Burrows-Wheeler Transform and Balanced words

Antonio Restivo and Giovanna Rosone

University of Palermo, Dipartimento di Matematica ed Applicazioni,
Via Archirafi 34, 90123 Palermo, ITALY

restivo@math.unipa.it

WORDS 2009, 14-18 September

Outline

- In 1994 M. Burrows and D. Wheeler introduced a new data compression method based on a preprocessing of the input string. Such a preprocessing is called Burrows-Wheeler Transform (BWT).
- The application of the BWT produces a clustering effect (occurrences of a given symbol tend to occur in clusters).
- We investigate the clustering effect of BWT and its relation with compression performances.
- In such an investigation we consider notions and introduce techniques that are relevant for combinatorics on words.

Outline

- In 1994 M. Burrows and D. Wheeler introduced a new data compression method based on a preprocessing of the input string. Such a preprocessing is called Burrows-Wheeler Transform (BWT).
- The application of the BWT produces a clustering effect (occurrences of a given symbol tend to occur in clusters).
- We investigate the clustering effect of BWT and its relation with compression performances.
- In such an investigation we consider notions and introduce techniques that are relevant for combinatorics on words.

Outline

- In 1994 M. Burrows and D. Wheeler introduced a new data compression method based on a preprocessing of the input string. Such a preprocessing is called Burrows-Wheeler Transform (BWT).
- The application of the BWT produces a clustering effect (occurrences of a given symbol tend to occur in clusters).
- We investigate the clustering effect of BWT and its relation with compression performances.
- In such an investigation we consider notions and introduce techniques that are relevant for combinatorics on words.

Outline

- In 1994 M. Burrows and D. Wheeler introduced a new data compression method based on a preprocessing of the input string. Such a preprocessing is called Burrows-Wheeler Transform (BWT).
- The application of the BWT produces a clustering effect (occurrences of a given symbol tend to occur in clusters).
- We investigate the clustering effect of BWT and its relation with compression performances.
- In such an investigation we consider notions and introduce techniques that are relevant for combinatorics on words.

How does BWT work?

- BWT takes as input a text v and produces:
  - a permutation $bwt(v)$ of the letters of v.
  - the index I, that is useful in order to recover the original word v.
- Example: $v = abraca$

M

- Each row of M is a conjugate of v in lexicographic order.
- $bwt(v)$ coincides with the last column L of the BW-matrix M.
- The index I is the row of M containing the original sequence.

		F $\downarrow$					L $\downarrow$
	1	a	a	b	r	a	c
$I \rightarrow$	2	a	b	r	a	c	a
	3	a	c	a	a	b	r
	4	b	r	a	c	a	a
	5	c	a	a	b	r	a
	6	r	a	c	a	a	b

- Notice that if we except the index, all the mutual conjugate words have the same Burrows-Wheeler Transform.
- Hence, the BWT can be thought as a transformation acting on circular words.

How does BWT work?

- BWT takes as input a text v and produces:
  - a permutation $bwt(v)$ of the letters of v.
  - the index I, that is useful in order to recover the original word v.
- Example: $v = abraca$

- Each row of M is a conjugate of v in lexicographic order.
- $bwt(v)$ coincides with the last column L of the BW-matrix M.
- The index I is the row of M containing the original sequence.

$$
\begin{array}{cc|cccccc}
 & & & & M & & & \\
 & & F & & & & & L \\
 & & \downarrow & & & & & \downarrow \\
 & 1 & a & a & b & r & a & c \\
I \rightarrow & 2 & a & b & r & a & c & a \\
 & 3 & a & c & a & a & b & r \\
 & 4 & b & r & a & c & a & a \\
 & 5 & c & a & a & b & r & a \\
 & 6 & r & a & c & a & a & b
\end{array}
$$

- Notice that if we except the index, all the mutual conjugate words have the same Burrows-Wheeler Transform.
- Hence, the BWT can be thought as a transformation acting on circular words.

How does BWT work?

- BWT takes as input a text v and produces:
  - a permutation $bwt(v)$ of the letters of v.
  - the index I, that is useful in order to recover the original word v.
- Example: $v = abraca$

- Each row of M is a conjugate of v in lexicographic order.
- $bwt(v)$ coincides with the last column L of the BW-matrix M.
- The index I is the row of M containing the original sequence.

$$M$$

$$\begin{array}{rc|ccccc|c}
 & & F & & & & & L \\
 & & \downarrow & & & & & \downarrow \\
 & 1 & a & a & b & r & a & c \\
I \rightarrow & 2 & a & b & r & a & c & a \\
 & 3 & a & c & a & a & b & r \\
 & 4 & b & r & a & c & a & a \\
 & 5 & c & a & a & b & r & a \\
 & 6 & r & a & c & a & a & b
\end{array}$$

- Notice that if we except the index, all the mutual conjugate words have the same Burrows-Wheeler Transform.
- Hence, the BWT can be thought as a transformation acting on circular words.

How does BWT work?

- BWT takes as input a text v and produces:
  - a permutation $bwt(v)$ of the letters of v.
  - the index I, that is useful in order to recover the original word v.
- Example: $v = abraca$

- Each row of M is a conjugate of v in lexicographic order.
- $bwt(v)$ coincides with the last column L of the BW-matrix M.
- The index I is the row of M containing the original sequence.

$$
\begin{array}{cc|cccccc}
 & & & & M & & & \\
 & & F & & & & & L \\
 & & \downarrow & & & & & \downarrow \\
 & 1 & a & a & b & r & a & c \\
I \rightarrow & 2 & a & b & r & a & c & a \\
 & 3 & a & c & a & a & b & r \\
 & 4 & b & r & a & c & a & a \\
 & 5 & c & a & a & b & r & a \\
 & 6 & r & a & c & a & a & b
\end{array}
$$

- Notice that if we except the index, all the mutual conjugate words have the same Burrows-Wheeler Transform.
- Hence, the BWT can be thought as a transformation acting on circular words.

