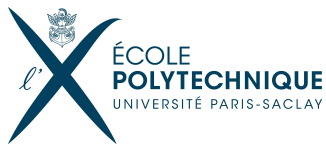
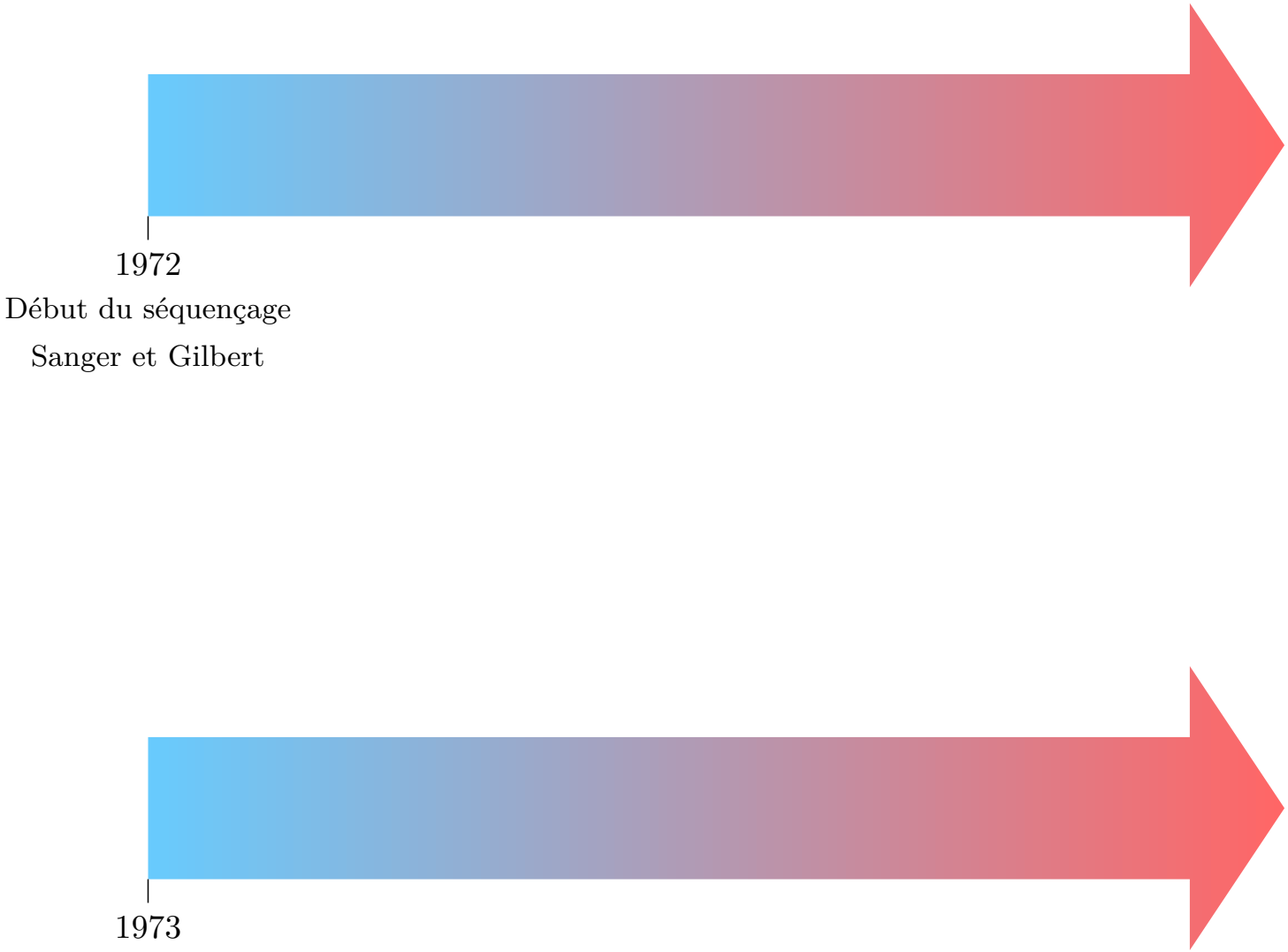
Analyse des séquences génomiques : Identification des ARNs circulaires et calcul de l'information négative

Soutenance de Thèse de Alice Héliou

Encadrée par Mireille Régnier (LIX) et co-encadrée par Hubert Becker (LOB)

LOB




1972
Début du séquençage
Sanger et Gilbert
1973
Arbre des suffixes
Weiner

Dévelopement des PCR

Réaction Polymérase en chaine

1990

1972

Début du séquençage

Sanger et Gilbert

Table des suffixes

Manber et Myers

&

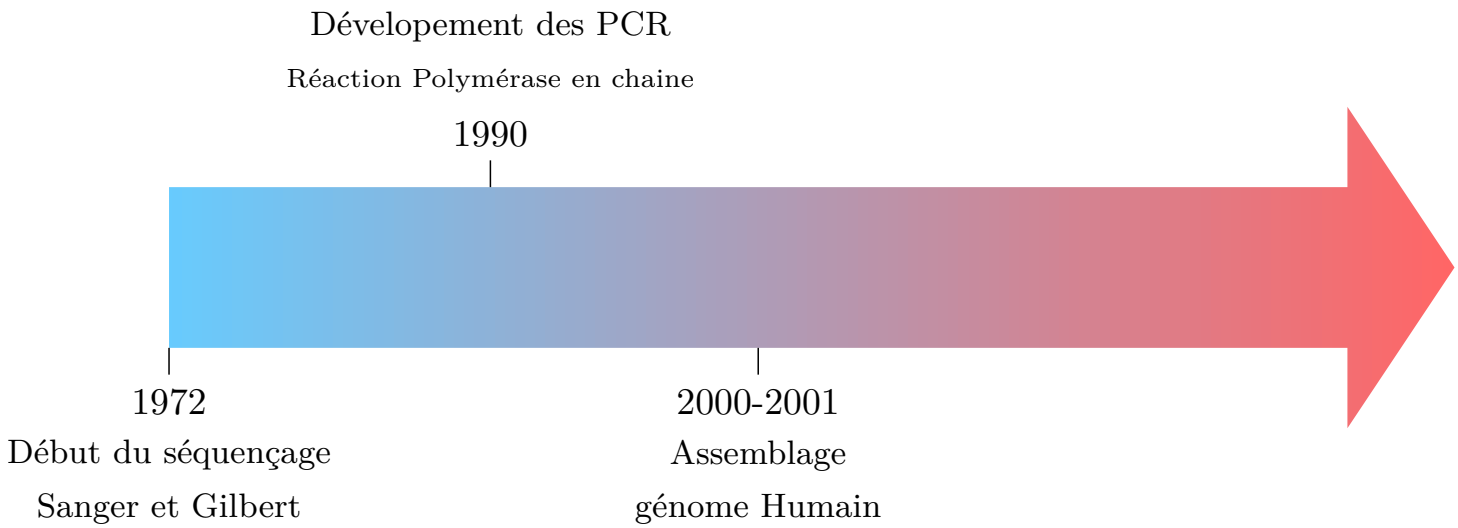
BLAST, Altschul

1990

1973

Arbre des suffixes

Weiner


Dévelopement des PCR
Réaction Polymérase en chaine
1990
1972
Début du séquençage
Sanger et Gilbert
2000-2001
Assemblage
génome Humain

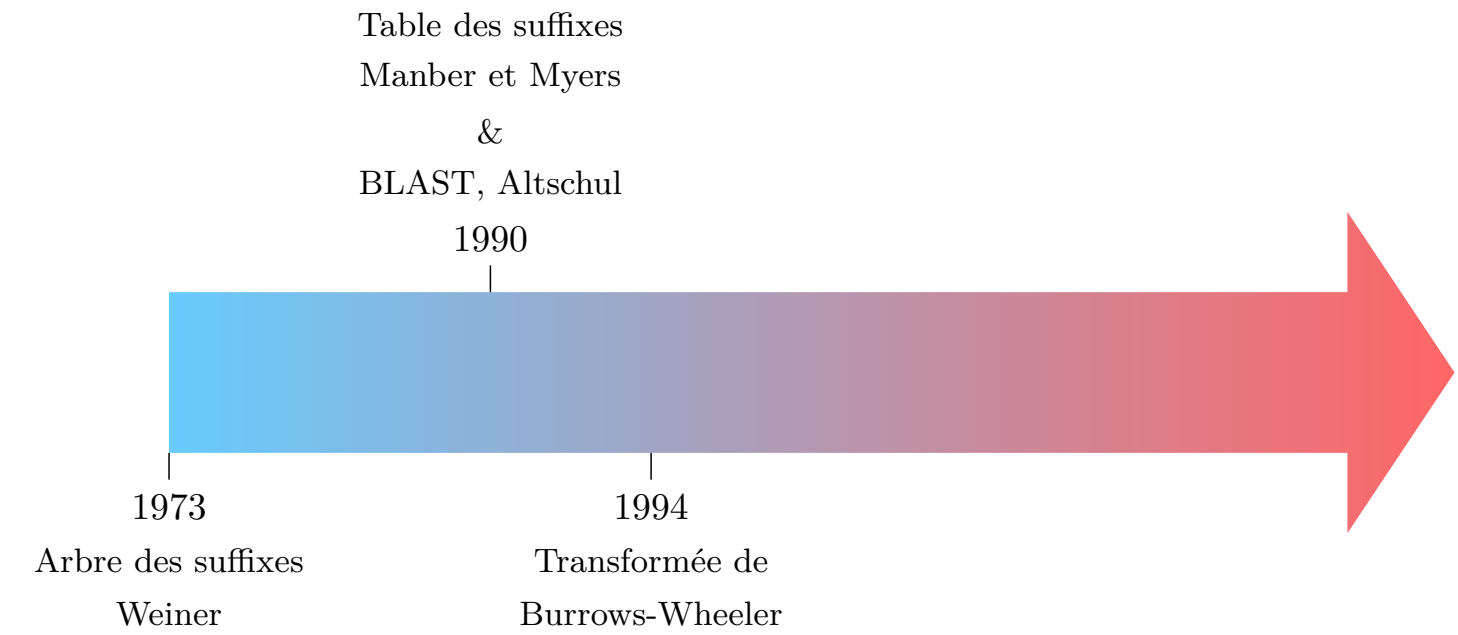
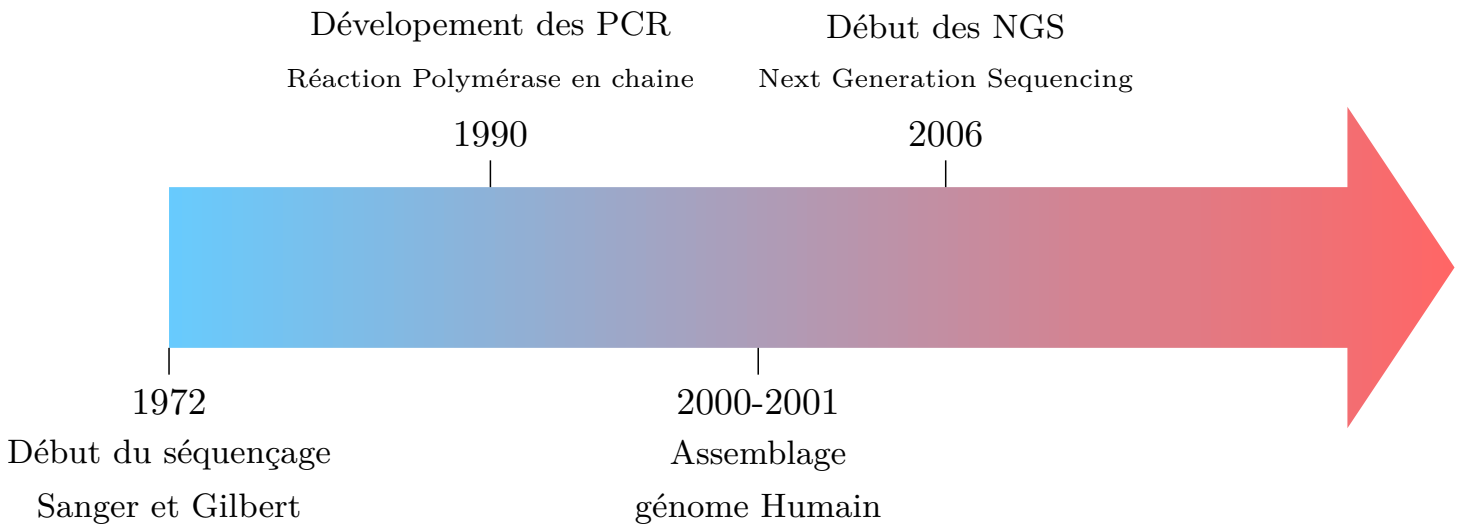


Table des suffixes
Manber et Myers
&
BLAST, Altschul
1990
1973
Arbre des suffixes
Weiner
1994
Transformée de
Burrows-Wheeler


Dévelopement des PCR
Réaction Polymérase en chaine
1990
Début des NGS
Next Generation Sequencing
2006
1972
Début du séquençage
Sanger et Gilbert
2000-2001
Assemblage
génome Humain

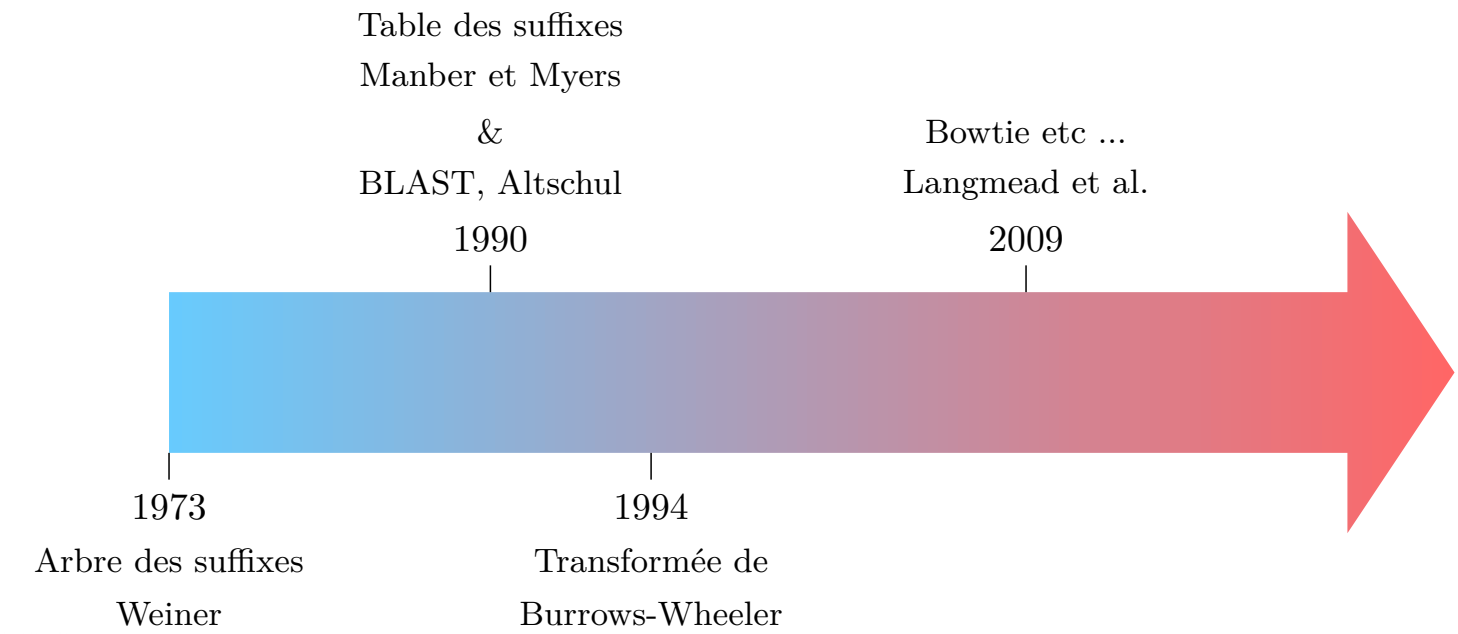
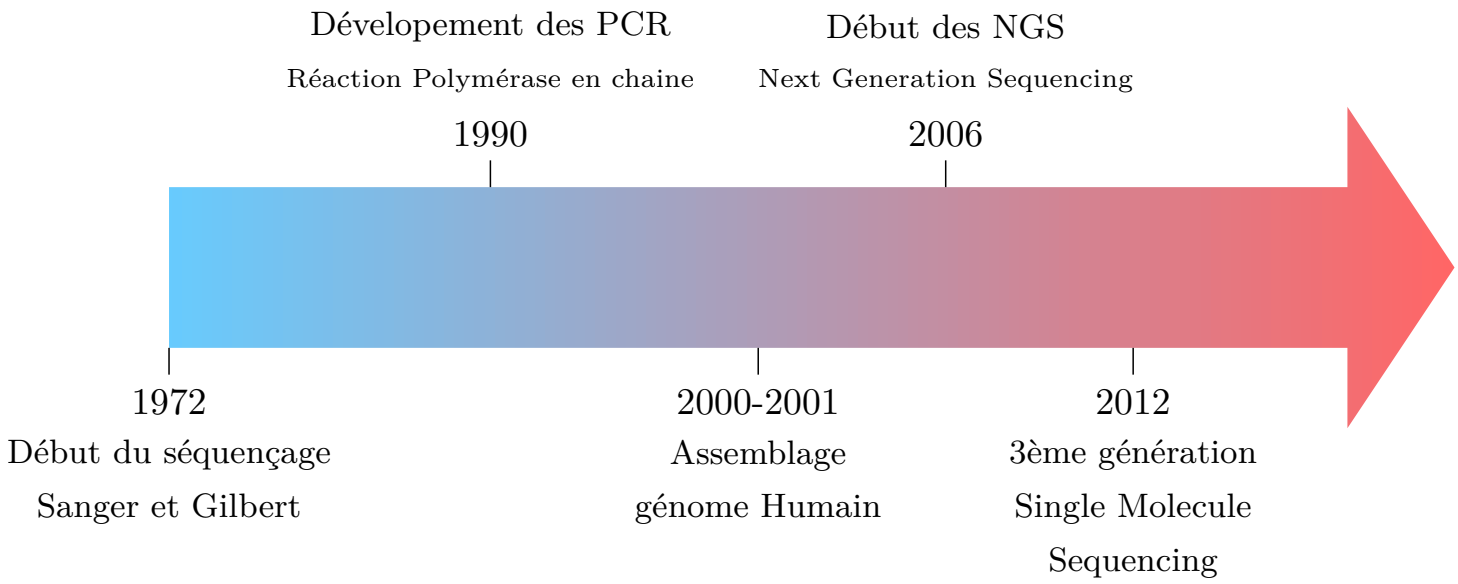


