: Let u and v be two primitive words.

$$u \leq_\omega v \iff \begin{cases} exp(u)<exp(v) \text{ if } root(u)=root(v) \\ u^\omega <_{lex} v^\omega \text{ otherwise} \end{cases}$$

Where $u^\omega <_{lex} v^\omega$ denotes the usual lexicographic order between infinite words.

A new order relation

Notice that the $<_\omega$ order is different than the lexicographic one. For instance $ab <_{lex} aba$, but $aba \leq_\omega ab$, since $abaababaaba\ldots <_{lex} abababab\ldots$

Proposition: Given two primitive words u and v,

$$u \leq_\omega v \iff pref_k(u^\omega) <_{lex} pref_k(v^\omega)$$

where $k=|u|+|v|-gcd(|u|,|v|)$.

The bound is tight: indeed $abaababa \leq_\omega abaab$ because

$$\overbrace{abaab}\overbrace{abaab}\overbrace{ab}\ldots$$
$$\underbrace{abaababaaba}\ldots$$

differ for the character in position 12=8+5-1.

The extended transform E

INPUT: S={*abac*, *cbab*, *bca*, *cba*}.

- Sort all the conjugates of the words in S by the $\leq_\omega$ order relation;
- Consider the sequence of the sorted words and take the word S′ obtained by concatenating the last letter of each word;
- Take the set I containing the positions of the words corresponding to the ones in S;
- The output of the E transformation is the pair (S′,I).

a b a c a b … →	*1*	*a b a c*
a b c a b c …	*2*	*a b c*
a b c b a b …	*3*	*a b c b*
a c a b a c …	*4*	*a c a b*
a c b a c b …	*5*	*a c b*
b a b c b a …	*6*	*b a b c*
b a c a b a …	*7*	*b a c a*
b a c b a c …	*8*	*b a c*
b c a b c a … →	*9*	*b c a*
b c b a b c …	*10*	*b c b a*
c a b a c a …	*11*	*c a b a*
c a b c a b …	*12*	*c a b*
c b a b c b … →	*13*	*c b a b*
c b a c b a … →	*14*	*c b a*

OUTPUT: (*ccbbbcacaaabba*, {1,9,13,14})

Properties of the E transform

INPUT: S={*abac, cbab, bca, cba*}.

→	*1*	*a b a c*
	2	*a b c*
	3	*a b c b*
	4	*a c a b*
	5	*a c b*
	6	*b a b c*
	7	*b a c a*
	8	*b a c*
→	*9*	*b c a*
	10	*b c b a*
	11	*c a b a*
	12	*c a b*
→	*13*	*c b a b*
→	*14*	*c b a*

- In any row $i \notin I$, the first symbol follows the last one, in a word in S.
- For each character, the i-th occurrence in the first column corresponds to the i-th occurrence in the red column.

The inverse transformation

Given E(S)=(ccbbbcacaaabba, {1,9,13,14}), the following permutation is defined:

F=	a	a	a	a	a	b	b	b	b	b	c	c	c	c
	1	2	3	4	5	6	7	8	9	10	11	12	13	14
	7	9	10	11	14	3	4	5	12	13	1	2	6	8
L=	c	c	b	b	b	c	a	c	a	a	a	b	b	a

Consider the cyclic decomposition of the permutation and read the corresponding letters in F:

(1	7	4	11)	(2	9	12)	(3	10	13	6)	(5	14	8)
a	b	a	c	a	b	c	a	b	c	b	a	c	b

Bijectivity

Let M be the family of multisets of conjugacy classes of primitive words of A^*. Then, if we don't care about the indices

$$E: M \longrightarrow A^*$$

Theorem: The transformation E is injective.

Theorem: For each word $u \in A^*$, there exists a multiset $S \in M$ such that E(S)=u.