Why Useful?

INTUITION

Let us consider the effect of BWT on an English text:

$$v = She \dots the \dots The \dots He \dots the \dots the \dots the \dots she \dots the \dots$$

$$\begin{array}{ccc}
F & & L \\
\downarrow & & \downarrow \\
he & \cdots & t \\
he & \cdots & s \\
he & \cdots & t \\
he & \cdots & t \\
\vdots & & \vdots \\
he & \cdots & t \\
he & \cdots & T \\
He & \cdots & \\
he & \cdots & S
\end{array}$$

The characters preceding *he* are grouped together inside $bwt(v)$.

Extensive experimental work confirms this "clustering effect" (M. Burrows and D. Wheeler,1994, P. Fenwick, 1996).

Empirical Entropy - Intuition

- $H_0(v)$: Maximum compression we can get without context information where a fixed codeword is assigned to each alphabet character (e.g.: Huffman code)
- $H_k(v)$: Lower bound for compression with order-k contexts: the codeword representing each symbol depends on the k symbols preceding it
- Traditionally, compression ratio of BWT-based compression algorithms are usually measured by using $H_k(s)$.
  Manzini, 2001,
  Ferragina, Giancarlo, Manzini, Sciortino, 2005

Empirical Entropy - Intuition

- $H_0(v)$: Maximum compression we can get without context information where a fixed codeword is assigned to each alphabet character (e.g.: Huffman code)
- $H_k(v)$: Lower bound for compression with order-k contexts: the codeword representing each symbol depends on the k symbols preceding it
- Traditionally, compression ratio of BWT-based compression algorithms are usually measured by using $H_k(s)$.
  Manzini, 2001,
  Ferragina, Giancarlo, Manzini, Sciortino, 2005

Empirical Entropy - Intuition

- $H_0(v)$: Maximum compression we can get without context information where a fixed codeword is assigned to each alphabet character (e.g.: Huffman code)
- $H_k(v)$: Lower bound for compression with order-k contexts: the codeword representing each symbol depends on the k symbols preceding it
- Traditionally, compression ratio of BWT-based compression algorithms are usually measured by using $H_k(s)$.
  Manzini, 2001,
  Ferragina, Giancarlo, Manzini, Sciortino, 2005

Burrows-Wheeler Transform

$$v \longrightarrow \boxed{\text{BWT}} \longrightarrow \text{bwt}(v)$$

They question the effectiveness of $H_k(v)$.
Is there a more appropriate statistic?

Our intuition:
the more balanced the input sequences is
the more local similarity we have after BWT.

H. Kaplan, S. Landau and E. Verbin, 2007. The more local similarity is found in the BWT of the string, the better the compression is.

Local Entropy: a statistic that measures local similarity.

They get an upper bound of compression ratio in terms of Local Entropy of $bwt(v)$.

Burrows-Wheeler Transform

$$v \longrightarrow \boxed{\text{BWT}} \longrightarrow \text{bwt}(v)$$

They question the effectiveness of $H_k(v)$.
Is there a more appropriate statistic?

Our intuition:
the more balanced the input sequences is
the more local similarity we have after BWT.

H. Kaplan, S. Landau and E. Verbin, 2007. The more local similarity is found in the BWT of the string, the better the compression is.

Local Entropy: a statistic that measures local similarity.

They get an upper bound of compression ratio in terms of Local Entropy of $bwt(v)$.

Burrows-Wheeler Transform

v $\longrightarrow$ $\boxed{\text{BWT}}$ $\longrightarrow$ bwt(v)

They question the effectiveness of $H_k(v)$.
Is there a more appropriate statistic?

Our intuition:
the more balanced the input sequences is
the more local similarity we have after BWT.

H. Kaplan, S. Landau and E. Verbin, 2007. The more local similarity is found in the BWT of the string, the better the compression is.

Local Entropy: a statistic that measures local similarity.

They get an upper bound of compression ratio in terms of Local Entropy of $bwt(v)$.

Burrows-Wheeler Transform

$$v \longrightarrow \boxed{\text{BWT}} \longrightarrow \text{bwt}(v)$$

They question the effectiveness of $H_k(v)$.
Is there a more appropriate statistic?

Our intuition:
the more balanced the input sequences is
the more local similarity we have after BWT.

H. Kaplan, S. Landau and E. Verbin, 2007. The more local similarity is found in the BWT of the string, the better the compression is.

Local Entropy: a statistic that measures local similarity.

They get an upper bound of compression ratio in terms of Local Entropy of $bwt(v)$.

Burrows-Wheeler Transform

$$\text{v} \longrightarrow \boxed{\text{BWT}} \longrightarrow \text{bwt(v)}$$

They question the effectiveness of $H_k(v)$.
Is there a more appropriate statistic?

Our intuition:
the more balanced the input sequences is
the more local similarity we have after BWT.

H. Kaplan, S. Landau and E. Verbin, 2007. The more local similarity is found in the BWT of the string, the better the compression is.

Local Entropy: a statistic that measures local similarity.

They get an upper bound of compression ratio in terms of Local Entropy of $bwt(v)$.

Balancing

- A (finite or infinite) word v is *balanced* if for each letter a of the alphabet A and for all factors u and u' of v s.t. $|u| = |u'|$ we have that

$$||u|_a - |u'|_a| \leq 1$$

- A finite word v is *circularly balanced* if v^ω is balanced, i.e. all its conjugates are balanced.

Example

- $w =$ *cacbcac* is a circularly balanced word.
- $v =$ *acacbbc* is an unbalanced word.
- $u =$ *babaabaab* is a balanced but not circularly balanced word.

Denote by $\mathcal{B}$ the set of circularly balanced words.

Laurent Vuillon. Balanced words. *Bull. Belg. Math.Soc.*, 10(5):787–805, 2003.