Table des suffixes
Manber et Myers
&
BLAST, Altschul
1990
Bowtie etc ...
Langmead et al.
2009
1973
Arbre des suffixes
Weiner
1994
Transformée de
Burrows-Wheeler


Dévelopement des PCR
Réaction Polymérase en chaine
1990
Début des NGS
Next Generation Sequencing
2006
1972
Début du séquençage
Sanger et Gilbert
2000-2001
Assemblage
génome Humain
2012
3ème génération
Single Molecule
Sequencing

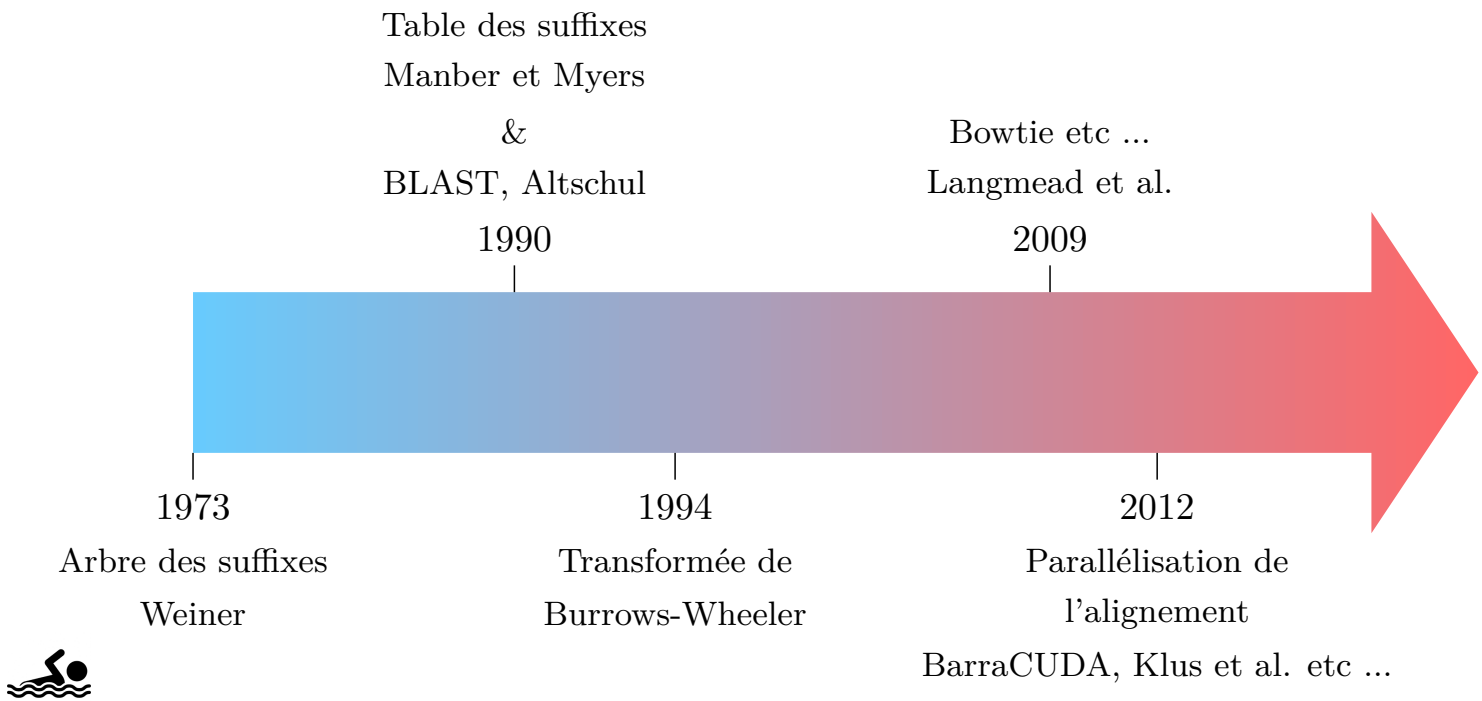


Table des suffixes
Manber et Myers
&
BLAST, Altschul
1990
Bowtie etc ...
Langmead et al.
2009
1973
Arbre des suffixes
Weiner
1994
Transformée de
Burrows-Wheeler
2012
Parallélisation de
l'alignement
BarraCUDA, Klus et al. etc ...

1. Ma structure de données préférée
   - La BWT et la table des suffixes
   - La table des LCP
   - Les applications
   - La compression des meta-data
2. Et si les absents avaient raison ?
3. Les cercles qui ne rentrent pas dans le moule

Table des suffixes (SA), 1990 et Transformée de Burrows-Wheeler (BWT), 1994

Table des suffixes : Index pour localiser des motifs.
BWT : Permutation réversible utilisée pour la compression et l'indexation.

Table des suffixes (SA), 1990 et Transformée de Burrows-Wheeler (BWT), 1994

Table des suffixes : Index pour localiser des motifs.
BWT : Permutation réversible utilisée pour la compression et l'indexation.

$$S = \overset{0}{A}\overset{1}{A}\overset{2}{C}\overset{3}{A}\overset{4}{C}\overset{5}{A}\overset{6}{C}\overset{7}{C}\overset{8}{\#}$$

0	A	A	C	A	C	A	C	C	#
1	A	C	A	C	A	C	C	#	A
2	C	A	C	A	C	C	#	A	A
3	A	C	A	C	C	#	A	A	C
4	C	A	C	C	#	A	A	C	A
5	A	C	C	#	A	A	C	A	C
6	C	C	#	A	A	C	A	C	A
7	C	#	A	A	C	A	C	A	C
8	#	A	A	C	A	C	A	C	C

Rotations de S | Rotations de S triées

Table des suffixes (SA), 1990 et Transformée de Burrows-Wheeler (BWT), 1994

Table des suffixes : Index pour localiser des motifs.
BWT : Permutation réversible utilisée pour la compression et l'indexation.

$$S = \overset{0}{A}\overset{1}{A}\overset{2}{C}\overset{3}{A}\overset{4}{C}\overset{5}{A}\overset{6}{C}\overset{7}{C}\overset{8}{\#}$$

0	A	A	C	A	C	A	C	C	#
1	A	C	A	C	A	C	C	#	A
2	C	A	C	A	C	C	#	A	A
3	A	C	A	C	C	#	A	A	C
4	C	A	C	C	#	A	A	C	A
5	A	C	C	#	A	A	C	A	C
6	C	C	#	A	A	C	A	C	A
7	C	#	A	A	C	A	C	A	C
8	#	**A**	**A**	**C**	**A**	**C**	**A**	**C**	**C**

Rotations de S

$\Rightarrow$

pos	BWT
8	# A A C A C A C C

Rotations de S triées

Table des suffixes (SA), 1990 et Transformée de Burrows-Wheeler (BWT), 1994

Table des suffixes : Index pour localiser des motifs.
BWT : Permutation réversible utilisée pour la compression et l'indexation.

$$S = \overset{0}{A}\overset{1}{A}\overset{2}{C}\overset{3}{A}\overset{4}{C}\overset{5}{A}\overset{6}{C}\overset{7}{C}\overset{8}{\#}$$

0	**A**	**A**	**C**	**A**	**C**	**A**	**C**	**C**	#
1	A	C	A	C	A	C	C	#	A
2	C	A	C	A	C	C	#	A	A
3	A	C	A	C	C	#	A	A	C
4	C	A	C	C	#	A	A	C	A
5	A	C	C	#	A	A	C	A	C
6	C	C	#	A	A	C	A	C	A
7	C	#	A	A	C	A	C	A	C
8	#	A	A	C	A	C	A	C	C

Rotations de S

$\Rightarrow$

pos									BWT
8	#	A	A	C	A	C	A	C	C
0	A	A	C	A	C	A	C	C	#

Rotations de S triées

Table des suffixes (SA), 1990 et Transformée de Burrows-Wheeler (BWT), 1994

Table des suffixes : Index pour localiser des motifs.
BWT : Permutation réversible utilisée pour la compression et l'indexation.

$$S = \overset{0}{A}\overset{1}{A}\overset{2}{C}\overset{3}{A}\overset{4}{C}\overset{5}{A}\overset{6}{C}\overset{7}{C}\overset{8}{\#}$$

0	A	A	C	A	C	A	C	C	#
1	**A**	**C**	**A**	**C**	**A**	**C**	**C**	**#**	**A**
2	C	A	C	A	C	C	#	A	A
3	A	C	A	C	C	#	A	A	C
4	C	A	C	C	#	A	A	C	A
5	A	C	C	#	A	A	C	A	C
6	C	C	#	A	A	C	A	C	A
7	C	#	A	A	C	A	C	A	C
8	#	A	A	C	A	C	A	C	C

Rotations de S

$\Rightarrow$

pos									BWT
8	#	A	A	C	A	C	A	C	C
0	A	A	C	A	C	A	C	C	#
1	A	C	A	C	A	C	C	#	A

Rotations de S triées

Table des suffixes (SA), 1990 et Transformée de Burrows-Wheeler (BWT), 1994

Table des suffixes : Index pour localiser des motifs.
BWT : Permutation réversible utilisée pour la compression et l'indexation.

$$S = \overset{0}{A}\overset{1}{A}\overset{2}{C}\overset{3}{A}\overset{4}{C}\overset{5}{A}\overset{6}{C}\overset{7}{C}\overset{8}{\#}$$

0	A	A	C	A	C	A	C	C	#
1	A	C	A	C	A	C	C	#	A
2	C	A	C	A	C	C	#	A	A
3	**A**	**C**	**A**	**C**	**C**	#	**A**	**A**	**C**
4	C	A	C	C	#	A	A	C	A
5	A	C	C	#	A	A	C	A	C
6	C	C	#	A	A	C	A	C	A
7	C	#	A	A	C	A	C	A	C
8	#	A	A	C	A	C	A	C	C

Rotations de S

$\Rightarrow$

pos									BWT
8	#	A	A	C	A	C	A	C	C
0	A	A	C	A	C	A	C	C	#
1	A	C	A	C	A	C	C	#	A
3	A	C	A	C	C	#	A	A	C

Rotations de S triées

Table des suffixes (SA), 1990 et Transformée de Burrows-Wheeler (BWT), 1994

Table des suffixes : Index pour localiser des motifs.
BWT : Permutation réversible utilisée pour la compression et l'indexation.

$$S = \overset{0}{A}\overset{1}{A}\overset{2}{C}\overset{3}{A}\overset{4}{C}\overset{5}{A}\overset{6}{C}\overset{7}{C}\overset{8}{\#}$$

0	A	A	C	A	C	A	C	C	#
1	A	C	A	C	A	C	C	#	A
2	C	A	C	A	C	C	#	A	A
3	A	C	A	C	C	#	A	A	C
4	C	A	C	C	#	A	A	C	A
5	**A**	**C**	**C**	#	**A**	**A**	**C**	**A**	**C**
6	C	C	#	A	A	C	A	C	A
7	C	#	A	A	C	A	C	A	C
8	#	A	A	C	A	C	A	C	C

Rotations de S

$\Rightarrow$

pos									BWT
8	#	A	A	C	A	C	A	C	C
0	A	A	C	A	C	A	C	C	#
1	A	C	A	C	A	C	C	#	A
3	A	C	A	C	C	#	A	A	C
5	A	C	C	#	A	A	C	A	C

Rotations de S triées

Table des suffixes (SA), 1990 et Transformée de Burrows-Wheeler (BWT), 1994

Table des suffixes : Index pour localiser des motifs.
BWT : Permutation réversible utilisée pour la compression et l'indexation.

$$S = \overset{0}{A}\overset{1}{A}\overset{2}{C}\overset{3}{A}\overset{4}{C}\overset{5}{A}\overset{6}{C}\overset{7}{C}\overset{8}{\#}$$

0	A	A	C	A	C	A	C	C	#
1	A	C	A	C	A	C	C	#	A
2	C	A	C	A	C	C	#	A	A
3	A	C	A	C	C	#	A	A	C
4	C	A	C	C	#	A	A	C	A
5	A	C	C	#	A	A	C	A	C
6	C	C	#	A	A	C	A	C	A
7	**C**	#	**A**	**A**	**C**	**A**	**C**	**A**	**C**
8	#	A	A	C	A	C	A	C	C

Rotations de S

$\Rightarrow$

pos									BWT
8	#	A	A	C	A	C	A	C	C
0	A	A	C	A	C	A	C	C	#
1	A	C	A	C	A	C	C	#	A
3	A	C	A	C	C	#	A	A	C
5	A	C	C	#	A	A	C	A	C
7	C	#	A	A	C	A	C	A	C

Rotations de S triées

Table des suffixes (SA), 1990 et Transformée de Burrows-Wheeler (BWT), 1994

Table des suffixes : Index pour localiser des motifs.
BWT : Permutation réversible utilisée pour la compression et l'indexation.

$$S = \overset{0}{A}\overset{1}{A}\overset{2}{C}\overset{3}{A}\overset{4}{C}\overset{5}{A}\overset{6}{C}\overset{7}{C}\overset{8}{\#}$$

0	A	A	C	A	C	A	C	C	#
1	A	C	A	C	A	C	C	#	A
2	**C**	**A**	**C**	**A**	**C**	**C**	**#**	**A**	**A**
3	A	C	A	C	C	#	A	A	C
4	C	A	C	C	#	A	A	C	A
5	A	C	C	#	A	A	C	A	C
6	C	C	#	A	A	C	A	C	A
7	C	#	A	A	C	A	C	A	C
8	#	A	A	C	A	C	A	C	C

Rotations de S

$\Rightarrow$

pos									BWT
8	#	A	A	C	A	C	A	C	C
0	A	A	C	A	C	A	C	C	#
1	A	C	A	C	A	C	C	#	A
3	A	C	A	C	C	#	A	A	C
5	A	C	C	#	A	A	C	A	C
7	C	#	A	A	C	A	C	A	C
2	C	A	C	A	C	C	#	A	A

Rotations de S triées

Table des suffixes (SA), 1990 et Transformée de Burrows-Wheeler (BWT), 1994

Table des suffixes : Index pour localiser des motifs.
BWT : Permutation réversible utilisée pour la compression et l'indexation.

$$\overset{0\ 1\ 2\ 3\ 4\ 5\ 6\ 7\ 8}{S = A\,A\,C\,A\,C\,A\,C\,C\,\#}$$

0	A	A	C	A	C	A	C	C	#
1	A	C	A	C	A	C	C	#	A
2	C	A	C	A	C	C	#	A	A
3	A	C	A	C	C	#	A	A	C
4	**C**	**A**	**C**	**C**	**#**	**A**	**A**	**C**	**A**
5	A	C	C	#	A	A	C	A	C
6	C	C	#	A	A	C	A	C	A
7	C	#	A	A	C	A	C	A	C
8	#	A	A	C	A	C	A	C	C

Rotations de S

$\Rightarrow$

pos									BWT
8	#	A	A	C	A	C	A	C	C
0	A	A	C	A	C	A	C	C	#
1	A	C	A	C	A	C	C	#	A
3	A	C	A	C	C	#	A	A	C
5	A	C	C	#	A	A	C	A	C
7	C	#	A	A	C	A	C	A	C
2	C	A	C	A	C	C	#	A	A
4	C	A	C	C	#	A	A	C	A

Rotations de S triées

Table des suffixes (SA), 1990 et Transformée de Burrows-Wheeler (BWT), 1994

Table des suffixes : Index pour localiser des motifs.
BWT : Permutation réversible utilisée pour la compression et l'indexation.

$$S = \overset{0}{A}\overset{1}{A}\overset{2}{C}\overset{3}{A}\overset{4}{C}\overset{5}{A}\overset{6}{C}\overset{7}{C}\overset{8}{\#}$$

0	A	A	C	A	C	A	C	C	#
1	A	C	A	C	A	C	C	#	A
2	C	A	C	A	C	C	#	A	A
3	A	C	A	C	C	#	A	A	C
4	C	A	C	C	#	A	A	C	A
5	A	C	C	#	A	A	C	A	C
6	**C**	**C**	**#**	**A**	**A**	**C**	**A**	**C**	**A**
7	C	#	A	A	C	A	C	A	C
8	#	A	A	C	A	C	A	C	C

Rotations de S

$\Rightarrow$

pos									BWT
8	#	A	A	C	A	C	A	C	C
0	A	A	C	A	C	A	C	C	#
1	A	C	A	C	A	C	C	#	A
3	A	C	A	C	C	#	A	A	C
5	A	C	C	#	A	A	C	A	C
7	C	#	A	A	C	A	C	A	C
2	C	A	C	A	C	C	#	A	A
4	C	A	C	C	#	A	A	C	A
6	C	C	#	A	A	C	A	C	A

Rotations de S triées

Table des suffixes (SA), 1990 et Transformée de Burrows-Wheeler (BWT), 1994

Table des suffixes : Index pour localiser des motifs.
BWT : Permutation réversible utilisée pour la compression et l'indexation.