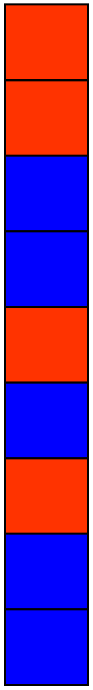
For instance, E(*ab*, *abcac*)=(*bccaaab*).

Theorem [Gessel-Reutenauer]: There exists a bijection between A^* and the family of multisets of conjugacy classes of primitive words in A^*.

A distance measure between words

u=bcaa *v=ccbab*

aabc
abca
abccb
babcc
bcaa
bccba
caab
cbabc
ccbab



u
u
v
v
u
v
u
v
v

$\gamma(u,v)=u^2v^2uvuv^2$

$\delta(u,v)=3$

In general:

$\gamma(u,v)=u^{n_1}v^{n_2}u^{n_3}\ldots v^{n_k}$

Definition

$$\delta(u,v)=\sum_{\substack{i=1,\\ n_i\neq 0}}^{k}(n_i-1)$$

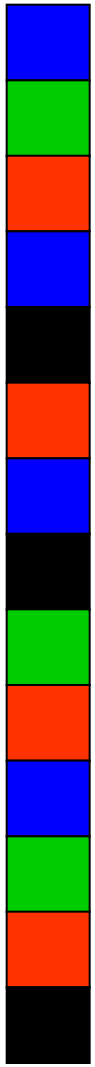
Properties of the distance measure δ

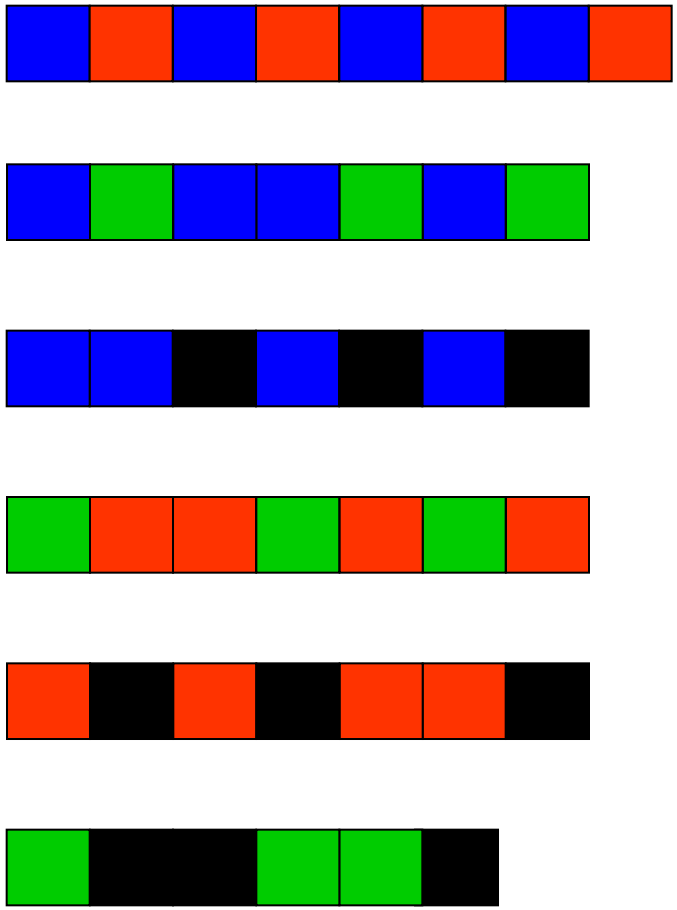
- $\delta(u,v)=\delta(v,u)$, that is the distance measure δ is symmetric.
- If u and v are conjugates, then $\delta(u,v)=0$. In fact in this case $\gamma(u,v)=(uv)^{|u|}$, then $n_i=1$ for all $i=1,\ldots,2|u|$.
- If u' is a conjugate of u and v' is a conjugate of v, then $\delta(u,v)= \delta(u',v')$. Then, δ is a similarity measure between conjugacy classes.
- $\delta(u,v)=0$ does not imply that u and v are conjugate. For instance, if u=*aabc* and v=*abbc*, $\gamma(u,v)=(uv)^4$. Then $\delta(u,v)=0$.
- δ does not satisfy the triangle inequality. For instance if u=abaab, v=babab, z=abbba, we have $\delta(u,v)=6$, $\delta(v,z)=3$ and $\delta(u,z)=2$.