Balancing

- A (finite or infinite) word v is *balanced* if for each letter a of the alphabet A and for all factors u and u' of v s.t. $|u| = |u'|$ we have that
$$||u|_a - |u'|_a| \leq 1$$
- A finite word v is *circularly balanced* if v^ω is balanced, i.e. all its conjugates are balanced.

Example

- $w = cacbcac$ is a circularly balanced word.
- $v = acacbbc$ is an unbalanced word.
- $u = babaabaab$ is a balanced but not circularly balanced word.

Denote by $\mathcal{B}$ the set of circularly balanced words.

Laurent Vuillon. Balanced words. *Bull. Belg. Math.Soc.*, 10(5):787–805, 2003.

Balancing

- A (finite or infinite) word v is *balanced* if for each letter a of the alphabet A and for all factors u and u' of v s.t. $|u| = |u'|$ we have that

$$||u|_a - |u'|_a| \leq 1$$

- A finite word v is *circularly balanced* if v^ω is balanced, i.e. all its conjugates are balanced.

Example

- $w = cacbcac$ is a circularly balanced word.
- $v = acacbbc$ is an unbalanced word.
- $u = babaabaab$ is a balanced but not circularly balanced word.

Denote by $\mathcal{B}$ the set of circularly balanced words.

Laurent Vuillon. Balanced words. *Bull. Belg. Math.Soc.*, 10(5):787–805, 2003.

Constant gap words and Clustered words

- A finite word v is *constant gap* if, for each letter a, the distance (the number of letters) between two consecutive occurrences of a is constant.

Example

The word *abcabdabcabe* is constant gap.

- Constant gap words are a special case of circularly balanced words.
- We remark that in a circularly balanced word, for each letter a, the distance between two consecutive occurrences of a is d or $d+1$.

The word v is a *clustered word* if the number of runs is equal to the size of alphabet.

Example

The word *dddddccccaaaaabbb* is a clustered word.

Constant gap words and Clustered words

- A finite word v is *constant gap* if, for each letter a, the distance (the number of letters) between two consecutive occurrences of a is constant.

Example

The word *abcabdabcabe* is constant gap.

- Constant gap words are a special case of circularly balanced words.
- We remark that in a circularly balanced word, for each letter a, the distance between two consecutive occurrences of a is d or $d+1$.

The word v is a *clustered word* if the number of runs is equal to the size of alphabet.

Example

The word *dddddddccccaaaaabbb* is a clustered word.

Constant gap words and Clustered words

- A finite word v is *constant gap* if, for each letter a, the distance (the number of letters) between two consecutive occurrences of a is constant.

Example

The word *abcabdabcabe* is constant gap.

- Constant gap words are a special case of circularly balanced words.
- We remark that in a circularly balanced word, for each letter a, the distance between two consecutive occurrences of a is d or $d + 1$.

The word v is a *clustered word* if the number of runs is equal to the size of alphabet.

Example

The word *ddddddccccaaaaabbb* is a clustered word.

Distance coding and Local Entropy

Distance coding: for each symbol of the input word, the DC algorithm outputs the distance to the previous occurrence of the same symbol (in circular way).

Example

v=	a	c	b	c	a	a	b
dc(v)=	1	4	2	1	3	0	3

Let $v = b_1 b_2 \cdots b_n$, $b_i \in A$ and $dc(v) = d_1 d_2 \cdots d_n$. Define the Local Entropy of v:

$$LE(v) = \frac{1}{n} \sum_{i=1}^{n} \log(d_i + 1)$$

Local entropy (LE) was considered by

- J. L. Bentley, D. D. Sleator, R. E. Tarjan, and V. K. Wei,1986
- G. Manzini, 2001
- H. Kaplan, S. Landau and E. Verbin, 2007

Bounds

Theorem

For any word v one has:

- $\Lambda(v) \leq LE(v) \leq H_0(v)$
- $LE(v) = H_0(v)$ *if and only if v is a constant gap word.*
- $LE(v) = \Lambda(v)$ *if and only if v is a clustered word.*

where

$$H_0(v) = \sum_{a \in A} \frac{|v|_a}{|v|} \log \frac{|v|}{|v|_a},$$

$$\Lambda(v) = \sum_{a \in A} \frac{1}{|v|} [\log(|v| - |v|_a + 1)]$$

Measure

- For any word v:

$$\delta(v) = \frac{H_0(v) - LE(v)}{H_0(v) - \Lambda(v)}, \qquad \tau(v) = \frac{LE(v) - \Lambda(v)}{H_0(v) - \Lambda(v)}$$

- Now, by using δ and τ, we can test, in a quantitative way, our intuition, i.e. the more is balanced the input sequences the more is local similarity after BWT.

- The experiments reported in the next slide confirm our intuition: actually they show that when $\delta(v)$ is less than 0.23, then $\tau(bwt(v))$ is less than 0.3 and the BWT-based compressor has good performances.

Experiments

File name	Size	H_0	Bst	Gzip	Diff %	$\delta(v)$	$\tau(bwt(v))$
bible	4,047,392	4.343	796,231	1,191,071	9.755	0.117	0.233
english	52,428,800	4.529	11,533,171	19,672,355	15.524	0.136	0.238
etext99	105,277,340	4.596	24,949,871	39,493,346	13.814	0.141	0.264
english	104,857,600	4.556	23,993,810	39,437,704	14.728	0.143	0.250
dblp.xml	52,428,800	5.230	4,871,450	9,034,902	7.941	0.152	0.093
dblp.xml	104,857,600	5.228	9,427,936	17,765,502	7.951	0.153	0.090
dblp.xml	209,715,200	5.257	18,522,167	35,897,168	8.285	0.162	0.088
dblp.xml	296,135,874	5.262	25,597,003	50,481,103	8.403	0.164	0.086
world192	2,473,400	4.998	430,225	724,606	11.902	0.174	0.183
rctail96	114,711,151	5.154	11,429,406	24,007,508	10.965	0.178	0.097
sprot34.dat	109,617,186	4.762	18,850,472	26,712,981	7.173	0.215	0.206
jdk13c	69,728,899	5.531	3,187,900	7,525,172	6.220	0.224	0.041
howto	39,886,973	4.857	8,713,851	12,638,334	9.839	0.231	0.229
rfc	116,421,901	4.623	17,565,908	26,712,981	7.857	0.239	0.163
w3c2	104,201,579	5.954	7,021,478	15,159,804	7.810	0.246	0.058
chr22.dna	34,553,758	2.137	8,015,707	8,870,068	2.473	0.341	0.575
pitches	52,428,800	5.633	18,651,999	16,884,651	-3.371	0.530	0.344
pitches	55,832,855	5.628	19,475,065	16,040,370	-6.152	0.533	0.337