Si SA[i]>0 alors BWT[i]=S[SA[i]-1], sinon BWT[i]=#

$$\overset{0\ 1\ 2\ 3\ 4\ 5\ 6\ 7\ 8}{S = \mathrm{A\,A\,C\,A\,C\,A\,C\,C\#}}$$

BWT(S)=C#ACCCAAA, and SA(S)=[8,0,1,3,5,7,2,4,6]

0	A	A	C	A	C	A	C	C	#
1	A	C	A	C	A	C	C	#	A
2	C	A	C	A	C	C	#	A	A
3	A	C	A	C	C	#	A	A	C
4	C	A	C	C	#	A	A	C	A
5	A	C	C	#	A	A	C	A	C
6	C	C	#	A	A	C	A	C	A
7	C	#	A	A	C	A	C	A	C
8	#	A	A	C	A	C	A	C	C

Rotations de S

$\Rightarrow$

SA									BWT
8	#	A	A	C	A	C	A	C	C
0	A	A	C	A	C	A	C	C	#
1	A	C	A	C	A	C	C	#	A
3	A	C	A	C	C	#	A	A	C
5	A	C	C	#	A	A	C	A	C
7	C	#	A	A	C	A	C	A	C
2	C	A	C	A	C	C	#	A	A
4	C	A	C	C	#	A	A	C	A
6	C	C	#	A	A	C	A	C	A

Rotations de S triées

La réversibilité de la BWT

	0	1	2	3	4	5	6	7	8
$S=$	A	A	C	A	C	A	C	C	#

SA	P	BWT
8	#	C
0	A	#
1	A	A
3	A	C
5	A	C
7	C	C
2	C	A
4	C	A
6	C	A

La réversibilité de la BWT

$S =$ A A C A C A C C # (positions 0 1 2 3 4 5 6 7 8)

SA	P	BWT
8	#	C
0	A	#
1	A	A
3	A	C
5	A	C
7	C	C
2	C	A
4	C	A
6	C	A

$S =$ ^{8}#

La réversibilité de la BWT

$S=$ A A C A C A C C # (positions 0 1 2 3 4 5 6 7 8)

SA	P	BWT
8	#	C
0	A	#
1	A	A
3	A	C
5	A	C
7	C	C
2	C	A
4	C	A
6	C	A

$S=$ C # (positions 7 8)

La réversibilité de la BWT

$S=$ A A C A C A C C # (positions 0 1 2 3 4 5 6 7 8)

SA	P	BWT
8	#	C
0	A	#
1	A	A
3	A	C
5	A	C
7	C	C
2	C	A
4	C	A
6	C	A

$S=$ C # (positions 7 8)

La réversibilité de la BWT

$S=$ A A C A C A C C # (0 1 2 3 4 5 6 7 8)

SA	P	BWT
8	#	C
0	A	#
1	A	A
3	A	C
5	A	C
7	C	C
2	C	A
4	C	A
6	C	A

$S=$ C C # (6 7 8)

La réversibilité de la BWT

$S=$ A A C A C A C C #
0 1 2 3 4 5 6 7 8

SA	P	BWT
8	#	C
0	A	#
1	A	A
3	A	C
5	A	C
7	C	C
2	C	A
4	C	A
6	C	A

$S=$ C C #
6 7 8

La réversibilité de la BWT

$S=$ A A C A C A C C # (positions 0 1 2 3 4 5 6 7 8)

SA	P	BWT
8	#	C
0	A	#
1	A	A
3	A	C
5	A	C
7	C	C
2	C	A
4	C	A
6	C	A

$S=$ A C C # (positions 5 6 7 8)

La réversibilité de la BWT

$S=$ A A C A C A C C # (positions 0 1 2 3 4 5 6 7 8)

SA	P	BWT
8	#	C
0	A	#
1	A	A
3	A	C
5	A	C
7	C	C
2	C	A
4	C	A
6	C	A

$S=$ A C C # (positions 5 6 7 8)

La réversibilité de la BWT

$S=$ A A C A C A C C # (positions 0 1 2 3 4 5 6 7 8)

SA	P	BWT
8	#	C
0	A	#
1	A	A
3	A	C
5	A	C
7	C	C
2	C	A
4	C	A
6	C	A

$S=$ C A C C # (positions 4 5 6 7 8)

La réversibilité de la BWT

$S=$ A A C A C A C C # (positions 0 1 2 3 4 5 6 7 8)

SA	P	BWT
8	#	C
0	A	#
1	A	A
3	A	C
5	A	C
7	C	C
2	C	A
4	C	A
6	C	A

$S=$ C A C C # (positions 4 5 6 7 8)

La réversibilité de la BWT

$S=$ A A C A C A C C # (0 1 2 3 4 5 6 7 8)

SA	P	BWT
8	#	C
0	A	#
1	A	A
3	A	C
5	A	C
7	C	C
2	C	A
4	C	A
6	C	A

$S=$ A C A C C # (3 4 5 6 7 8)

La réversibilité de la BWT

$S =$ A A C A C A C C # (0 1 2 3 4 5 6 7 8)

SA	P	BWT
8	#	C
0	A	#
1	A	A
3	A	C
5	A	C
7	C	C
2	C	A
4	C	A
6	C	A

$S =$ A C A C C # (3 4 5 6 7 8)

La réversibilité de la BWT

$S =$ A A C A C A C C # (0 1 2 3 4 5 6 7 8)

SA	P	BWT
8	#	C
0	A	#
1	A	A
3	A	C
5	A	C
7	C	C
2	C	A
4	C	A
6	C	A

$S =$ C A C A C C # (2 3 4 5 6 7 8)

La réversibilité de la BWT

$S=$ A A C A C A C C # (0 1 2 3 4 5 6 7 8)

SA	P	BWT
8	#	C
0	A	#
1	A	A
3	A	C
5	A	C
7	C	C
2	C	A
4	C	A
6	C	A

$S=$ C A C A C C # (2 3 4 5 6 7 8)

La réversibilité de la BWT

$S=$ A A C A C A C C # (0 1 2 3 4 5 6 7 8)

SA	P	BWT
8	#	C
0	A	#
1	A	A
3	A	C
5	A	C
7	C	C
2	C	A
4	C	A
6	C	A

$S=$ A C A C A C C # (1 2 3 4 5 6 7 8)

La réversibilité de la BWT

$S=$ A A C A C A C C #
0 1 2 3 4 5 6 7 8

SA	P	BWT
8	#	C
0	A	#
1	A	A
3	A	C
5	A	C
7	C	C
2	C	A
4	C	A
6	C	A

$S=$ A C A C A C C #
1 2 3 4 5 6 7 8

La réversibilité de la BWT

$S=$ A A C A C A C C # (0 1 2 3 4 5 6 7 8)

SA	P	BWT
8	#	C
0	A	#
1	A	A
3	A	C
5	A	C
7	C	C
2	C	A
4	C	A
6	C	A

$S=$ A A C A C A C C # (0 1 2 3 4 5 6 7 8)

La réversibilité de la BWT

$$S = \overset{0}{A}\,\overset{1}{A}\,\overset{2}{C}\,\overset{3}{A}\,\overset{4}{C}\,\overset{5}{A}\,\overset{6}{C}\,\overset{7}{C}\,\overset{8}{\#}$$

SA	P	BWT
8	#	C
0	A	#
1	A	A
3	A	C
5	A	C
7	C	C
2	C	A
4	C	A
6	C	A

$$S = \overset{0}{A}\,\overset{1}{A}\,\overset{2}{C}\,\overset{3}{A}\,\overset{4}{C}\,\overset{5}{A}\,\overset{6}{C}\,\overset{7}{C}\,\overset{8}{\#}$$

La réversibilité de la BWT

$S=$ A A C A C A C C # (0 1 2 3 4 5 6 7 8)

SA	P	BWT
8	#	C
0	A	#
1	A	A
3	A	C
5	A	C
7	C	C
2	C	A
4	C	A
6	C	A

$S=$ A A C A C A C C # (0 1 2 3 4 5 6 7 8)

Suffix Array Manber& Myers 1990 and Burrows-Wheeler Transform 1994

Suffix Array : Index pour localiser des motifs.
BWT : Permutation réversible utilisée pour la compression et l'indexation.
LCP : Le plus long préfixe commun entre deux lignes.

$$S = \overset{0}{A}\overset{1}{A}\overset{2}{C}\overset{3}{A}\overset{4}{C}\overset{5}{A}\overset{6}{C}\overset{7}{C}\overset{8}{\#}$$

0	A	A	C	A	C	A	C	C	#
1	A	C	A	C	A	C	C	#	
2	C	A	C	A	C	C	#		
3	A	C	A	C	C	#			
4	C	A	C	C	#				
5	A	C	C	#					
6	C	C	#						
7	C	#							
8	#								

Suffixes de S

$\Rightarrow$

LCP	SA								BWT
0	8	#							C
0	0	A	A	C	A	C	A	C	C #
1	1	A	C	A	C	A	C	C	# A
4	3	A	C	A	C	C	#		C
2	5	A	C	C	#				C
0	7	C	#						C
1	2	C	A	C	A	C	C	#	A
3	4	C	A	C	C	#			A
1	6	C	C	#					A

Suffixes de S triés

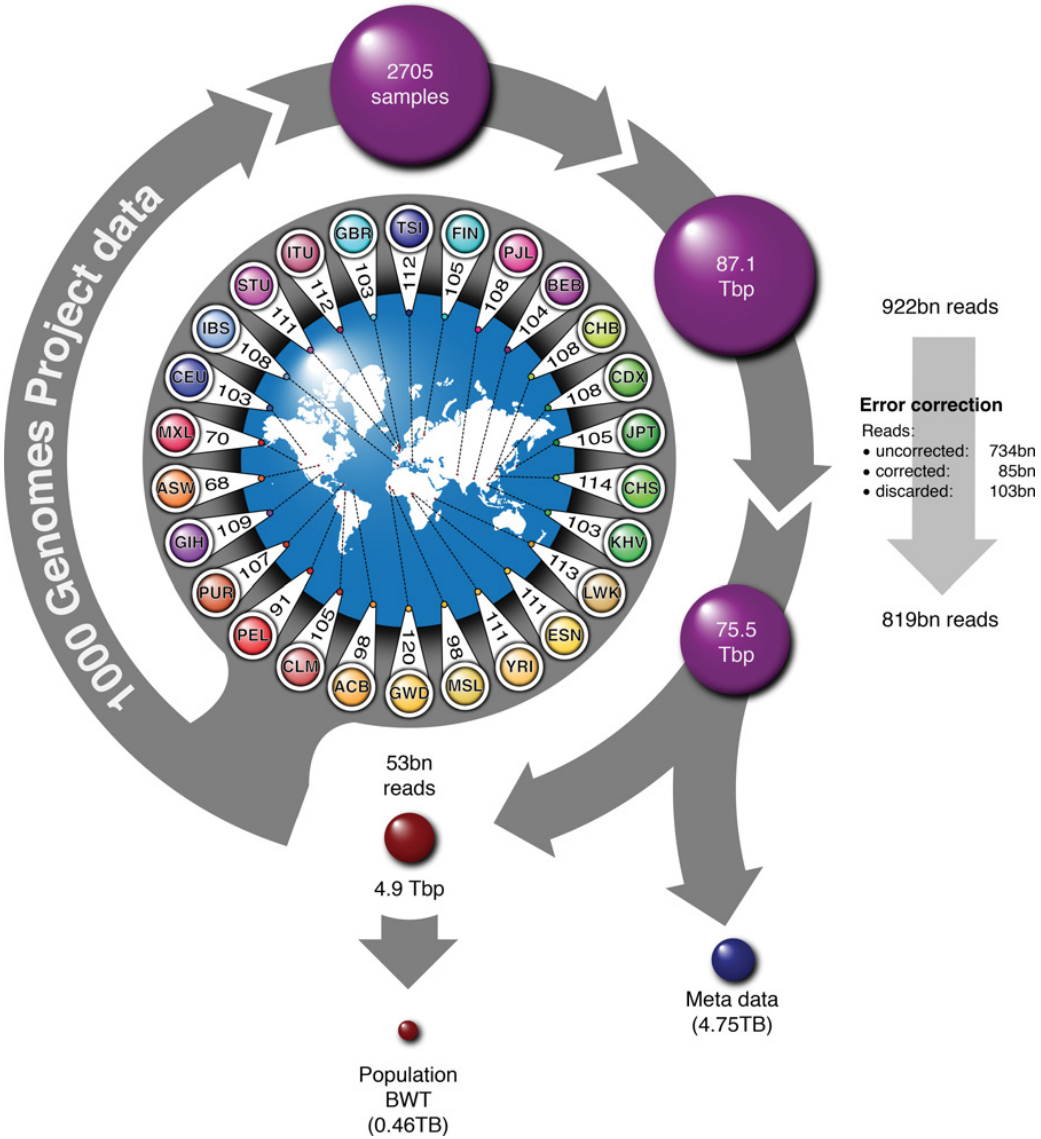
Les applications

- Compression sans perte des données,
- Recherche de motifs en combinant la table des suffixes et la BWT.

Les applications

- Compression sans perte des données,
- Recherche de motifs en combinant la table des suffixes et la BWT.

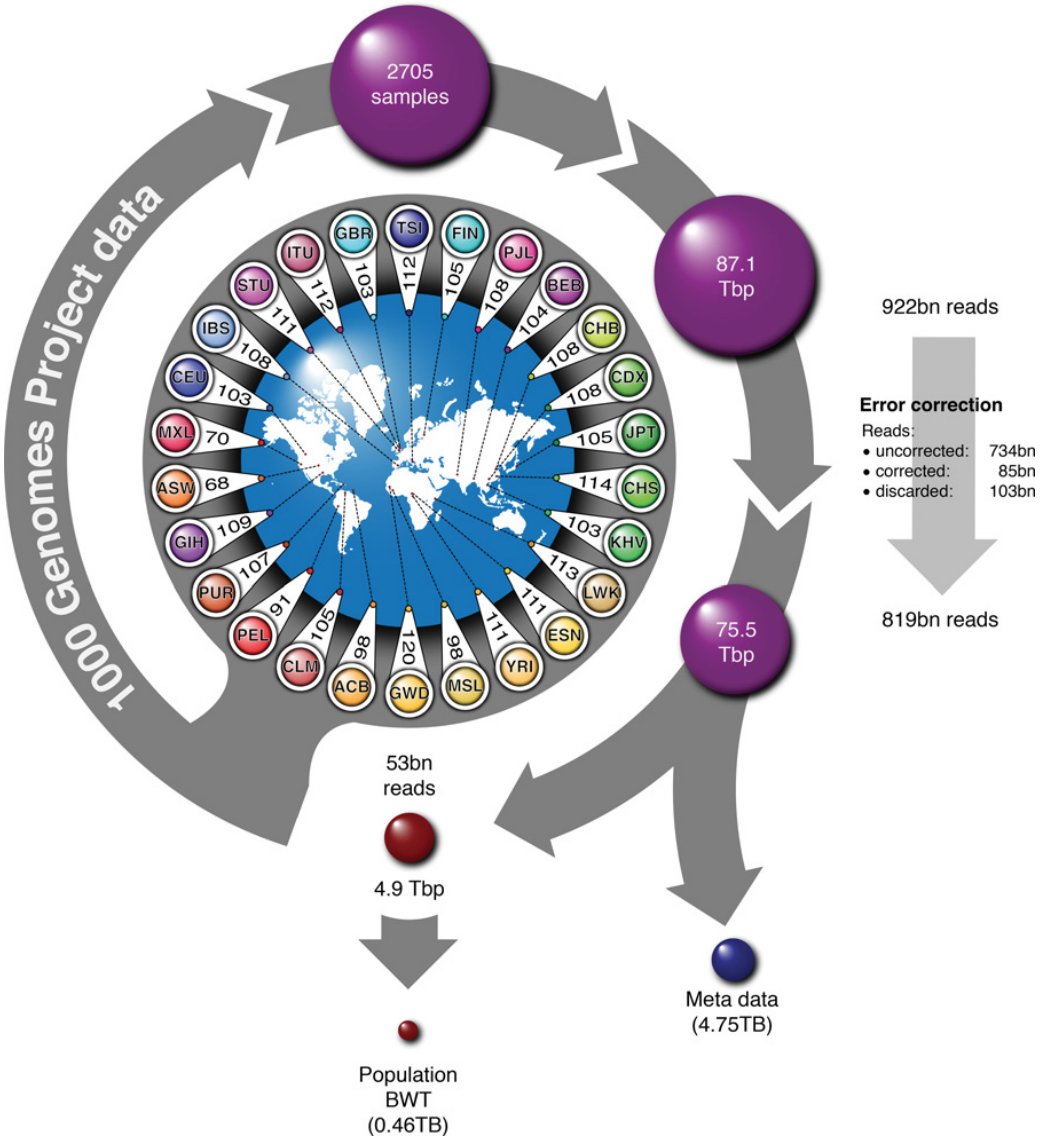
- Alignement des données de séquençage sur le génome, BWA Li and Durbin 2009, Bowtie Langmead et al. 2009 etc ...



The Population BWT, Dolle et al. 2017

Pour chaque lecture ils stockent (4 bytes) :

- Le groupe de la lecture (2 bytes),
- Le nombre de bases corrigées (1 byte),
- Le nombre de bases de faibles qualités (1 byte).



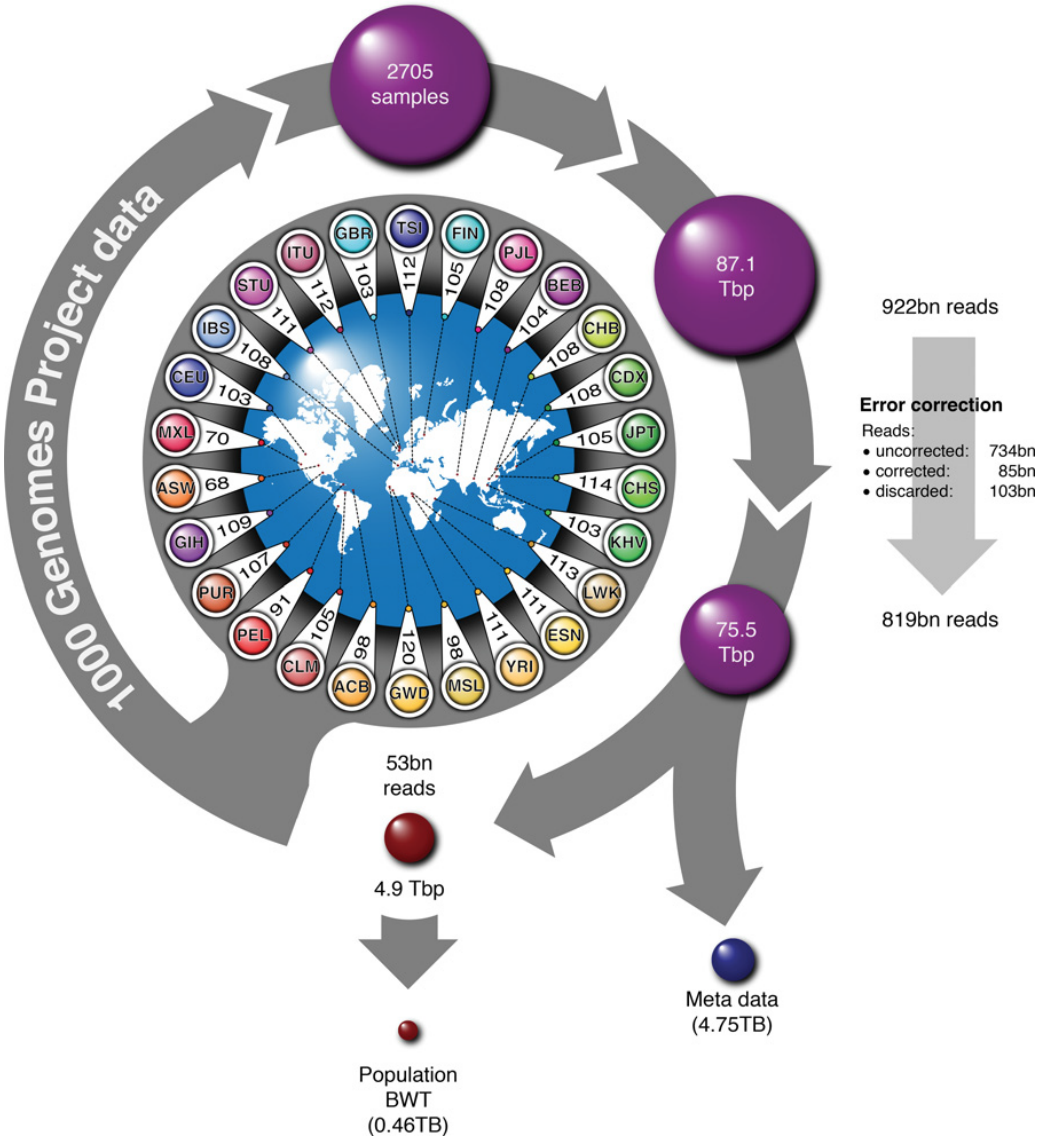
The Population BWT, Dolle et al. 2017

Pour chaque lecture ils stockent (4 bytes) :

- Le groupe de la lecture (2 bytes),
- Le nombre de bases corrigées (1 byte),
- Le nombre de bases de faibles qualités (1 byte).