Multiple sequence comparison

INPUT: S={u=abac, v=cbab, w=bca, z=cba}.

a b a c
a b c
a b c b
a c a b
a c b
b a b c
b a c a
b a c
b c a
b c b a
c a b a
c a b
c b a b
c b a





$\delta(u,v)=0$

$\delta(u,w)=1$

$\delta(u,z)=1$

$\delta(v,w)=1$

$\delta(u,z)=1$

$\delta(w,z)=1$

Multiple sequence comparison

- We can compute the distance δ of all pairs taken out from a set S of k sequences of length n by symultaneously applying the transformation E to the entire set S.
- In order to obtain the k×k-matrix of distances we can perform a single sorting of kn sequences of length n instead of k^2 sortings of 2n sequences of length n.

- We can define the notion of distance between sets

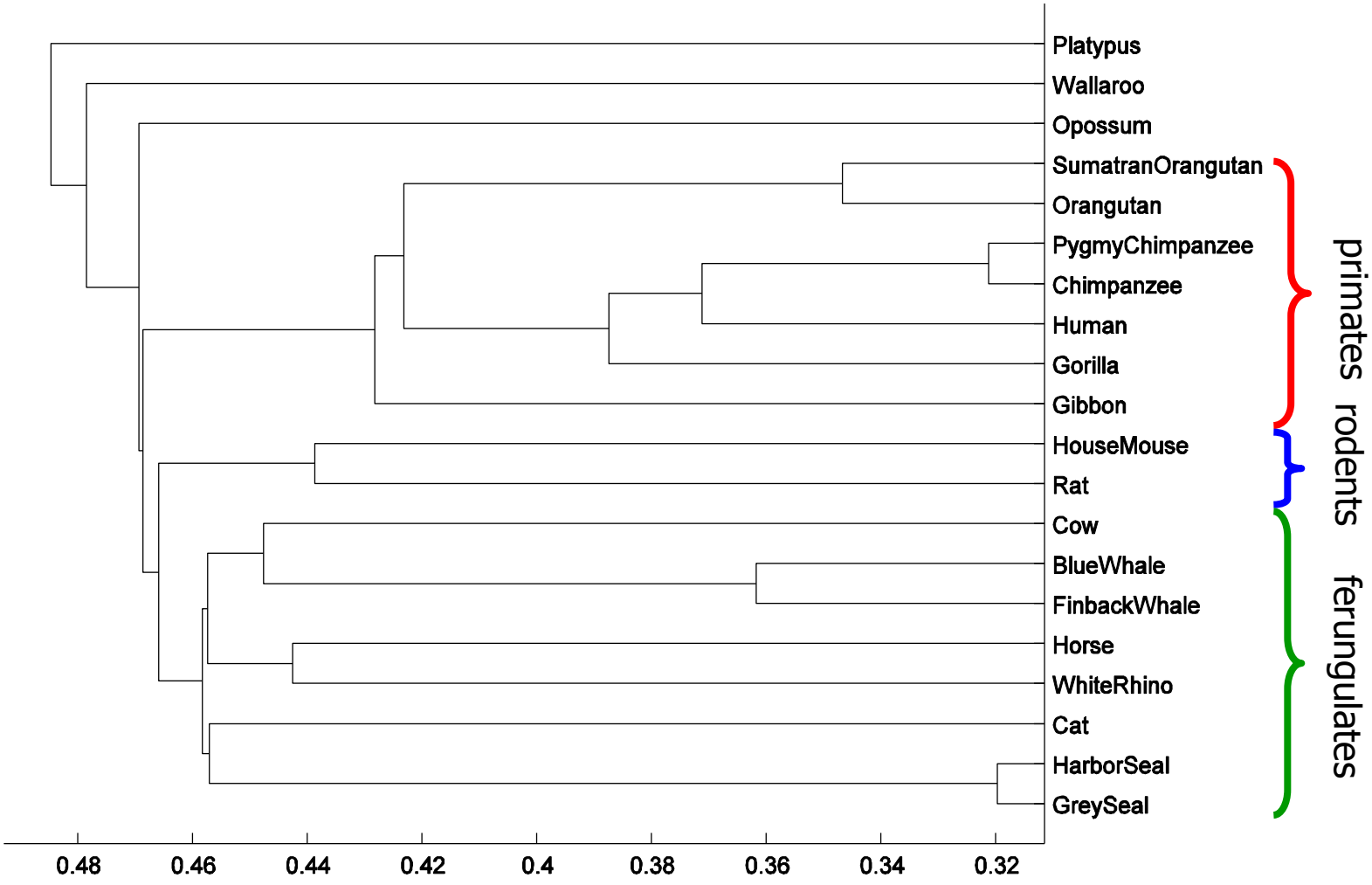
  If S and T are two sets, we encode each conjugate of an element of S and T by U and V, respectively. If $\gamma(S,T)=U^{n_1}V^{n_2}U^{n_3}\ldots V^{n_k}$ is the sequence so obtained, we define