Practical application: the computation of $\delta(v)$ is a fast test for the choice between bst and gzip.

Extremal case: Balanced words
Binary case

- An infinite aperiodic sequence v is balanced if and only if v is a sturmian sequence.
- An infinite periodic sequence v^ω is balanced if and only if v is a conjugate of a standard word.

Fibonacci words

$f_0 = b$
$f_1 = a$
$f_2 = ab$
$f_3 = aba$

$$f_0 = b \qquad f_1 = a$$
$$f_{n+1} = f_n f_{n-1} \ (n \geq 1)$$

Standard words

Directive sequence $d_1, d_2, \ldots, d_n, \ldots$, with $d_1 \geq 0$ and $d_i > 0$ for $i = 2, \ldots, n, \ldots$.

$$s_0 = b \qquad s_1 = a \qquad s_{n+1} = s_n^{d_n} s_{n-1} \text{ for } n \geq 1$$

Standard words are special prefixes of Sturmian sequences.

Extremal case: Balanced words
Binary case

- An infinite aperiodic sequence v is balanced if and only if v is a sturmian sequence.
- An infinite periodic sequence v^ω is balanced if and only if v is a conjugate of a standard word.

Fibonacci words

$f_0 = b$
$f_1 = a$
$f_2 = ab$
$f_3 = aba$

$$f_0 = b \qquad f_1 = a$$
$$f_{n+1} = f_n f_{n-1} \ (n \geq 1)$$

Standard words

Directive sequence $d_1, d_2, \ldots, d_n, \ldots$, with $d_1 \geq 0$ and $d_i > 0$ for $i = 2, \ldots, n, \ldots$.

$$s_0 = b \qquad s_1 = a \qquad s_{n+1} = s_n^{d_n} s_{n-1} \text{ for } n \geq 1$$

Standard words are special prefixes of Sturmian sequences.

Binary alphabets

Theorem (Mantaci, R. and Sciortino, 2003)

Given a word $v \in \{a, b\}$, the following conditions are equivalent:

1. $bwt(v) = b^p a^q$ *with* $p, q \geq 1$*;*
2. *v is a circularly balanced word;*
3. *v is a conjugate of a power of a Standard words.*

- The words in this theorem correspond to the Christoffel classes investigated in Borel and Reutenauer, 2006.
- They appear in several contexts and applications (G. Castiglione, A. R., M. Sciortino, Circular Sturmian words and Hopcroft algorithm, 2009)
- In alphabets with more than two letters, the notions considered in the previous theorem (or their generalization) do not coincide.

Circularly Balanced words on larger alphabets

- If $|A| > 2$, the general structure of circularly balanced words is not known.
  E. Altman, B. Gaujal, and A. Hordijk, 2000
  R. Mantaci, S. Mantaci, and A. R., 2008
- We note that the notion of circularly balanced words over an alphabet of size larger than two also appears in the statement of the Fraenkel's conjecture.
- As a direct consequence of a result of Graham, one has that balanced sequences on a set of letters having different frequencies must be periodic, i.e. of the form v^ω, where v is a circularly balanced word.

Fraenkel's conjecture

Let $A_k = \{a_1, a_2, \ldots, a_k\}$. For each $k > 2$, there is only one circularly balanced word $F_k \in A_k^*$, having different frequencies. It is defined recursively as follow $F_1 = a_1$ and $F_k = F_{k-1} a_k F_{k-1}$ for all $k \geq 2$.

Simple BWT words

In 2008, Simpson and Puglisi introduce the notion of *Simple BWT words*.

Let v be a word over a finite ordered alphabet $A = \{a_1, a_2, \ldots, a_k\}$, with $a_1 < a_2 < \ldots < a_k$. The word v is a *simple BWT word* if

$$bwt(v) = a_k^{n_k} a_{k-1}^{n_{k-1}} \cdots a_2^{n_2} a_1^{n_1}$$

for some non-negative integers $n_1, n_2, \ldots, n_k$.

We denote by S the set of the simple BWT words.

Example

$v = acbcbcadad \in S$, in fact $bwt(v) = ddcccbbaaa$.

Simpson and Puglisi get a constructive characterization of the set S in the case of three letters alphabet.

Simple BWT words

In 2008, Simpson and Puglisi introduce the notion of *Simple BWT words*.

Let v be a word over a finite ordered alphabet $A = \{a_1, a_2, \ldots, a_k\}$, with $a_1 < a_2 < \ldots < a_k$. The word v is a *simple BWT word* if

$$bwt(v) = a_k^{n_k} a_{k-1}^{n_{k-1}} \cdots a_2^{n_2} a_1^{n_1}$$

for some non-negative integers $n_1, n_2, \ldots, n_k$.

We denote by S the set of the simple BWT words.

Example

$v = acbcbcadad \in S$, in fact $bwt(v) = ddcccbbaaa$.

Simpson and Puglisi get a constructive characterization of the set S in the case of three letters alphabet.