Notre objectif était de stocker (? bytes) :

- L'identifiant de l'échantillon (1 byte),
- La lecture appariée (?).



Nos résultats :

Il faut changer l'ordre des lectures.

→ Perte d'espace pour la BWT.

→ Gain d'espace important pour les méta-données.

→ Estimation d'une division par deux de l'espace total.

1. Ma structure de données préférée

2. Et si les absents avaient raison ?
   - Definition
   - Les applications
   - Le lien avec les répétitions maximales
   - Les calculs des mots absents minimaux
   - Mots absents minimaux sur une fenêtre glissante

3. Les cercles qui ne rentrent pas dans le moule

Definition : Mot Absent Minimal

Un mot est absent d'une séquence lors qu'il n'y apparait pas.

Definition : Mot Absent Minimal

Un mot est absent d'une séquence lors qu'il n'y apparait pas.
Un mot absent est minimal lorsque tous ses facteurs propres (le plus long préfixe et le plus long suffixe) apparaissent dans la séquence.

Definition : Mot Absent Minimal

Un mot est absent d'une séquence lors qu'il n'y apparait pas.
Un mot absent est minimal lorsque tous ses facteurs propres (le plus long préfixe et le plus long suffixe) apparaissent dans la séquence.

La borne haute pour le nombre de mots absents minimaux d'une séquence de taille n est $\mathcal{O}(n)$.

Crochemore et al. 1998, Mignosi et al. 2002

$$\overset{0}{A}\overset{1}{A}\overset{2}{C}\overset{3}{A}\overset{4}{C}\overset{5}{A}\overset{6}{C}\overset{7}{C}$$

AAA, AACACC, AACC, CAA, CACACA, CCA, CCC

Definition : Mot Absent Minimal

Un mot est absent d'une séquence lors qu'il n'y apparait pas.
Un mot absent est minimal lorsque tous ses facteurs propres (le plus long préfixe et le plus long suffixe) apparaissent dans la séquence.

La borne haute pour le nombre de mots absents minimaux d'une séquence de taille n est $\mathcal{O}(n)$.

Crochemore et al. 1998, Mignosi et al. 2002

$$\overset{0}{S}\overset{1}{=}\overset{2}{A}\overset{3}{A}\overset{4}{C}\overset{5}{A}\overset{6}{C}\overset{7}{A}\overset{8}{C}\overset{9}{C}$$

AAA, AACACC, AACC, CAA, CACACA, CCA, CCC

Definition : Mot Absent Minimal

Un mot est absent d'une séquence lors qu'il n'y apparait pas.
Un mot absent est minimal lorsque tous ses facteurs propres (le plus long préfixe et le plus long suffixe) apparaissent dans la séquence.

La borne haute pour le nombre de mots absents minimaux d'une séquence de taille n est $\mathcal{O}(n)$.

Crochemore et al. 1998, Mignosi et al. 2002

$$\overset{0}{S}\overset{1}{=}\overset{2}{A}\overset{3}{A}\overset{4}{C}\overset{5}{A}\overset{6}{C}\overset{7}{A}\overset{8}{C}\overset{9}{C}$$

AAA, AACACC, AACC, CAA, CACACA, CCA, CCC

Definition : Mot Absent Minimal

Un mot est absent d'une séquence lors qu'il n'y apparait pas.
Un mot absent est minimal lorsque tous ses facteurs propres (le plus long préfixe et le plus long suffixe) apparaissent dans la séquence.

La borne haute pour le nombre de mots absents minimaux d'une séquence de taille n est $\mathcal{O}(n)$.

Crochemore et al. 1998, Mignosi et al. 2002

$$\overset{0}{A}\overset{1}{A}\overset{2}{C}\overset{3}{A}\overset{4}{C}\overset{5}{A}\overset{6}{C}\overset{7}{C}$$

AAA, AACACC, AACC, CAA, CACACA, CCA, CCC

Definition : Mot Absent Minimal

Un mot est absent d'une séquence lors qu'il n'y apparait pas.
Un mot absent est minimal lorsque tous ses facteurs propres (le plus long préfixe et le plus long suffixe) apparaissent dans la séquence.

La borne haute pour le nombre de mots absents minimaux d'une séquence de taille n est $\mathcal{O}(n)$.

Crochemore et al. 1998, Mignosi et al. 2002

$$\overset{0}{A}\overset{1}{A}\overset{2}{C}\overset{3}{A}\overset{4}{C}\overset{5}{A}\overset{6}{C}\overset{7}{C}$$

S=AACACACC

AAA, AACACC, AACC, CAA, CACACA, CCA, CCC

Les applications

Biologie

- 3 séquences (TTTCGCCCGACT, TACGCCCTATCG, CCTACGCGCAAA), retrouvées dans des régions codantes de souches d'Ebola, sont absentes du génome humain (Silva et al. 2015).

Les applications

Biologie

- 3 séquences (TTTCGCCCGACT, TACGCCCTATCG, CCTACGCGCAAA), retrouvées dans des régions codantes de souches d'Ebola, sont absentes du génome humain (Silva et al. 2015).

BioInformatique

- Distance basée sur les mots absents minimaux
  $\rightarrow$ Phylogénie (Chairungsee et al., 2012, Crochemore et al, 2016).

Les applications

Biologie

- 3 séquences (TTTCGCCCGACT, TACGCCCTATCG, CCTACGCGCAAA), retrouvées dans des régions codantes de souches d'Ebola, sont absentes du génome humain (Silva et al. 2015).

BioInformatique

- Distance basée sur les mots absents minimaux
  $\rightarrow$ Phylogénie (Chairungsee et al., 2012, Crochemore et al, 2016).

Informatique

- Compression des donnés avec les anti-dictionnaires (Crochemore et al., 2000, Fiala and Holub, 2008).

Définition : Paire de répétitions maximale

Une paire de répétitions maximale dans S est un triplet (i, j, w) tq :

- w apparait dans S aux positions i et j,
- $S[i-1] \neq S[j-1]$,
- $S[i+|w|] \neq S[j+|w|]$.

Définition : Paire de répétitions maximale

Une paire de répétitions maximale dans S est un triplet (i, j, w) tq :

- w apparait dans S aux positions i et j,
- $S[i-1] \neq S[j-1]$,
- $S[i+|w|] \neq S[j+|w|]$.

Lemme

Si awb est un mot absent minimal de S, alors il existe des positions i et j telles que (i, j, w) est une paire de répétitions maximale de S.

Définition : Paire de répétitions maximale

Une paire de répétitions maximale dans S est un triplet (i, j, w) tq :

- w apparait dans S aux positions i et j,
- $S[i-1] \neq S[j-1]$,
- $S[i+|w|] \neq S[j+|w|]$.

Lemme

Si awb est un mot absent minimal de S, alors il existe des positions i et j telles que (i, j, w) est une paire de répétitions maximale de S.

Séquence S

A un mot absent minimal de S

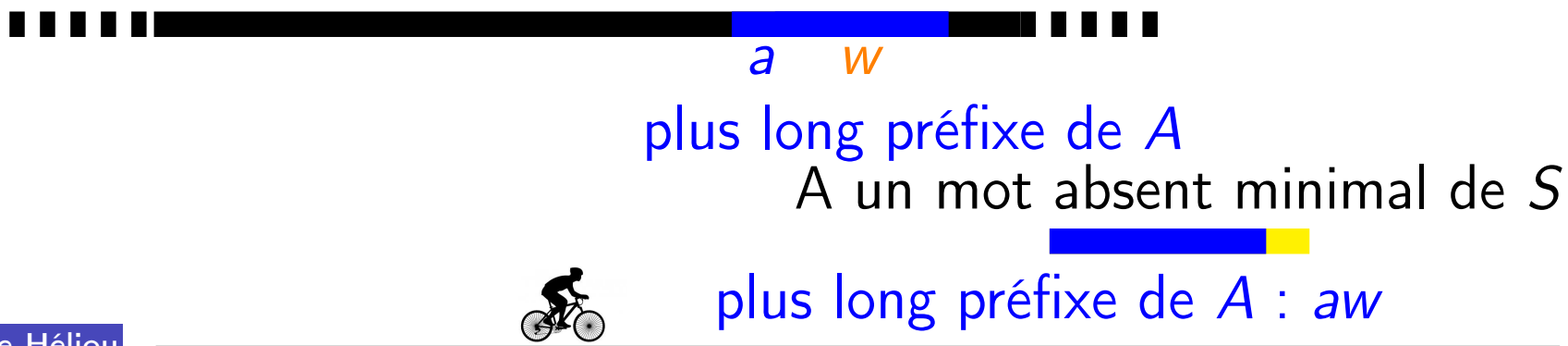
Définition : Paire de répétitions maximale

Une paire de répétitions maximale dans S est un triplet (i, j, w) tq :

- w apparait dans S aux positions i et j,
- $S[i-1] \neq S[j-1]$,
- $S[i+|w|] \neq S[j+|w|]$.

Lemme

Si awb est un mot absent minimal de S, alors il existe des positions i et j telles que (i, j, w) est une paire de répétitions maximale de S.



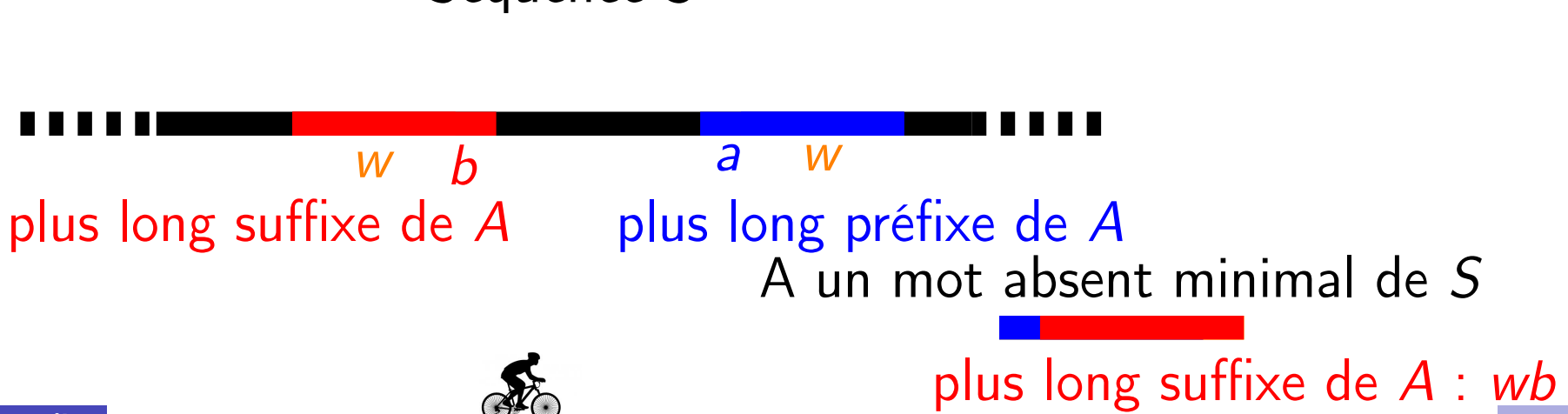
Définition : Paire de répétitions maximale

Une paire de répétitions maximale dans S est un triplet (i, j, w) tq :

- w apparait dans S aux positions i et j,
- $S[i-1] \neq S[j-1]$,
- $S[i+|w|] \neq S[j+|w|]$.

Lemme

Si awb est un mot absent minimal de S, alors il existe des positions i et j telles que (i, j, w) est une paire de répétitions maximale de S.



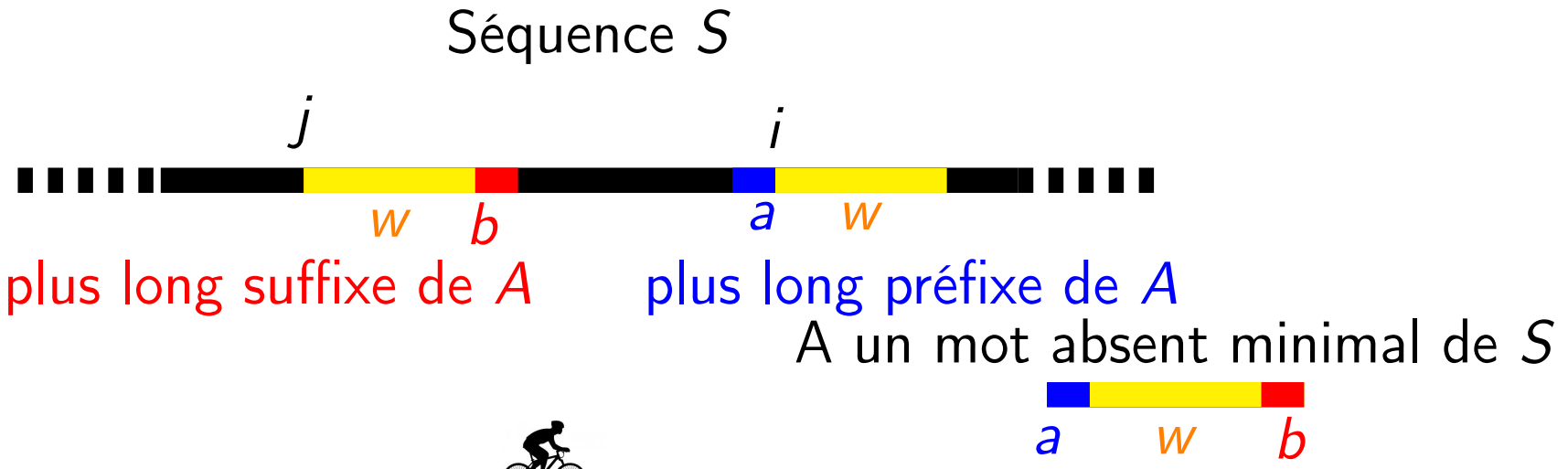
Définition : Paire de répétitions maximale

Une paire de répétitions maximale dans S est un triplet (i, j, w) tq :

- w apparait dans S aux positions i et j,
- $S[i-1] \neq S[j-1]$,
- $S[i+|w|] \neq S[j+|w|]$.

Lemme

Si awb est un mot absent minimal de S, alors il existe des positions i et j telles que (i, j, w) est une paire de répétitions maximale de S.



$$S = \overset{0}{A}\overset{1}{A}\overset{2}{C}\overset{3}{A}\overset{4}{C}\overset{5}{A}\overset{6}{C}\overset{7}{C}\overset{8}{\#}$$

LCP	SA	BWT								
0	8	C	#							
0	0	#	A	A	C	A	C	A	C	C
1	1	A	A	C	A	C	A	C	C	#
4	3	C	A	C	A	C	C	#		
2	5	C	A	C	C	#				
0	7	C	C	#						
1	2	A	C	A	C	A	C	C	#	
3	4	A	C	A	C	C	#			
1	6	A	C	C	#					

$$S = \overset{0}{A}\overset{1}{A}\overset{2}{C}\overset{3}{A}\overset{4}{C}\overset{5}{A}\overset{6}{C}\overset{7}{C}\overset{8}{\#}$$

LCP	SA	BWT	
0	8	C	#
0	0	#	A A C A C A C C
1	1	A	A C A C A C C #
4	3	C	A C A C C #
2	5	C	A C C #
0	7	C	C #
1	2	A	C A C A C C #
3	4	A	C A C C #
1	6	A	C C #

Les autres algorithmes de calcul des mots absents minimaux

Références	Structures	Inconvénients
Crochemore et al., 1998	Automate des suffixes	Trop couteux en espace
Belazzougui et al. 2013	BWT compact bidirectionelle	Non implémenté
Ota et al. 2014	Arbre des suffixes, approche dynamique	Quadratique en temps
Belazzougui et al. 2015	BWT & structures supplémentaires	Non implémenté

Linear-time computation of minimal absent words using suffix array. BMC Bioinformatics, 2014

Pré-calcul

- Table des suffixes, temps et espace linéaires 2003,
- Table des LCP, temps et espace linéaires avec la table des suffixes et la séquence en entrée.

Linear-time computation of minimal absent words using suffix array. BMC Bioinformatics, 2014

Pré-calcul

- Table des suffixes, temps et espace linéaires 2003,
- Table des LCP, temps et espace linéaires avec la table des suffixes et la séquence en entrée.

Calcul

- On traverse deux fois ces tables, pour construire l'ensemble de lettres qui apparaissent avant chaque répétition maximale à droite.
- On en déduit l'ensemble des mots absents minimaux.

Linear-time computation of minimal absent words using suffix array. BMC Bioinformatics, 2014

Pré-calcul

- Table des suffixes, temps et espace linéaires 2003,
- Table des LCP, temps et espace linéaires avec la table des suffixes et la séquence en entrée.

Calcul

- On traverse deux fois ces tables, pour construire l'ensemble de lettres qui apparaissent avant chaque répétition maximale à droite.
- On en déduit l'ensemble des mots absents minimaux.

Performances

Pour l'ensemble du génome humain :
$\simeq$ 9000s avec 130GB de mémoire interne.

$$S = \overset{0}{A}\overset{1}{A}\overset{2}{C}\overset{3}{A}\overset{4}{C}\overset{5}{A}\overset{6}{C}\overset{7}{C}\overset{8}{\#}$$

LCP	SA	BWT	
0	8	C	#
0	0	#	A A C A C A C C
1	1	A	A C A C A C C #
4	3	C	A C A C C #
2	5	C	A C C #
0	7	C	C #
1	2	A	C A C A C C #
3	4	A	C A C C #
1	6	A	C C #

Améliorations et compromis des performances de calcul

Parallelising the Computation of Minimal Absent Words, PPAM 2105

- Parallélisation efficace,
- Calcul pour le génome humain : $\simeq$ 5000s avec 4 coeurs et 130GB de mémoire interne.

Améliorations et compromis des performances de calcul

Parallelising the Computation of Minimal Absent Words, PPAM 2105

- Parallélisation efficace,
- Calcul pour le génome humain : $\simeq$ 5000s avec 4 coeurs et 130GB de mémoire interne.