$$\delta(S,T)=\sum_{\substack{i=1,\\ n_i\neq 0}}^{k}(n_i-1)$$

Comparison of biological sequences

- δ is an example of alignment free distance measure.
- δ measures how dissimilar two conjugacy classes of sequences are.
- In order to test our method we applied the normalized version of our distance to the whole mitochondrial genome phylogeny.
- The results we have obtained are very close to the ones derived, with other approaches in most of the papers in which the considered species are the same.

Evolutionary tree built from mtDNA sequences of 20 species



Simultaneous compression of k texts

- The BWT was introduced as a tool in order to get a word easier to compress.
- We use the transformation E as a preprocessing for the simultaneous compression of a set of k texts.
- If $\{x_1,x_2,\ldots,x_k\}$ is a multiset of words, we denote by $C(x_1,x_2,\ldots,x_k)$ the word obtained by applying a compressor C to the output of $E(\{x_1,x_2,\ldots x_k\})$.
- If π is a permutation on $\{1,\ldots,k\}$, then

$$C(x_1,\ldots,x_k)=C(x_{\pi(1)},x_{\pi(2)},\ldots,x_{\pi(k)})$$

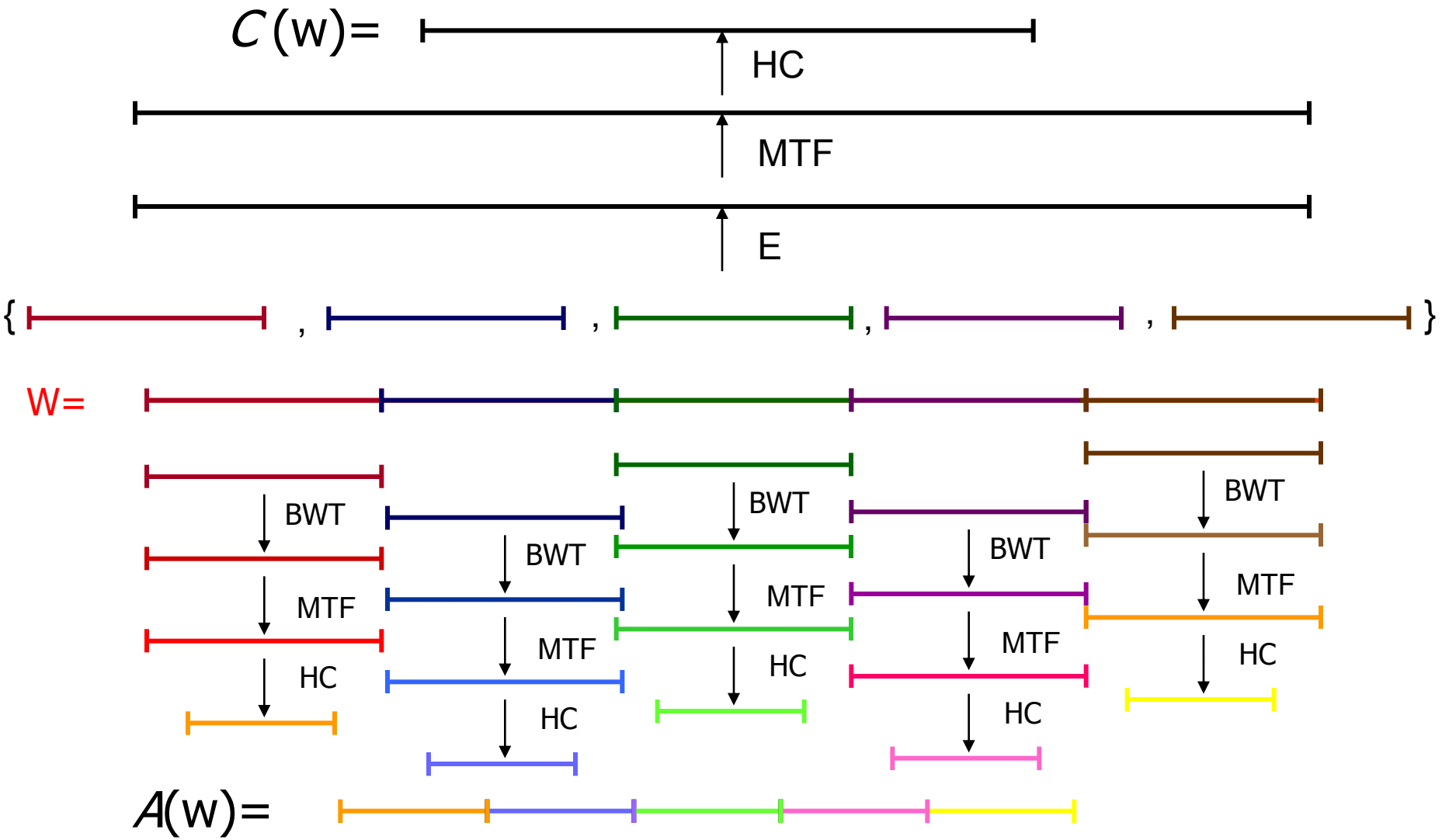
Simultaneous compression of k texts

- In most of practical cases, if X and Y are two multisets of words, then

$$|C(XUY)|<|C(X)|+|C(Y)|$$

This means that the simultaneous compression of $\{x_1,x_2,\ldots x_k\}$ by using E as preprocessing is better than compressing each word x_i separately and concatenating the outputs of the compressed words.

An E-based compressor



Simultaneous compression of k texts

A is a BWT-based compressor (BWT+MTF+HC).
C is the E-based compressor.
M and N are the size of the blocks and the whole text, respectively.
We compare the compression ratios (expressed as output bits per input character) obtained with some files of the Calgary Corpus.

File	Size(in bytes)	Alg	M=16K	M=64K	M=N
bib	111261	*C*	2.547	2.461	2.425
		A	3.204	2.634	
obj1	21504	*C*	4.743		4.740
		A	5.076		
paper2	82199	*C*	2.805	2.786	2.779
		A	3.330	2.917	
progl	71646	*C*	2.145	2.138	2.131
		A	2.440	2.200	
trans	93695	*C*	2.064	1.978	1.950
		A	2.667	2.123	