Simple BWT words

In 2008, Simpson and Puglisi introduce the notion of *Simple BWT words*.

Let v be a word over a finite ordered alphabet $A = \{a_1, a_2, \ldots, a_k\}$, with $a_1 < a_2 < \ldots < a_k$. The word v is a *simple BWT word* if

$$bwt(v) = a_k^{n_k} a_{k-1}^{n_{k-1}} \cdots a_2^{n_2} a_1^{n_1}$$

for some non-negative integers $n_1, n_2, \ldots, n_k$.

We denote by S the set of the simple BWT words.

Example

$v = acbcbcadad \in S$, in fact $bwt(v) = ddcccbbaaa$.

Simpson and Puglisi get a constructive characterization of the set S in the case of three letters alphabet.

Simple BWT words

In 2008, Simpson and Puglisi introduce the notion of *Simple BWT words*.

Let v be a word over a finite ordered alphabet $A = \{a_1, a_2, \ldots, a_k\}$, with $a_1 < a_2 < \ldots < a_k$. The word v is a *simple BWT word* if

$$bwt(v) = a_k^{n_k} a_{k-1}^{n_{k-1}} \cdots a_2^{n_2} a_1^{n_1}$$

for some non-negative integers $n_1, n_2, \ldots, n_k$.

We denote by S the set of the simple BWT words.

Example

$v = acbcbcadad \in S$, in fact $bwt(v) = ddcccbbaaa$.

Simpson and Puglisi get a constructive characterization of the set S in the case of three letters alphabet.

Matrix M and R

$$
\begin{array}{cccccc}
 & & M & & & \\
F_M & & & & & L_M \\
a & a & b & r & a & c \\
a & b & r & a & c & a \\
a & c & a & a & b & r \\
b & r & a & c & a & a \\
c & a & a & b & r & a \\
r & a & c & a & a & b
\end{array}
\qquad
\begin{array}{cccccc}
 & & R & & & \\
F_R & & & & & L_R \\
b & a & a & c & a & r \\
a & r & b & a & a & c \\
a & a & c & a & r & b \\
r & b & a & a & c & a \\
a & c & a & r & b & a \\
c & a & r & b & a & a
\end{array}
$$

The matrix R is obtained from M by a rotation of 180°: it follows that the ith conjugate of M is the reverse of the $(n-i+1)$th conjugate of R.

Theorem

A word $v \in S$ if and only if $M = R$.

Matrix M and R

$$
\begin{array}{cccccc}
\multicolumn{6}{c}{M} \\
F_M & & & & & L_M \\
a & a & b & r & a & c \\
a & b & r & a & c & a \\
a & c & a & a & b & r \\
b & r & a & c & a & a \\
c & a & a & b & r & a \\
r & a & c & a & a & b
\end{array}
\qquad
\begin{array}{cccccc}
\multicolumn{6}{c}{R} \\
F_R & & & & & L_R \\
b & a & a & c & a & r \\
a & r & b & a & a & c \\
a & a & c & a & r & b \\
r & b & a & a & c & a \\
a & c & a & r & b & a \\
c & a & r & b & a & a
\end{array}
$$

The matrix R is obtained from M by a rotation of 180°: it follows that the ith conjugate of M is the reverse of the $(n-i+1)$th conjugate of R.

Theorem

A word $v \in S$ if and only if $M = R$.

A word $v \in S$ iff
$M = R$

$$\begin{matrix} a & c & a & d & a & c & b & b & c & a & d \\ a & c & b & b & c & a & d & a & c & a & d \\ a & d & a & c & a & d & a & c & b & b & c \\ a & d & a & c & b & b & c & a & d & a & c \\ b & b & c & a & d & a & c & a & d & a & c \\ b & c & a & d & a & c & a & d & a & c & b \\ c & a & d & a & c & a & d & a & c & b & b \\ c & a & d & a & c & b & b & c & a & d & a \\ c & b & b & c & a & d & a & c & a & d & a \\ d & a & c & a & d & a & c & b & b & c & a \\ d & a & c & b & b & c & a & d & a & c & a \end{matrix}$$

$v_i = \widetilde{v_{n-i+1}}$

So $[v]$ and its factors are closed under reverse. Under these conditions each conjugate of v has the two palindrome property, i.e. v is product of two palindromes (cf. Simpson and Puglisi, 2008).

A word $v \in S$ iff
$M = R$

$$\begin{array}{ccccccccccc}
a & c & a & d & a & c & b & b & c & a & d \\
a & c & b & b & c & a & d & a & c & a & d \\
a & d & a & c & a & d & a & c & b & b & c \\
a & d & a & c & b & b & c & a & d & a & c \\
b & b & c & a & d & a & c & a & d & a & c \\
b & c & a & d & a & c & a & d & a & c & b \\
c & a & d & a & c & a & d & a & c & b & b \\
c & a & d & a & c & b & b & c & a & d & a \\
c & b & b & c & a & d & a & c & a & d & a \\
d & a & c & a & d & a & c & b & b & c & a \\
d & a & c & b & b & c & a & d & a & c & a
\end{array}$$

$$v_i = \widetilde{v_{n-i+1}}$$

So $[v]$ and its factors are closed under reverse. Under these conditions each conjugate of v has the two palindrome property, i.e. v is product of two palindromes (cf. Simpson and Puglisi, 2008).

Balanced and Simple BWT words

$$\mathcal{B} \neq \mathcal{S}$$

The set of circularly balanced words over more than two letters alphabets does not coincide with the set of Simple BWT words.

Example

- $v = cacbcac$ is circularly balanced and $bwt(v) = ccccbaa$
- $w = ababc$ is circularly balanced and $bwt(w) = cbaab$
- $u = acacbbc$ is unbalanced and $bwt(u) = cccbbaa$

A generalization of Sturmian: Episturmian

- An infinite word t on A is *episturmian* (Droubay, J. Justin, G. Pirillo, 2001) if:
  - $F(t)$ (its set of factors) is closed under reversal;
  - t has at most one right special factor of each length.