Computing Minimal Absent Words in External Memory, Bioinformatics 2017

Calcul pour le génome humain :

- $\simeq$ 8000s avec 8GB de mémoire interne,
- $\simeq$ 12000s avec 1GB de mémoire interne.

Améliorations et compromis des performances de calcul

Parallelising the Computation of Minimal Absent Words, PPAM 2105

- Parallélisation efficace,
- Calcul pour le génome humain : $\simeq$ 5000s avec 4 coeurs et 130GB de mémoire interne.

Computing Minimal Absent Words in External Memory, Bioinformatics 2017

Calcul pour le génome humain :

- $\simeq$ 8000s avec 8GB de mémoire interne,
- $\simeq$ 12000s avec 1GB de mémoire interne.

Les implémentations sont disponibles là :
`https://github.com/solonas13/maw`

Calcul des mots absents minimaux sur une fenêtre glissante

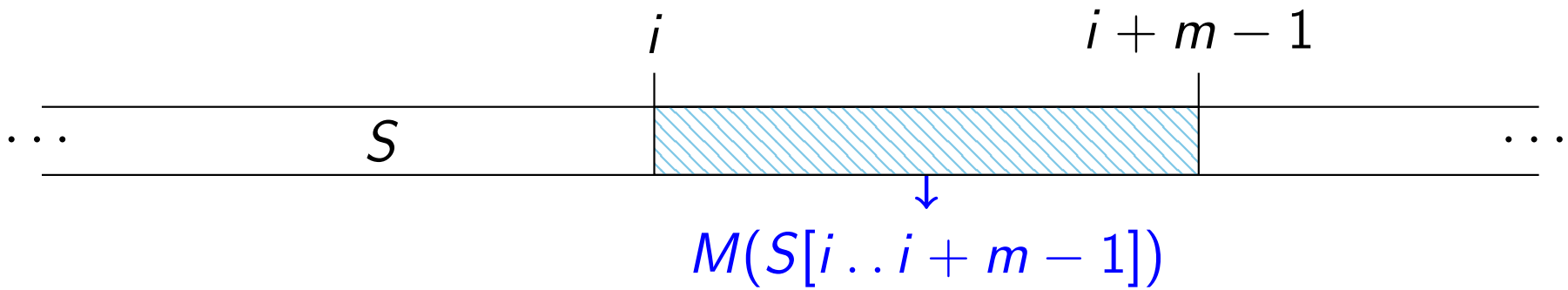
- Une séquence S de taille n, sur un alphabet de taille constante,
- Une fenêtre glissante de taille m sur S : $S[i \,..\, i + m - 1]$.

Pour tout mot x on note $M(x)$ l'ensemble de ses mots absents minimaux.

Calcul des mots absents minimaux sur une fenêtre glissante

- Une séquence S de taille n, sur un alphabet de taille constante,
- Une fenêtre glissante de taille m sur S : $S[i \,..\, i+m-1]$.

Pour tout mot x on note $M(x)$ l'ensemble de ses mots absents minimaux.



Application à la recherche de motifs

Mots absents minimaux dans une fenêtre glissante

Pour une sequence S de taille n et une fenêtre de taille m on calcule :
$\forall i, 0 \leq i \leq n - m, M(S[i \,..\, i + m - 1])$,
en temps $\mathcal{O}(n)$ et en espace $\mathcal{O}(m)$

Application à la recherche de motifs

Mots absents minimaux dans une fenêtre glissante

Pour une sequence S de taille n et une fenêtre de taille m on calcule :
$\forall i, 0 \leq i \leq n - m, M(S[i \,..\, i + m - 1])$,
en temps $\mathcal{O}(n)$ et en espace $\mathcal{O}(m)$

Length Weighted Index (LWI), introduit par Chairungsee en 2012

Mesure basée sur la différence symétrique des ensembles de mots absents minimaux.

$$\mathrm{LWI}(M(x), M(y)) = \sum_{w \in M(x) \triangle M(y)} \frac{1}{|w|^2}.$$

Application à la recherche de motifs

Mots absents minimaux dans une fenêtre glissante

Pour une sequence S de taille n et une fenêtre de taille m on calcule :
$\forall i, 0 \leq i \leq n - m, M(S[i \,..\, i + m - 1])$,
en temps $\mathcal{O}(n)$ et en espace $\mathcal{O}(m)$

Length Weighted Index (LWI), introduit par Chairungsee en 2012

Mesure basée sur la différence symétrique des ensembles de mots absents minimaux.

$$\mathrm{LWI}(M(x), M(y)) = \sum_{w \in M(x) \triangle M(y)} \frac{1}{|w|^2}.$$

$\rightarrow$ On obtient la position de distance minimale.

Algorithmes de calcul des mots absents minimaux

Avec C. Barton, L. Mouchard, S. P. Pissis et S. Puglisi :

- Calcul en temps et espace linéaires avec la table des suffixes, BMC Bioinformatics 2014,
- Calcul parallèle, PPAM 2015,
- Calcul en mémoire externe, Bioinformatics, 2017.

Avec M. Crochemore, G. Kucherov, L. Mouchard, S. P. Pissis et Y. Ramusat :

- Calcul sur une fenêtre glissante et recherche de motifs, FCT 2017.

1. Ma structure de données préférée

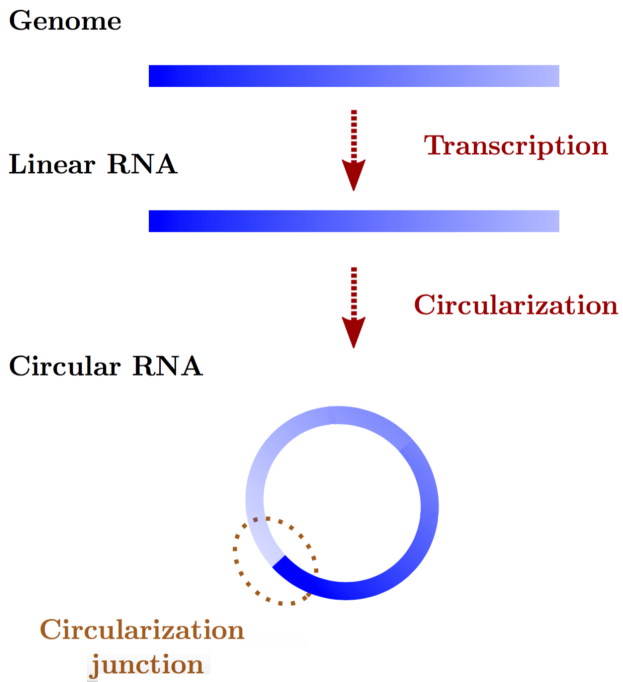
2. Et si les absents avaient raison ?

3. Les cercles qui ne rentrent pas dans le moule
   - Les ARNs circulaires
   - L'alignment des lectures provenant d'ARNs circulaires
   - Le cas de Pyrococcus Abyssi
   - L'analyse d'autres organismes

Les ARNs circulaires

Caractéristiques

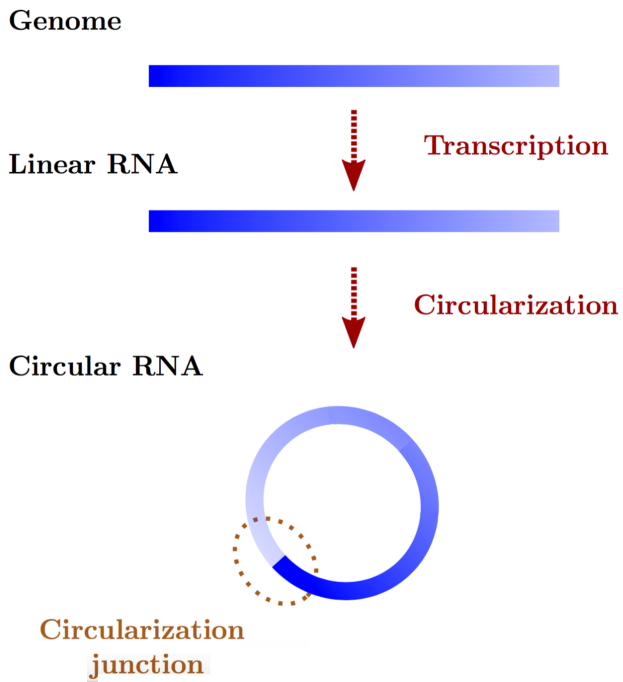
- Pas d'extrémités $\rightarrow$ stabilité accrue,
- Découverts dans tous les domaines du vivant,
- Très étudiés grâce au séquençage haut débit.



Les ARNs circulaires

Caractéristiques

- Pas d'extrémités → stabilité accrue,
- Découverts dans tous les domaines du vivant,
- Très étudiés grâce au séquençage haut débit.



Fonctions

- Modifications post-transcriptionnelles pour la maturation d'ARNs,
- Éponges à ARNmi → régulation de l'expression des gènes,
- ...

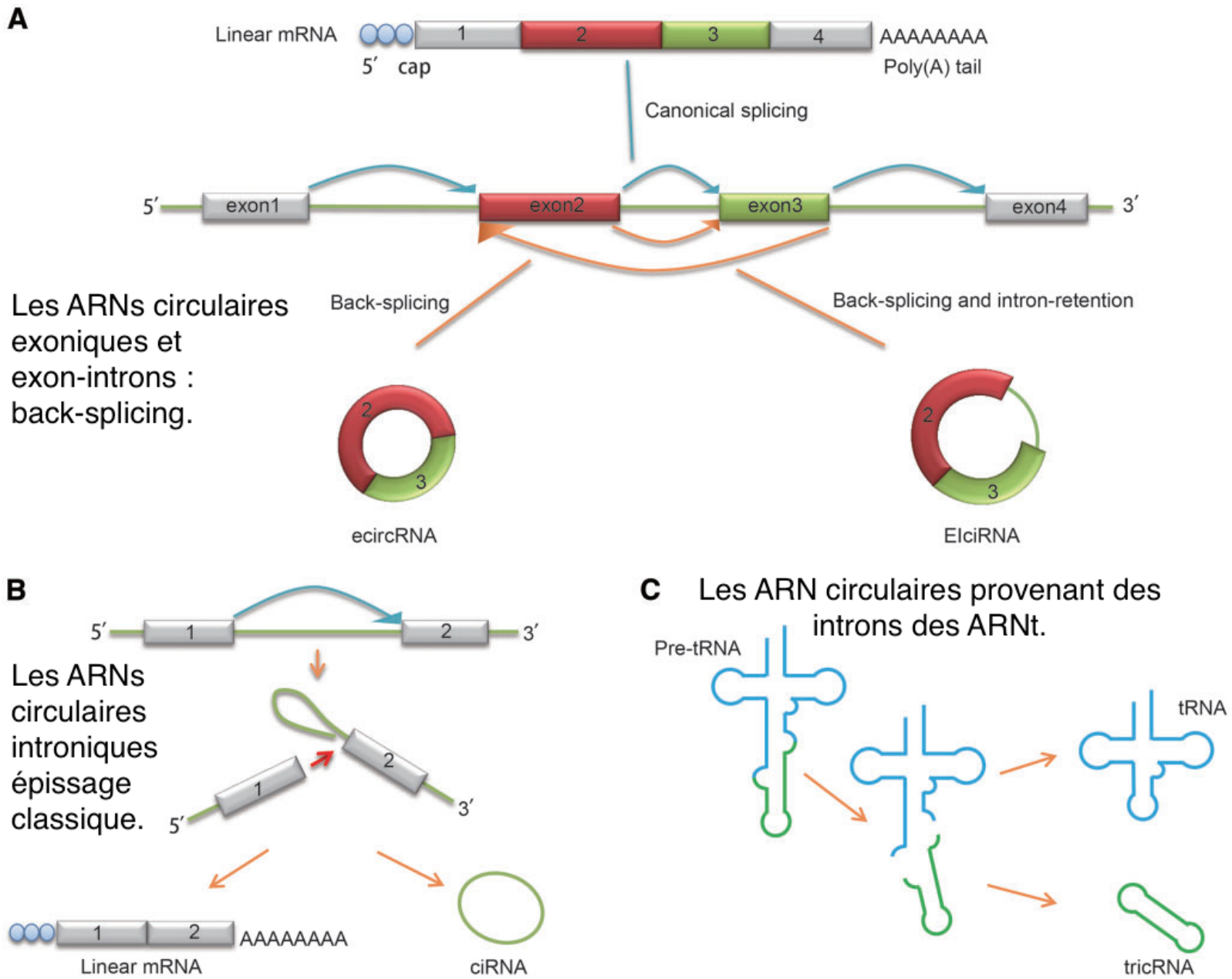
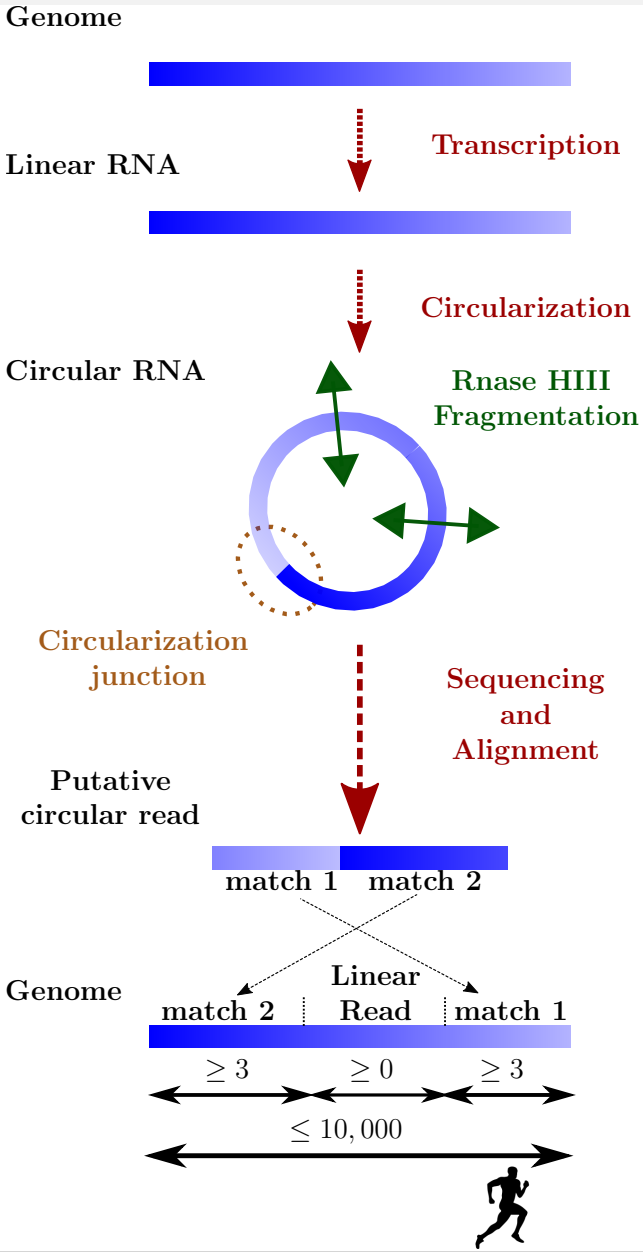
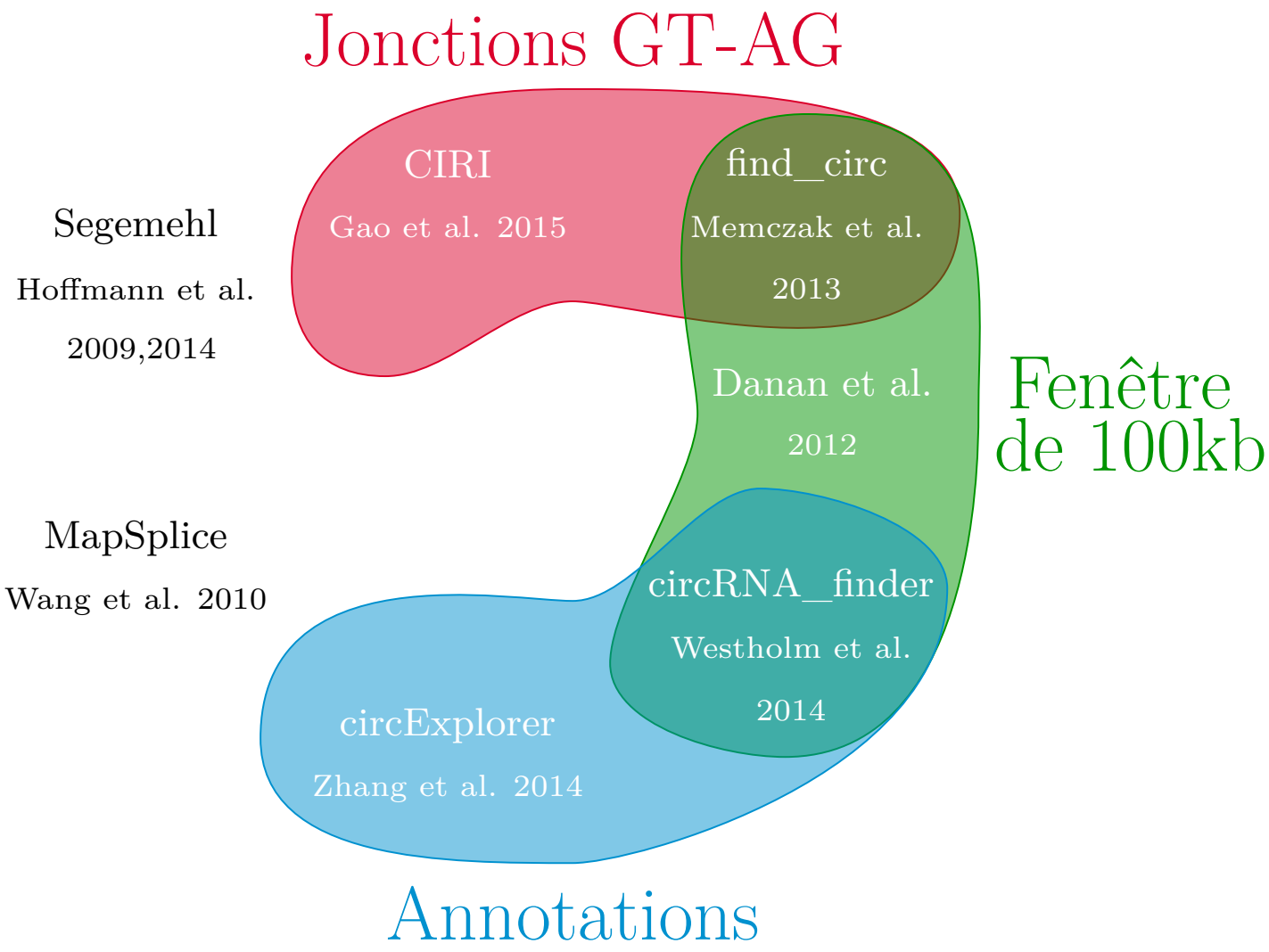


Figure adaptée de Meng et al. Briefings in Bioinformatics, 2016

L'alignement des lectures provenant d'ARNs circulaires



Les critères de sélection



Le cas de Pyrococcus Abyssi

Figure: PDB de PAB1020

L'équipe de H.Myllykallio a montré que l'enzyme PAB1020 est capable de circulariser :

- Des petits ARNs synthétiques *in vitro* (Brooks et al., 2009),
- Trois ARNs *in vivo* (ARNr 5S, deux ARN à boites C/D).