Let s be an infinite word, then a factor u of s is *right* (resp. *left*) *special* if there exist $x, y \in A$, $x \neq y$, such that $ux, uy \in F(s)$ (resp. $xu, yu \in F(s)$).

The *palindromic right-closure* $v^{(+)}$ of a finite word v is the (unique) shortest palindrome having v as a prefix (A. de Luca, 1997).

The *iterated palindromic closure* function (J. Justin, 2005), denoted by Pal, is recursively defined as follows. Set $Pal(\varepsilon) = \varepsilon$ and, for any word v and letter x, define $Pal(vx) = (Pal(v)x)^{(+)}$.

Amy Glen and Jacques Justin. Episturmian words: a survey. *RAIRO Theoretical Informatics and Applications*, 2009.

A generalization of Sturmian: Episturmian

- An infinite word t on A is *episturmian* (Droubay, J. Justin, G. Pirillo, 2001) if:
  - $F(t)$ (its set of factors) is closed under reversal;
  - t has at most one right special factor of each length.

Let s be an infinite word, then a factor u of s is *right* (resp. *left*) *special* if there exist $x, y \in A$, $x \neq y$, such that $ux, uy \in F(s)$ (resp. $xu, yu \in F(s)$).

The *palindromic right-closure* $v^{(+)}$ of a finite word v is the (unique) shortest palindrome having v as a prefix (A. de Luca, 1997).

The *iterated palindromic closure* function (J. Justin, 2005), denoted by Pal, is recursively defined as follows. Set $Pal(\varepsilon) = \varepsilon$ and, for any word v and letter x, define $Pal(vx) = (Pal(v)x)^{(+)}$.

Amy Glen and Jacques Justin. Episturmian words: a survey. *RAIRO Theoretical Informatics and Applications*, 2009.

A generalization of Sturmian: Episturmian

- An infinite word t on A is *episturmian* (Droubay, J. Justin, G. Pirillo, 2001) if:
  - $F(t)$ (its set of factors) is closed under reversal;
  - t has at most one right special factor of each length.

Let s be an infinite word, then a factor u of s is *right* (resp. *left*) *special* if there exist $x, y \in A$, $x \neq y$, such that $ux, uy \in F(s)$ (resp. $xu, yu \in F(s)$).

The *palindromic right-closure* $v^{(+)}$ of a finite word v is the (unique) shortest palindrome having v as a prefix (A. de Luca, 1997).

The *iterated palindromic closure* function (J. Justin, 2005), denoted by *Pal*, is recursively defined as follows. Set $Pal(\varepsilon) = \varepsilon$ and, for any word v and letter x, define $Pal(vx) = (Pal(v)x)^{(+)}$.

Amy Glen and Jacques Justin. Episturmian words: a survey. *RAIRO Theoretical Informatics and Applications*, 2009.

A generalization of Standard: Finite epistandard

Rauzy rules.

	Rules	1	2	3	4
R_0		a	b	c	d
R_1	1	a	ab	ac	ad
R_2	1	a	aab	aac	aad
R_3	4	$aada$	$aadaab$	$aadaac$	aad
R_4	3	$aadaacaada$	$aadaacaadaab$	$aadaac$	$aadaacaad$

- Let $|A| = k$. A word $v \in A^*$ is called finite epistandard if v is an element of a k-tuples R_n, for some $n = 1$.
- We denote by $\mathcal{EP}$ the set of words that are powers of a conjugate of a finite epistandard word.
- The elements of $\mathcal{EP}$ are closed related to epichristoffel classes (G. Paquin, 2009)

A generalization of Standard: Finite epistandard

Rauzy rules.

	Rules	1	2	3	4
R_0		a	b	c	d
R_1	1	a	ab	ac	ad
R_2	1	a	aab	aac	aad
R_3	4	$aada$	$aadaab$	$aadaac$	aad
R_4	3	$aadaacaada$	$aadaacaadaab$	$aadaac$	$aadaacaad$

- Let $|A| = k$. A word $v \in A^*$ is called finite epistandard if v is an element of a k-tuples R_n, for some $n = 1$.
- We denote by $\mathcal{EP}$ the set of words that are powers of a conjugate of a finite epistandard word.
- The elements of $\mathcal{EP}$ are closed related to epichristoffel classes (G. Paquin, 2009)

A generalization of Standard: Finite epistandard

Rauzy rules.

	Rules	1	2	3	4
R_0		a	b	c	d
R_1	1	a	ab	ac	ad
R_2	1	a	aab	aac	aad
R_3	4	$aada$	$aadaab$	$aadaac$	aad
R_4	3	$aadaacaada$	$aadaacaadaab$	$aadaac$	$aadaacaad$

- Let $|A| = k$. A word $v \in A^*$ is called finite epistandard if v is an element of a k-tuples R_n, for some $n = 1$.
- We denote by $\mathcal{EP}$ the set of words that are powers of a conjugate of a finite epistandard word.
- The elements of $\mathcal{EP}$ are closed related to epichristoffel classes (G. Paquin, 2009)

A generalization of Standard: Finite epistandard

Rauzy rules.

	Rules	1	2	3	4
R_0		a	b	c	d
R_1	1	a	ab	ac	ad
R_2	1	a	aab	aac	aad
R_3	4	$aada$	$aadaab$	$aadaac$	aad
R_4	3	$aadaacaada$	$aadaacaadaab$	$aadaac$	$aadaacaad$

- Let $|A| = k$. A word $v \in A^*$ is called finite epistandard if v is an element of a k-tuples R_n, for some $n = 1$.
- We denote by $\mathcal{EP}$ the set of words that are powers of a conjugate of a finite epistandard word.
- The elements of $\mathcal{EP}$ are closed related to epichristoffel classes (G. Paquin, 2009)

A generalization of Standard: Finite epistandard

Rauzy rules.