Le cas de Pyrococcus Abyssi

L'équipe de H.Myllykallio a montré que l'enzyme PAB1020 est capable de circulariser :

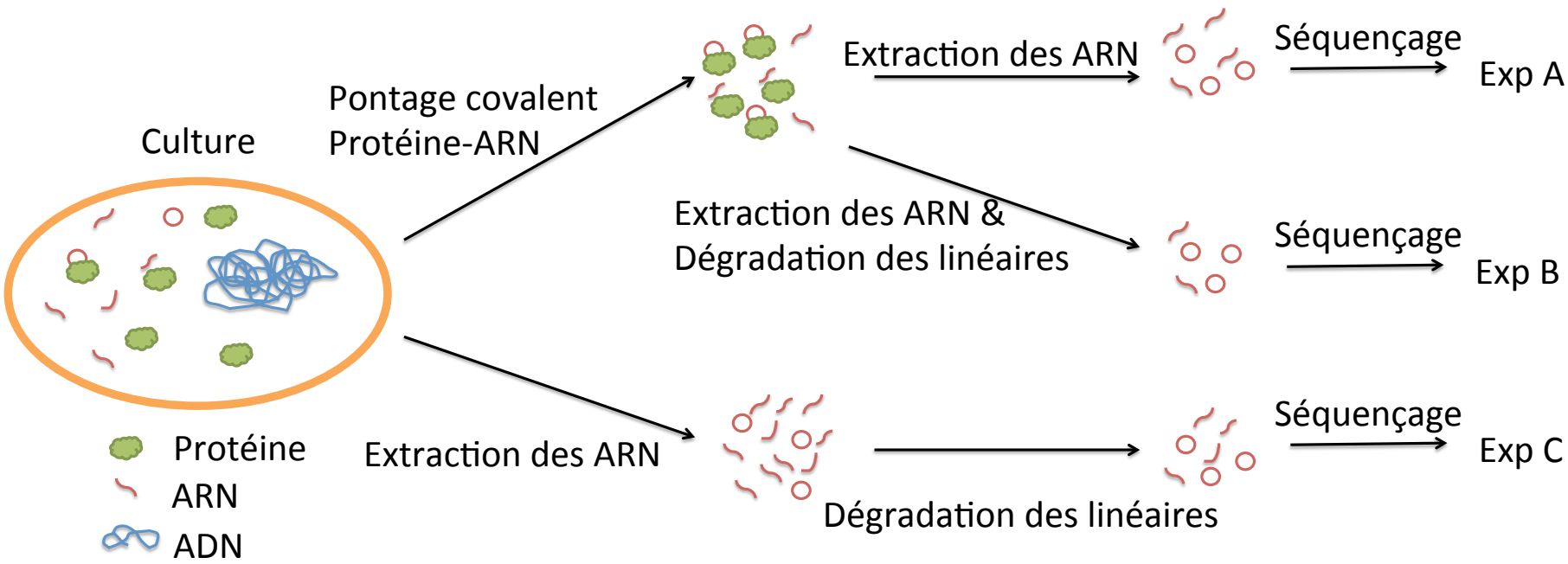
- Des petits ARNs synthétiques *in vitro* (Brooks et al., 2009),
- Trois ARNs *in vivo* (ARNr 5S, deux ARN à boites C/D).

Figure: PDB de PAB1020

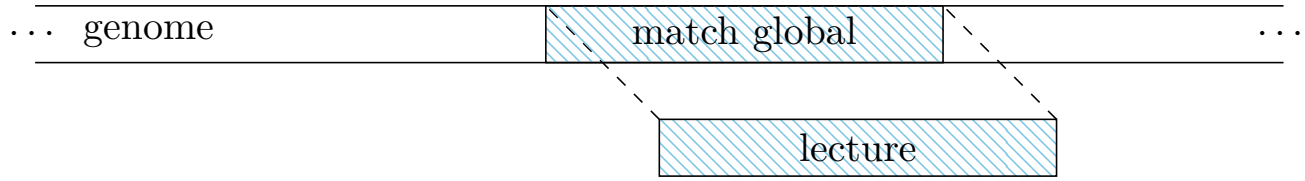
Objectifs des analyses des données de séquençage :

- Confirmer l'implication de la ligase dans la circularisation,
- Identifier les ARNs circulaires.

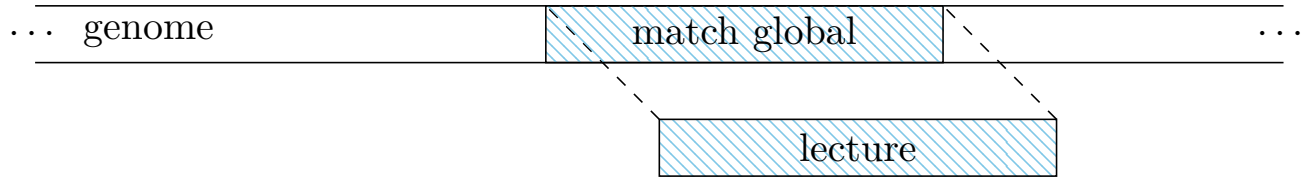
Les expériences de sequençage



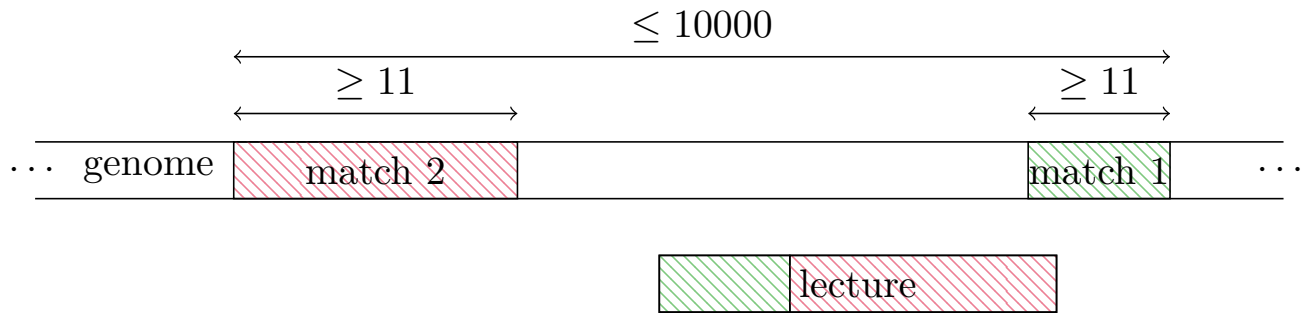
cas 1 : la lecture s'aligne linéairement



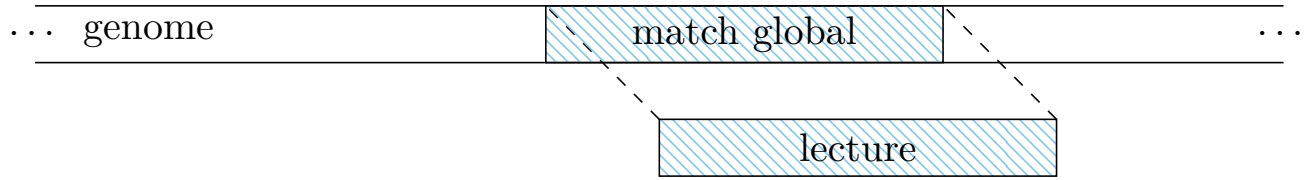
cas 1 : la lecture s'aligne linéairement



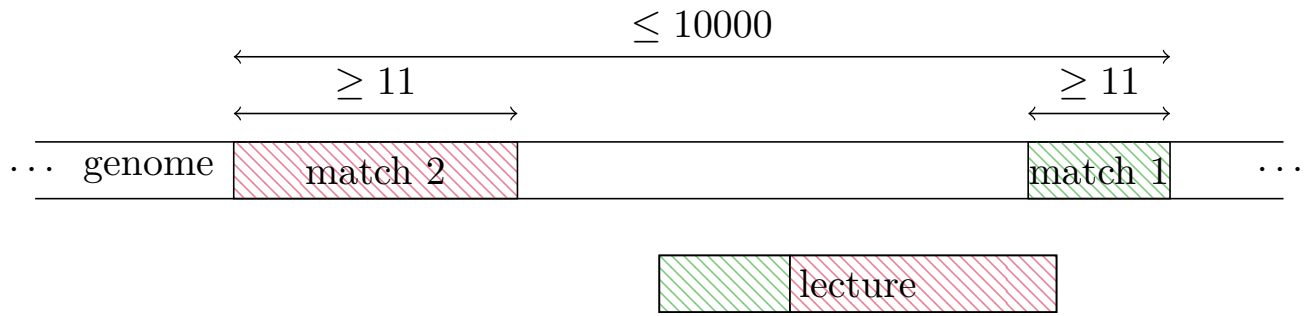
cas 2 : lecture circulaire avec deux matchs inversés



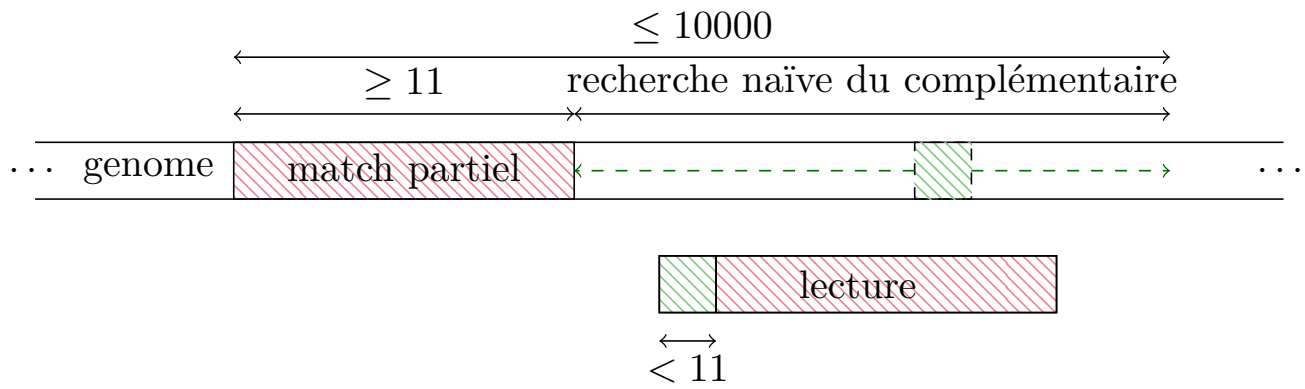
cas 1 : la lecture s'aligne linéairement



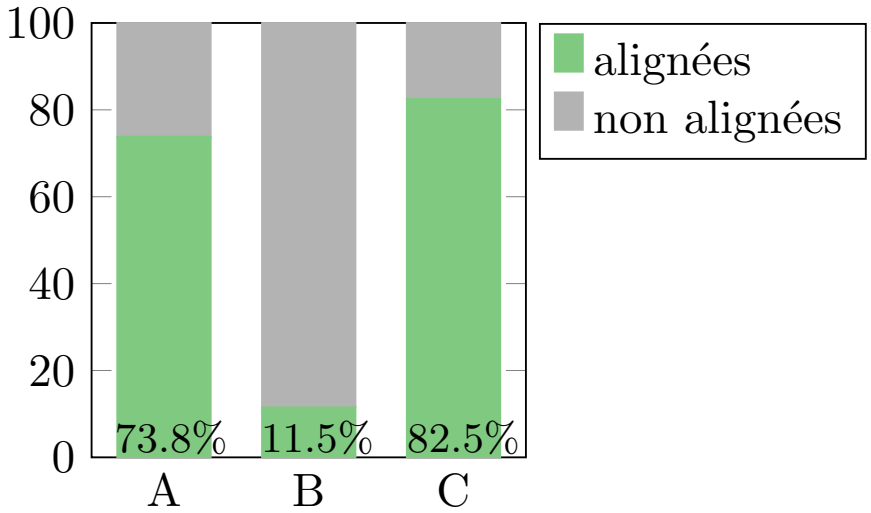
cas 2 : lecture circulaire avec deux matchs inversés



cas 3 : lecture avec un long match incomplet

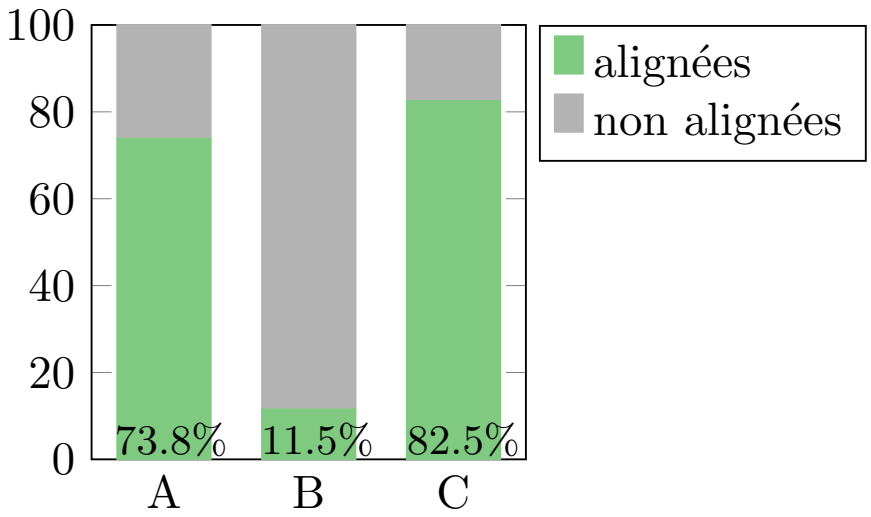


Ratio lectures alignées et non alignées



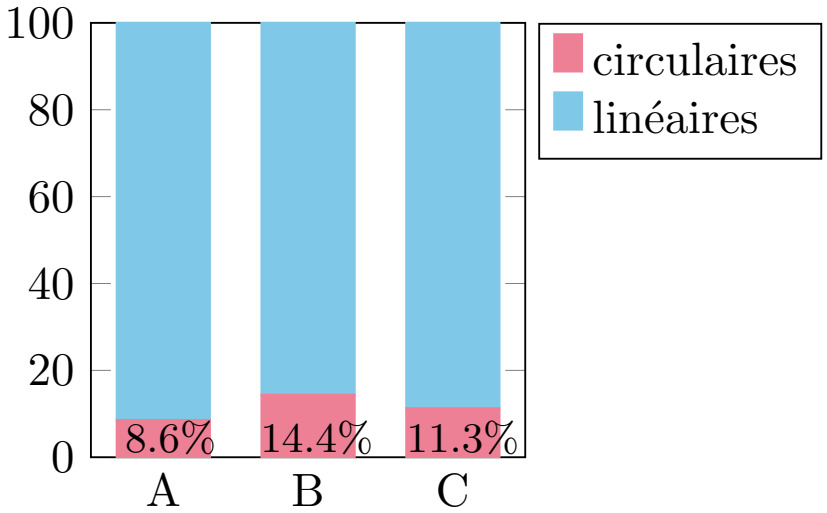
A) Ligase-ARN
B) Ligase-ARN sans linéaires
C) ARN totaux sans linéaires

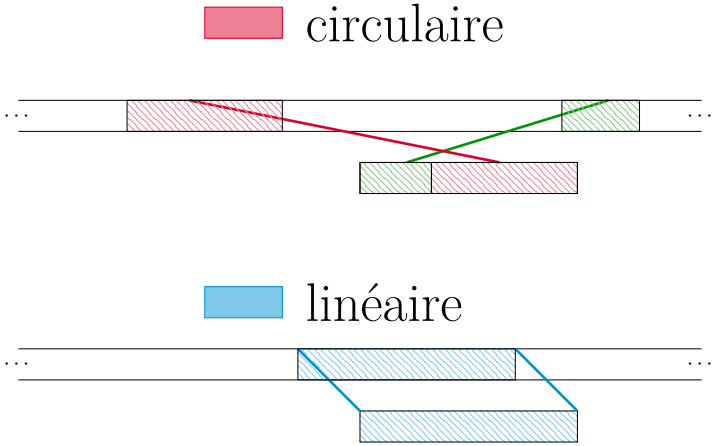
Ratio lectures alignées et non alignées



A) Ligase-ARN
B) Ligase-ARN sans linéaires
C) ARN totaux sans linéaires

Ratio circulaires et linéaires parmi les lectures alignées





Identification des jonctions

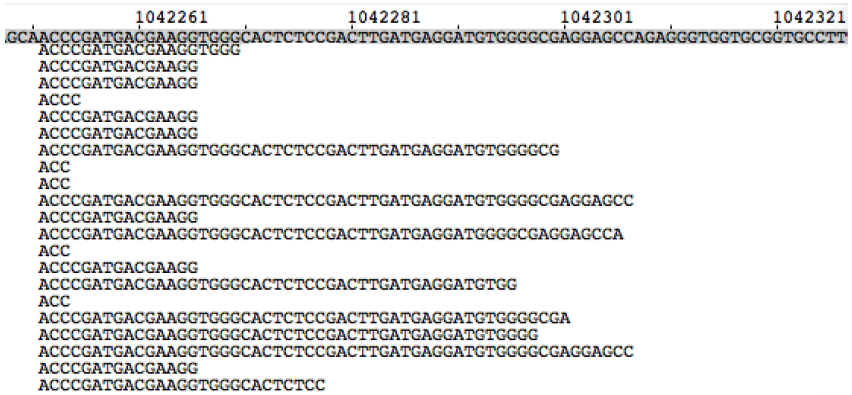
Visualisation de l'alignement des lectures circulaires sur l'ARN à boite C/D sR53

On regroupe les expériences. Une jonction doit être soutenue par :

- au moins 3 lectures venant d'au moins 2 expériences,

Identification des jonctions

Uniquement les premiers matchs
$\rightarrow$ ils commencent tous à la même position

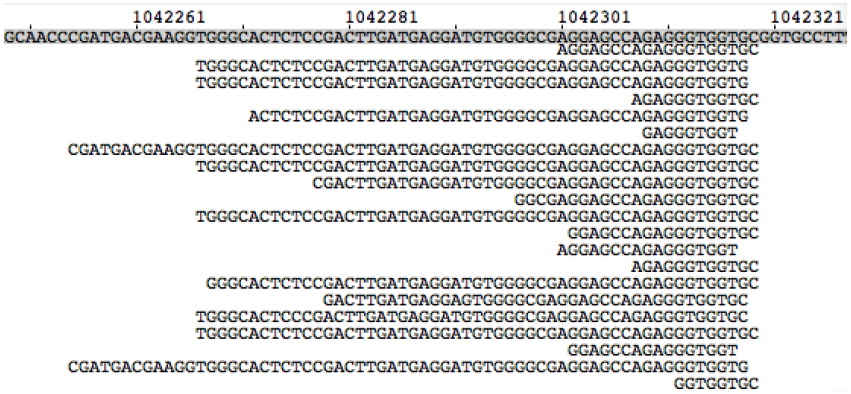


On regroupe les expériences. Une jonction doit être soutenue par :

- au moins 3 lectures venant d'au moins 2 expériences,

Identification des jonctions

Uniquement les seconds matchs $\rightarrow$ ils finissent tous à la même position (à 3 nucléotides près)



On regroupe les expériences. Une jonction doit être soutenue par :

- au moins 3 lectures venant d'au moins 2 expériences,

Identification des jonctions

Jonction sur l'ARN à boite C/D sR53

On regroupe les expériences. Une jonction doit être soutenue par :

- au moins 3 lectures venant d'au moins 2 expériences,

Identification des jonctions

Jonction sur l'ARN à boite C/D sR53

On regroupe les expériences. Une jonction doit être soutenue par :

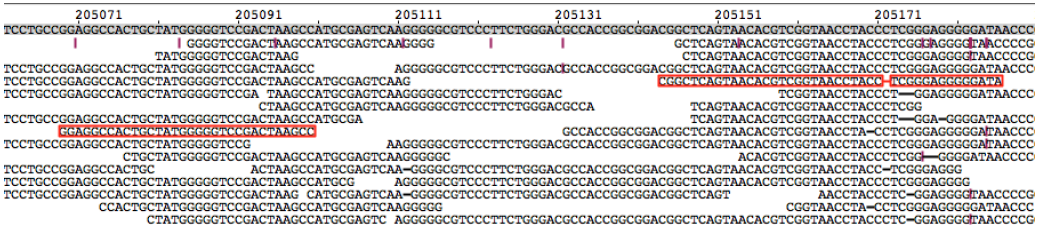
- au moins 3 lectures venant d'au moins 2 expériences,
- au moins la moitié des lectures circulaires dans la jonction.