	Rules	1	2	3	4
R_0		a	b	c	d
R_1	1	a	ab	ac	ad
R_2	1	a	aab	aac	aad
R_3	4	$aada$	$aadaab$	$aadaac$	aad
R_4	3	$aadaacaada$	$aadaacaadaab$	$aadaac$	$aadaacaad$

- Let $|A| = k$. A word $v \in A^*$ is called finite epistandard if v is an element of a k-tuples R_n, for some $n = 1$.
- We denote by $\mathcal{EP}$ the set of words that are powers of a conjugate of a finite epistandard word.
- The elements of $\mathcal{EP}$ are closed related to epichristoffel classes (G. Paquin, 2009)

Balancing and Epistandard

$$\mathcal{B} \neq \mathcal{EP}$$

The set of circularly balanced words over more than two letters alphabets does not coincide with the set of conjugate of powers of epistandard words.

Example

- $v = aadaacaad$ is epistandard, but it is not circularly balanced.
- $u = abcabdabcabe$ is circularly balanced, but it is not epistandard.

Palindromic Richness

Droubay, Justin, Pirillo, 2001:

- The number of distinct palindromic factors (including ε) of a word v is at most $|v| + 1$
- A finite word v is (palindromic) rich if it has exactly $|v| + 1$ distinct palindromic factors, including ε.
- A factor of finite rich word is rich.
- A infinite word is rich if all of its factors are rich.

Example

$v = ccaacb$ is rich, $|v| = 6$, in fact: $P(v) = \{\varepsilon, c, cc, caac, a, aa, b\}$, $|P(v)| = 7$.

A. Glen, J. Justin, S. Widmer, and L. Q. Zamboni. Palindromic richness. *European Journal of Combinatorics*, 30(2):510–531, 2009.

Palindromic Richness

Droubay, Justin, Pirillo, 2001:

- The number of distinct palindromic factors (including ε) of a word v is at most $|v| + 1$
- A finite word v is (palindromic) rich if it has exactly $|v| + 1$ distinct palindromic factors, including ε.
- A factor of finite rich word is rich.
- A infinite word is rich if all of its factors are rich.

Example

$v = ccaacb$ is rich, $|v| = 6$, in fact: $P(v) = \{\varepsilon, c, cc, caac, a, aa, b\}$, $|P(v)| = 7$.

A. Glen, J. Justin, S. Widmer, and L. Q. Zamboni. Palindromic richness. *European Journal of Combinatorics*, 30(2):510–531, 2009.

Palindromic Richness

Droubay, Justin, Pirillo, 2001:

- The number of distinct palindromic factors (including ε) of a word v is at most $|v| + 1$
- A finite word v is (palindromic) rich if it has exactly $|v| + 1$ distinct palindromic factors, including ε.
- A factor of finite rich word is rich.
- A infinite word is rich if all of its factors are rich.

Example

$v = ccaacb$ is rich, $|v| = 6$, in fact: $P(v) = \{\varepsilon, c, cc, caac, a, aa, b\}$, $|P(v)| = 7$.

A. Glen, J. Justin, S. Widmer, and L. Q. Zamboni. Palindromic richness. *European Journal of Combinatorics*, 30(2):510–531, 2009.

Palindromic Richness

Droubay, Justin, Pirillo, 2001:

- The number of distinct palindromic factors (including ε) of a word v is at most $|v| + 1$
- A finite word v is (palindromic) rich if it has exactly $|v| + 1$ distinct palindromic factors, including ε.
- A factor of finite rich word is rich.
- A infinite word is rich if all of its factors are rich.

Example

$v = ccaacb$ is rich, $|v| = 6$, in fact: $P(v) = \{\varepsilon, c, cc, caac, a, aa, b\}$, $|P(v)| = 7$.

A. Glen, J. Justin, S. Widmer, and L. Q. Zamboni. Palindromic richness. *European Journal of Combinatorics*, 30(2):510–531, 2009.

Palindromic Richness

Droubay, Justin, Pirillo, 2001:

- The number of distinct palindromic factors (including ε) of a word v is at most $|v| + 1$
- A finite word v is (palindromic) rich if it has exactly $|v| + 1$ distinct palindromic factors, including ε.
- A factor of finite rich word is rich.
- A infinite word is rich if all of its factors are rich.

Example

$v = ccaacb$ is rich, $|v| = 6$, in fact: $P(v) = \{\varepsilon, c, cc, caac, a, aa, b\}$, $|P(v)| = 7$.

A. Glen, J. Justin, S. Widmer, and L. Q. Zamboni. Palindromic richness. *European Journal of Combinatorics*, 30(2):510–531, 2009.

Circularly rich words

Lemma (Glen, Justin, Widmer and Zamboni, 2009)

For a finite word v, the following properties are equivalent:

1. v^ω *is rich;*
2. v^2 *is rich;*
3. v *is a product of two palindromes and all of the conjugates of* v *(including itself) are rich.*

- We say that a finite word v is circularly rich if the infinite word v^ω is rich.
- We denote by $\mathcal{R}$ the set of the circularly rich words.

Circularly rich words

Lemma (Glen, Justin, Widmer and Zamboni, 2009)

For a finite word v, the following properties are equivalent:

1. v^ω *is rich;*
2. v^2 *is rich;*
3. v *is a product of two palindromes and all of the conjugates of* v *(including itself) are rich.*

- We say that a finite word v is circularly rich if the infinite word v^ω is rich.
- We denote by $\mathcal{R}$ the set of the circularly rich words.

Circularly rich words

Lemma (Glen, Justin, Widmer and Zamboni, 2009)

For a finite word v, the following properties are equivalent:

1. v^ω *is rich;*
2. v^2 *is rich;*
3. v *is a product of two palindromes and all of the conjugates of* v *(including itself) are rich.*

- We say that a finite word v is circularly rich if the infinite word v^ω is rich.
- We denote by $\mathcal{R}$ the set of the circularly rich words.

Balancing and Richness

$$\mathcal{R} \neq \mathcal{B}$$

The set of circularly balanced words over more than two letters alphabets does not coincide with the set of circularly rich words.

- The word $w = bbbbbacaca$ is circularly rich, but it is not circularly balanced.
- The word $u = abcabdabcabe$ is circularly balanced, but it is not circularly rich.

$\mathcal{S} \cap \mathcal{B} = \mathcal{R} \cap \mathcal{B} = \mathcal{EP} \cap \mathcal{B}$

Theorem (R., Rosone, 2009)

Let $v \in A^*$ be a circularly balanced word over A. The following statements are equivalent:

i) *v is a simple BWT word;*
ii) *v is a circularly rich word;*
iii) *v is a conjugate of a power of a finite epistandard word.*

Proof: $3 \rightarrow 1$: The finite balanced epistandard words belong to $\mathcal{S}$.

From a result of Paquin and Vuillon (2006), one can prove that each finite balanced epistandard word t is of the form:

i) $t = pa_2$, with $p = Pal(a_1^m a_k a_{k-1} \cdots a_3)$, where $k \geq 3$ and $m \geq 1$;
ii) $t = pa_2$, with $p = Pal(a_1 a_k a_{k-1} \cdots a_{k-\ell} a_1 a_{k-\ell-1} a_{k-\ell-2} \cdots a_3)$, where $0 \leq \ell \leq k-4$ and $k \geq 4$;
iii) $t = Pal(a_1 a_k a_{k-1} \cdots a_2)$, where $k \geq 3$ (*Fraenkel's words*).

In order to prove that t belongs to $\mathcal{S}$ it suffices to show that words of the form $i), ii)$ and $iii)$ have simple BWT.

Proof: $3 \to 1$: The finite balanced epistandard words belong to $\mathcal{S}$.

From a result of Paquin and Vuillon (2006), one can prove that each finite balanced epistandard word t is of the form:

i) $t = pa_2$, with $p = Pal(a_1^m a_k a_{k-1} \cdots a_3)$, where $k \geq 3$ and $m \geq 1$;

ii) $t = pa_2$, with $p = Pal(a_1 a_k a_{k-1} \cdots a_{k-\ell} a_1 a_{k-\ell-1} a_{k-\ell-2} \cdots a_3)$, where $0 \leq \ell \leq k-4$ and $k \geq 4$;

iii) $t = Pal(a_1 a_k a_{k-1} \cdots a_2)$, where $k \geq 3$ (*Fraenkel's words*).

In order to prove that t belongs to $\mathcal{S}$ it suffices to show that words of the form *i*), *ii*) and *iii*) have simple BWT.

Proof: $2 \leftrightarrow 3$:
v is circularly rich if and only if v is a conjugate of a power of a finite epistandard.

The proof is a consequence of the following results:

- The set of the episturmian sequences is a subset of the set of the rich words (Glen, Justin, Widmer and Zamboni, 2009).
- Recurrent balanced rich infinite words are precisely the balanced episturmian words (Glen, Justin, Widmer and Zamboni, 2009).

Proof: $2 \leftrightarrow 3$: v is circularly rich if and only if v is a conjugate of a power of a finite epistandard.

The proof is a consequence of the following results:

- The set of the episturmian sequences is a subset of the set of the rich words (Glen, Justin, Widmer and Zamboni, 2009).
- Recurrent balanced rich infinite words are precisely the balanced episturmian words (Glen, Justin, Widmer and Zamboni, 2009).

Proof: $2 \leftrightarrow 3$: v is circularly rich if and only if v is a conjugate of a power of a finite epistandard.

The proof is a consequence of the following results:

- The set of the episturmian sequences is a subset of the set of the rich words (Glen, Justin, Widmer and Zamboni, 2009).
- Recurrent balanced rich infinite words are precisely the balanced episturmian words (Glen, Justin, Widmer and Zamboni, 2009).

Proof: $1 \to 3$

Theorem (R., Rosone, 2009)

If the word w belongs to $\mathcal{S}$ then w is circularly rich.

Example

The word $v = acbcbcadad \in \mathcal{S}$, $|v| = 10$, in fact
$bwt(acbcbcadad) = ddcccbbaaa$ $|P(v^2)| = 21$, so v is circularly rich.

We note that the converse of this result is false. The word $u = ccaaccb$ is circularly rich, but $bwt(ccaaccb) = cacccba$ $(u \notin \mathcal{S})$.

Proof: $1 \to 3$

Theorem (R., Rosone, 2009)

If the word w belongs to $\mathcal{S}$ then w is circularly rich.

Example

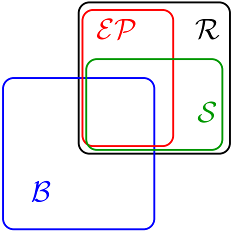
The word $v = acbcbcadad \in \mathcal{S}$, $|v| = 10$, in fact
$bwt(acbcbcadad) = ddcccbbaaa$ $|P(v^2)| = 21$, so v is circularly rich.

We note that the converse of this result is false. The word $u = ccaaccb$ is circularly rich, but $bwt(ccaaccb) = cacccba$ $(u \notin \mathcal{S})$.

Conclusions

Only under the condition of circularly balanced, the following statements are equivalent:

- $v \in \mathcal{S}$ (simple BWT words);
- v is circularly rich,
- v is a conjugate of a power of a finite epistandard.



The following example shows that there exist words **unbalanced** which belong to $\mathcal{EP} \cap \mathcal{S}$: $v = \underline{aad}\,\underline{aaca}\,a\underline{d}$ is not a circularly balanced word: $v \in \mathcal{EP}$ and $v \in \mathcal{S}$.