Identification des jonctions

Jonction sur l'ARN à boite C/D sR53

Visualisation de l'alignement des lectures circulaires sur l'ARNr 16S

$\rightarrow$ pas de jonction identifiable

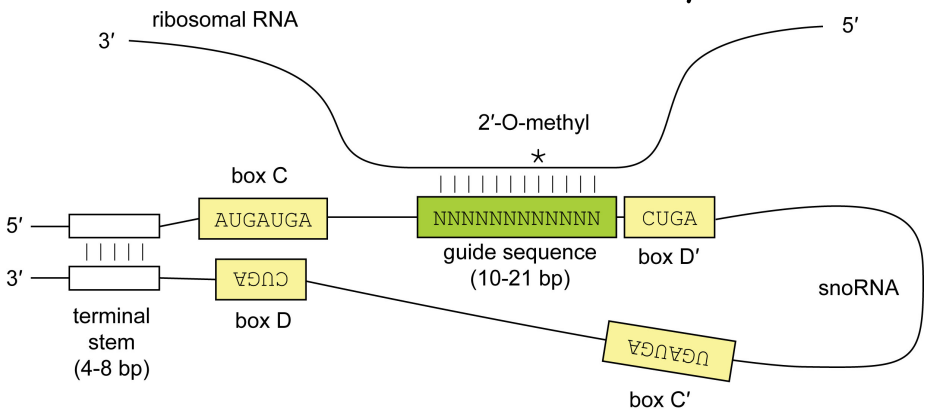


On regroupe les expériences. Une jonction doit être soutenue par :

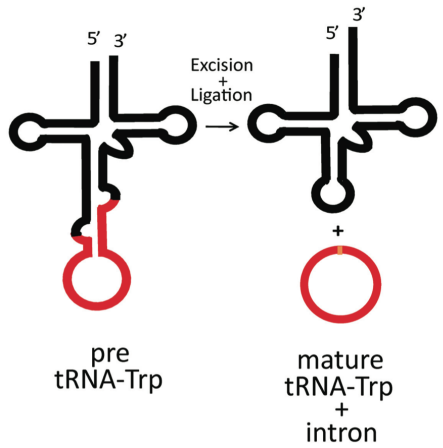
- au moins 3 lectures venant d'au moins 2 expériences,
- au moins la moitié des lectures circulaires dans la jonction.

Loci contenant une jonction identifiée

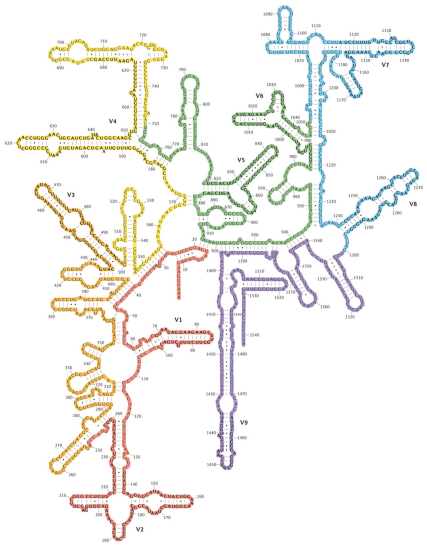
ARNsno à boites C/D



Intron du ARNt-Trp



ARNr

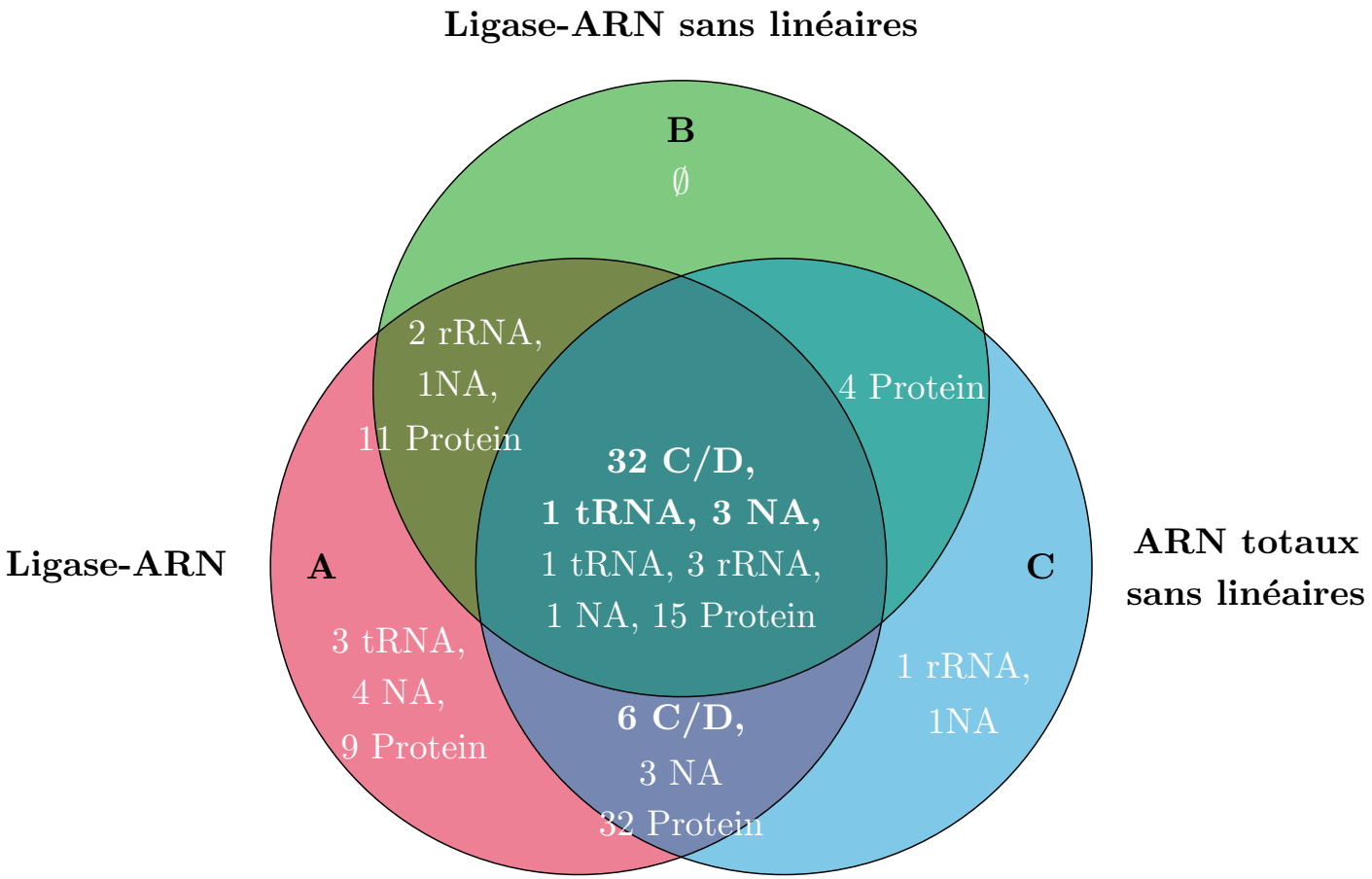


ARN codant pour une protéine

ARN non annoté

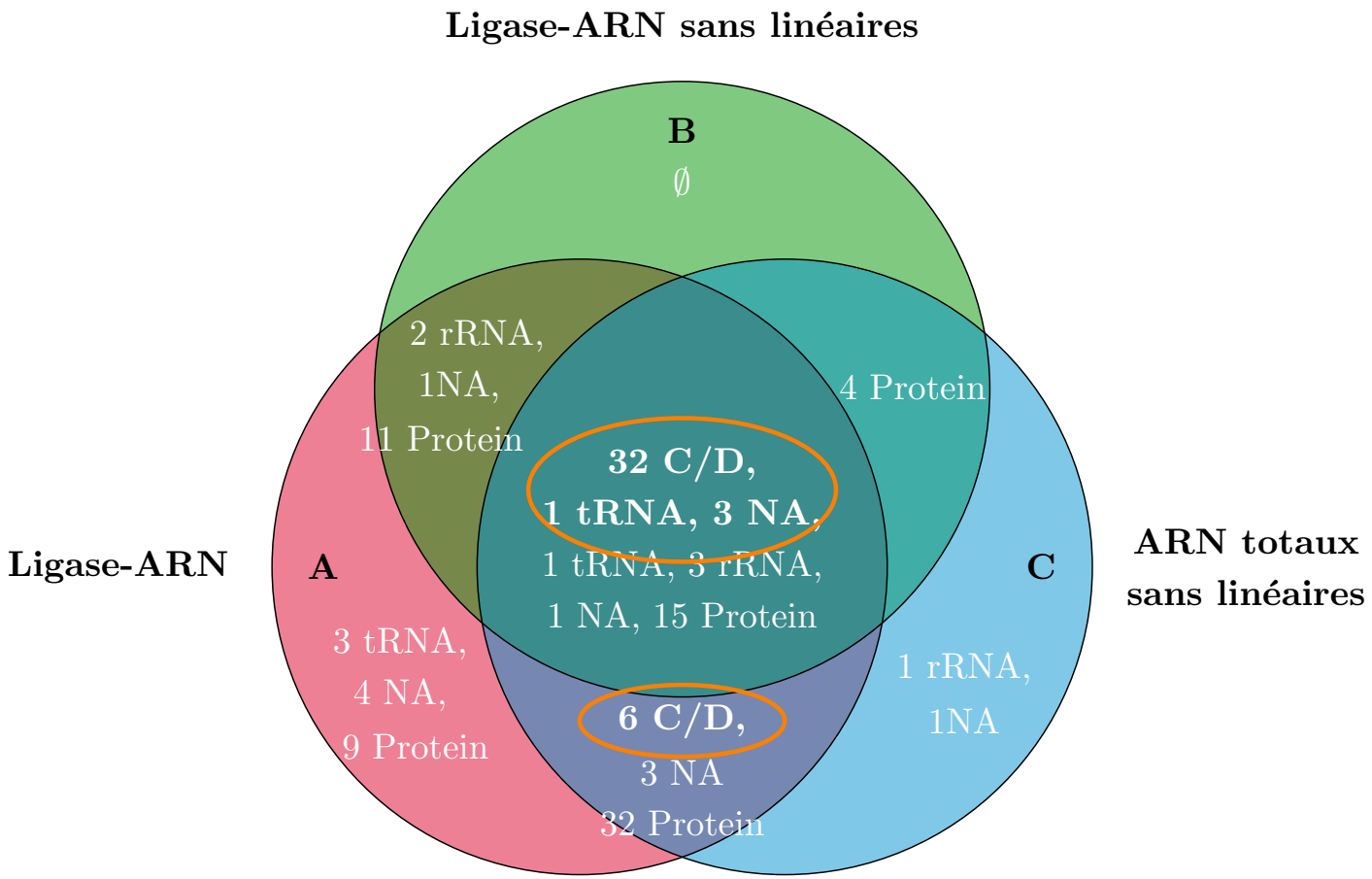
H.F. Becker, A. Héliou, K. Djaout, R. Lestini, M. Regnier, H. Myllykallio, RNA Biology 2017

Diagramme de Venn des jonctions identifiées dans nos expériences



H.F. Becker, A. Héliou, K. Djaout, R. Lestini, M. Regnier, H. Myllykallio, RNA Biology 2017

Diagramme de Venn des jonctions identifiées dans nos expériences



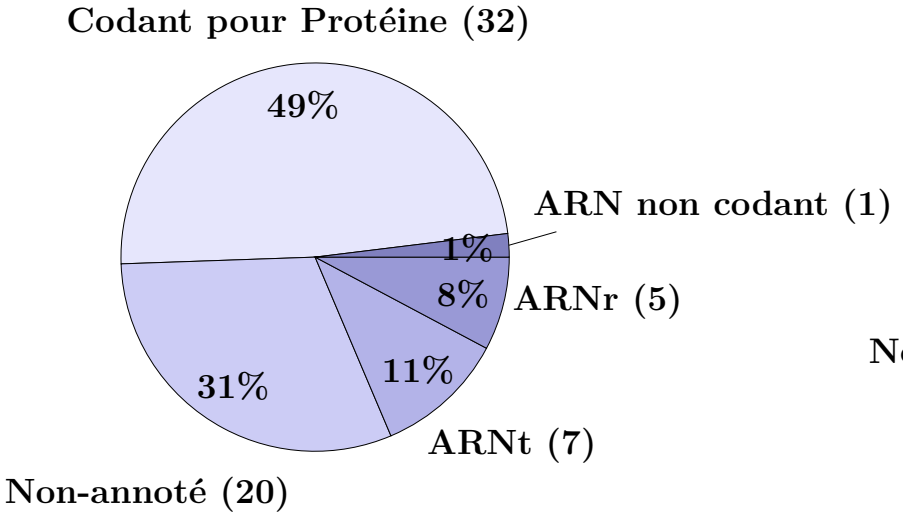
Les jonctions entourées sont enrichies en circulaires avec le traitement RNase R (dégrade les linéaires).

Analyse des ARNs chez d'autres organismes

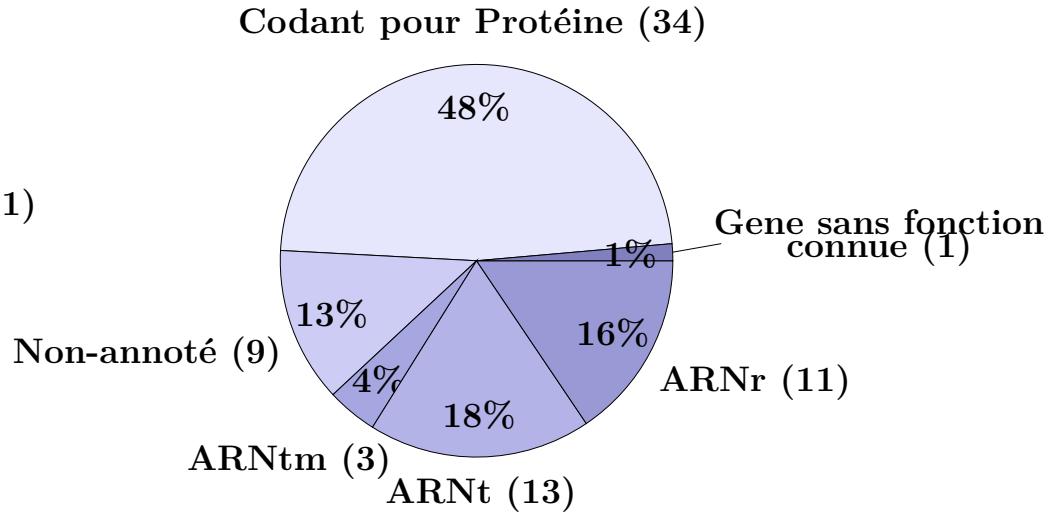
Organisme	Caractéristique	Protéine homolgue à Pab1020
Natrialba magadii	Archée halophile	oui
Haloferax volcanii	Archée halophile	non
Aquifex aeolicus	Bactérie hyperthermophile	oui
Halorhodospira halophila	Bactérie halophile	oui

Nos premiers résultats, quantification par loci

Haloferax volcanii (65)

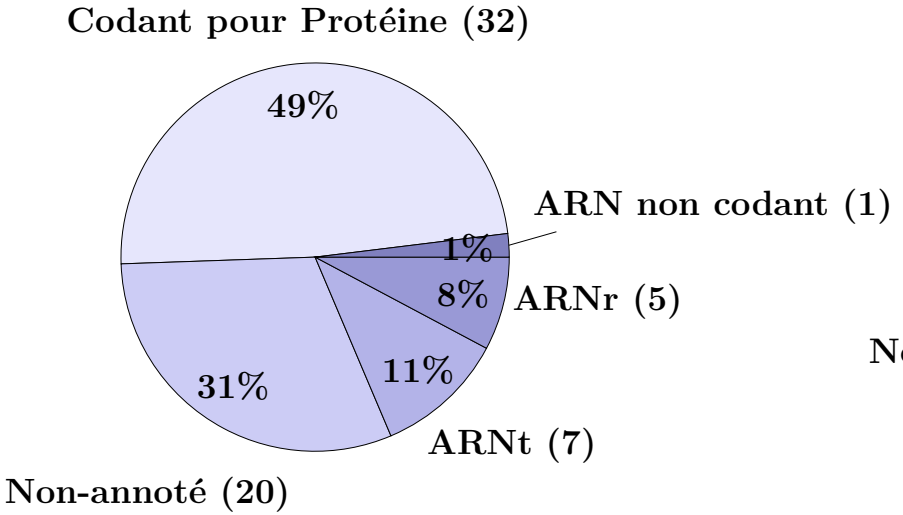


Aquifex aeolicus (71)

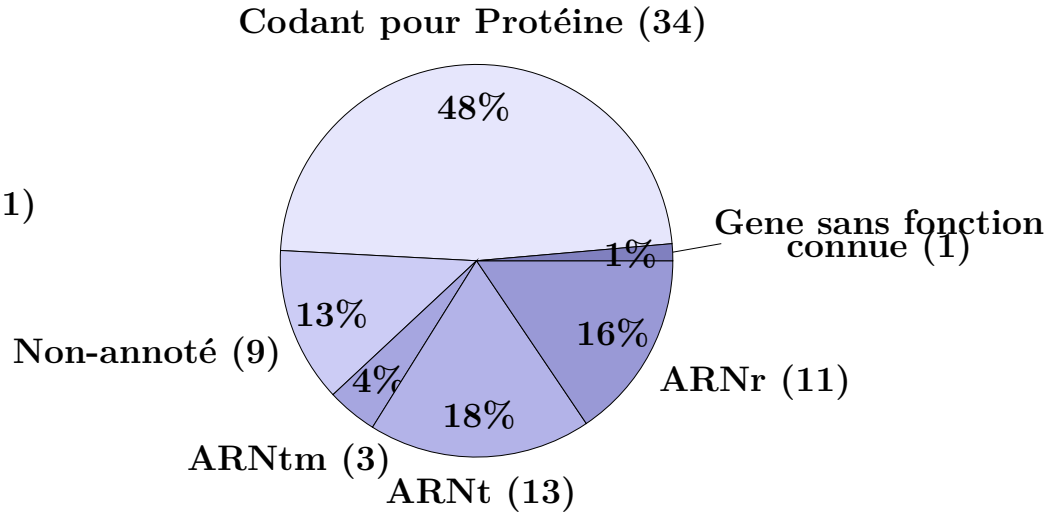


Nos premiers résultats, quantification par loci

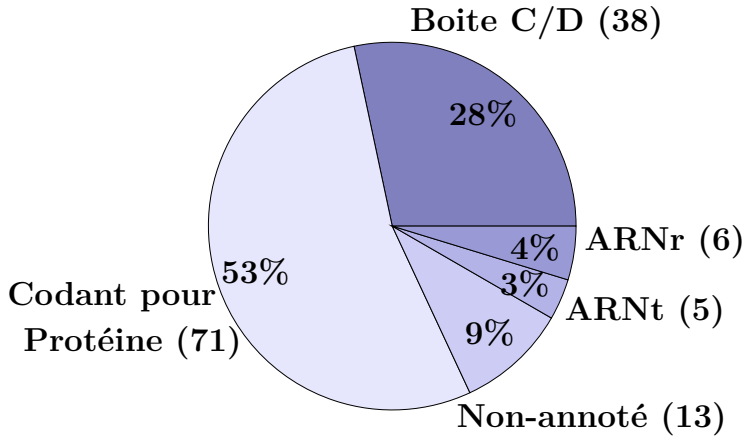
Haloferax volcanii (65)



Aquifex aeolicus (71)



Pyrococcus abyssi (133)



Analyse de la fonction de la ligase *in vivo*

Chez *Thermococcus barophilus*, Archée hyperthermophile.

- Inactivation du gène codant pour la ligase, collaboration avec l'Ifremer (Brest).
- RNA-seq, et analyse des données de séquençage.

Analyse de la fonction de la ligase *in vivo*

Chez *Thermococcus barophilus*, Archée hyperthermophile.

- Inactivation du gène codant pour la ligase, collaboration avec l'Ifremer (Brest).
- RNA-seq, et analyse des données de séquençage.

→ Comparaison des populations d'ARN circulaires avec et sans la ligase.
→ L'ARN ligase de la famille Rnl3 est-elle indispensable à la circularisation ?

Identification des ARNs circulaires

Avec H.F. Becker, K. Djaout, R. Lestini, M. Regnier, H. Myllykallio,

- "Hight-Throughput Sequencing Reveals Circular Substrates for an Archaeal RNA ligase", RNA Biology, 2017

Identification des ARNs circulaires

Avec H.F. Becker, K. Djaout, R. Lestini, M. Regnier, H. Myllykallio,

- "Hight-Throughput Sequencing Reveals Circular Substrates for an Archaeal RNA ligase", RNA Biology, 2017

Algorithmes de calcul des mots absents minimaux

Avec C. Barton, L. Mouchard, S. P. Pissis et S. Puglisi

- Calcul avec la table des suffixes, BMC Bioinformatics, 2014
- Calcul parallèle, PPAM, 2015
- Calcul en mémoire externe, Bioinformatics, 2017

Avec M. Crochemore, G. Kucherov, L. Mouchard, S. P. Pissis et Y. Ramusat

- Calcul sur une fenêtre glissante et recherche de motifs, FCT 2017

Perspectives

Algorithmes de calcul des mots absents minimaux

- Implémenter l'algorithme sur une fenêtre glissante.
- Comparer l'alignement de lectures avec cette méthode et les algorithmes classiques.

Perspectives

Algorithmes de calcul des mots absents minimaux

- Implémenter l'algorithme sur une fenêtre glissante.
- Comparer l'alignement de lectures avec cette méthode et les algorithmes classiques.

Identification des ARNs circulaires

- Séquencer d'autres organismes procaryotes et comparer leurs ARNs circulaires.

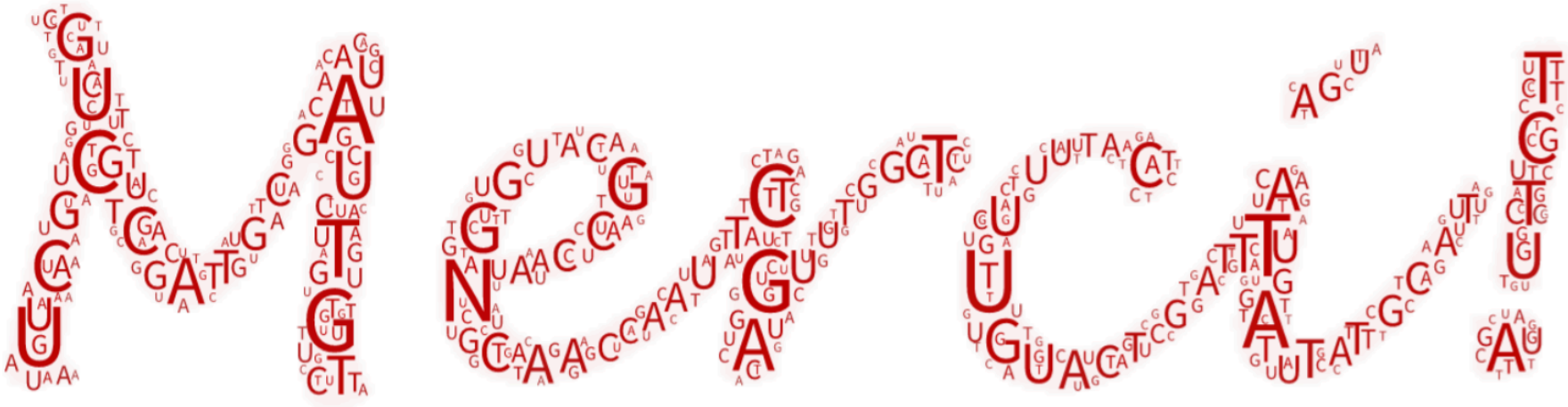
Perspectives

Algorithmes de calcul des mots absents minimaux

- Implémenter l'algorithme sur une fenêtre glissante.
- Comparer l'alignement de lectures avec cette méthode et les algorithmes classiques.

Identification des ARNs circulaires

- Séquencer d'autres organismes procaryotes et comparer leurs ARNs circulaires.
- Étendre la méthode aux organismes avec épissage.
  $\rightarrow$ Utiliser un algorithme d'alignement plus approprié : STAR (Dobin et al., 2013), BWA-MEM (Li et al, 2013), ...



LOB

- Hubert Becker
- Hannu Myllykallio
- Roxane Lestini
- Yoann Collien
- et tous les autres

LIX

- Mireille Régnier
- Yann Ponty
- Philippe Chassignet
- Amélie Héliou
- Afaf Saaidi
- Juraj Michalik
- et tous les autres

Université de Rouen

- Laurent Mouchard

King's College London

- Solon Pissis
- Carl Barton

Université d'Helsinki

- Simon Puglisi

Université Paris Est

- Maxime Crochemore
- Gregory Kucherov

ENS Paris

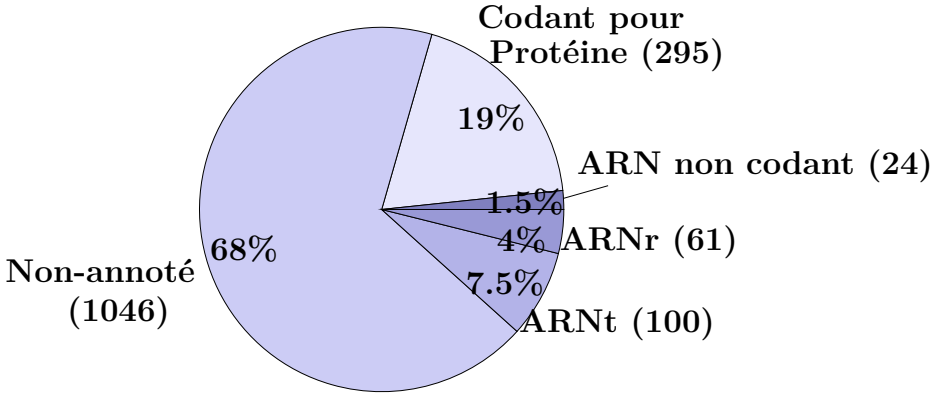
- Yann Ramusat

Institut Sanger

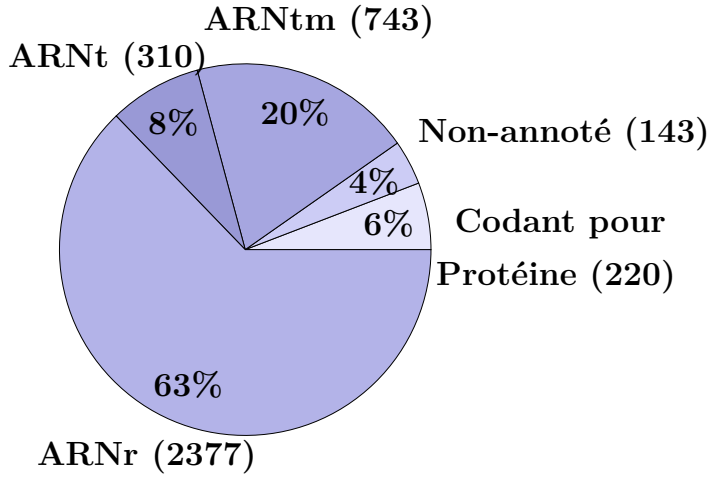
- Thomas Keane
- Dirk Dolle

Nos premiers résultats, quantification par lectures alignées

Haloferax volcanii (1540)

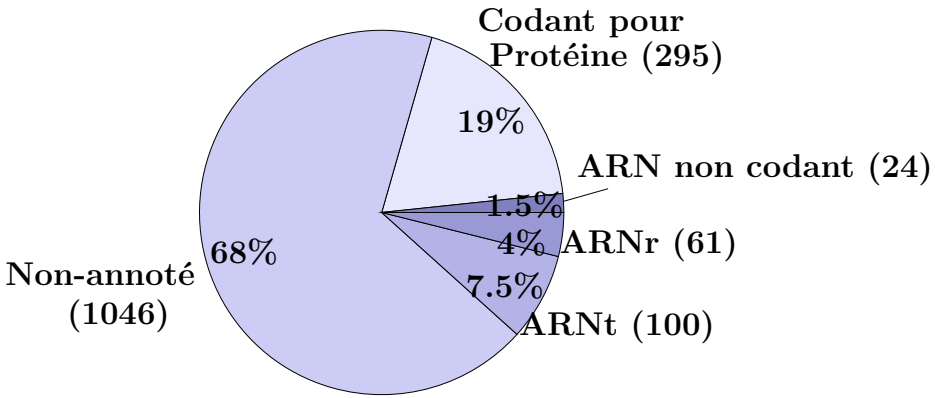


Aquifex aeolicus (3795)

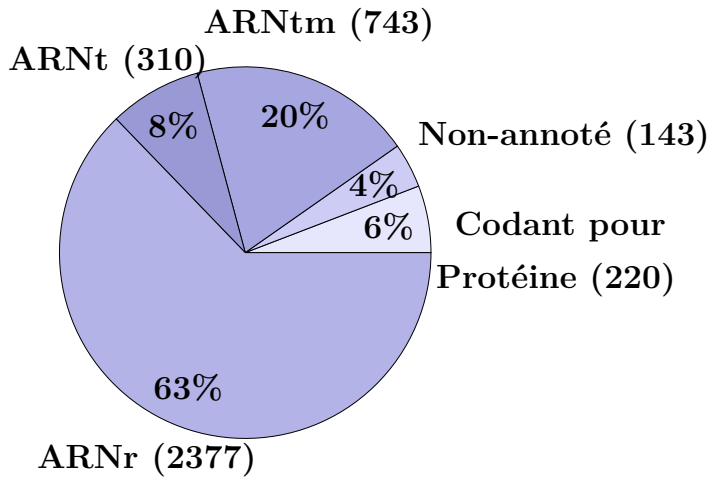


Nos premiers résultats, quantification par lectures alignées

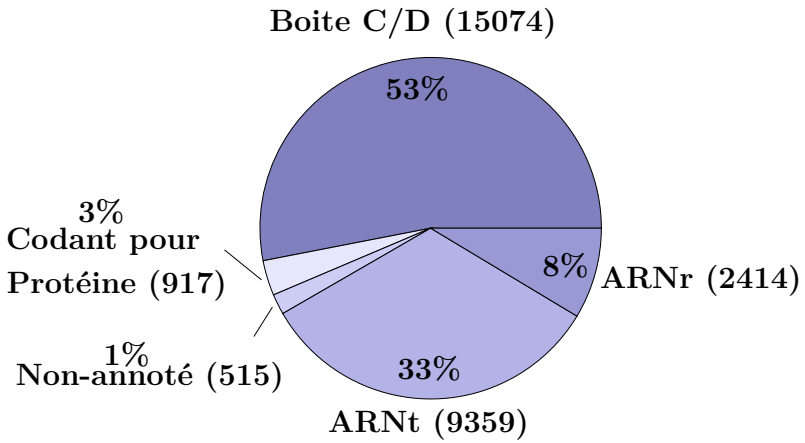
Haloferax volcanii (1540)



Aquifex aeolicus (3795)



Pyrococcus abyssi (28279)



Calcul des mots absents minimaux sur une fenêtre glissante

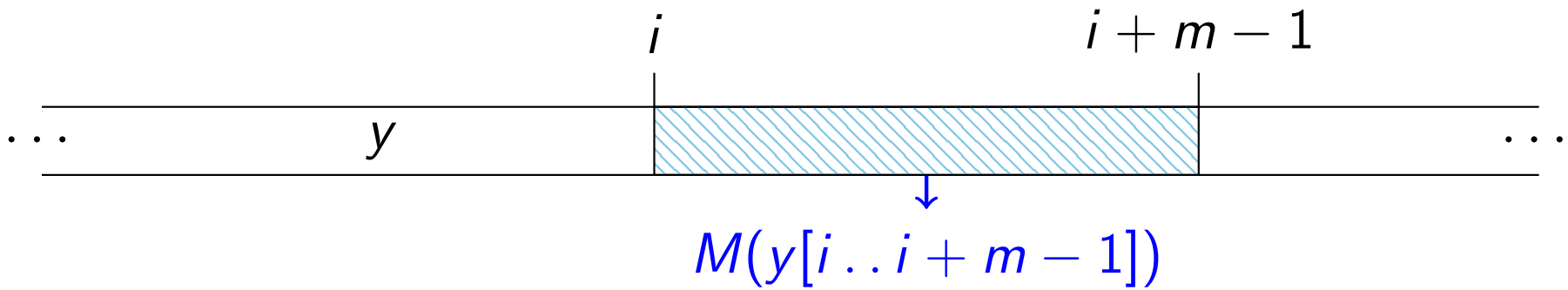
- Un mot y de taille n, sur un alphabet de taille constante.
- Une fenêtre glissante de taille m sur y : $y[i \,..\, i + m - 1]$

Pour tout mot x on note $M(x)$ l'ensemble de ses mots absents minimaux.

Calcul des mots absents minimaux sur une fenêtre glissante

- Un mot y de taille n, sur un alphabet de taille constante.
- Une fenêtre glissante de taille m sur y : $y[i \,..\, i+m-1]$

Pour tout mot x on note $M(x)$ l'ensemble de ses mots absents minimaux.



Calcul des mots absents minimaux sur une fenêtre glissante

Lemme

La borne haute de $\sum_{i=0}^{n-m} |M(y[i\,..\,i+m-1])|$ est $\mathcal{O}(nm)$.

Calcul des mots absents minimaux sur une fenêtre glissante

Lemme

La borne haute de $\sum_{i=0}^{n-m} |M(y[i\,..\,i+m-1])|$ est $\mathcal{O}(nm)$.

$\rightarrow$ On ne peut pas écrire l'ensemble des mots absents minimaux de chaque facteur de taille m en temps $\mathcal{O}(n)$.

Calcul des mots absents minimaux sur une fenêtre glissante

Lemme

La borne haute de $\sum_{i=0}^{n-m} |M(y[i\,..\,i+m-1])|$ est $\mathcal{O}(nm)$.

$\rightarrow$ On ne peut pas écrire l'ensemble des mots absents minimaux de chaque facteur de taille m en temps $\mathcal{O}(n)$.

Théoreme

La borne haute de $\sum_{i=0}^{n-m-1} |M(y[i\,..\,i+m-1]) \triangle M(y[i+1\,..\,i+m])|$ est $\mathcal{O}(n)$.

Calcul des mots absents minimaux sur une fenêtre glissante

Lemme

La borne haute de $\sum_{i=0}^{n-m} |M(y[i\,..\,i+m-1])|$ est $\mathcal{O}(nm)$.

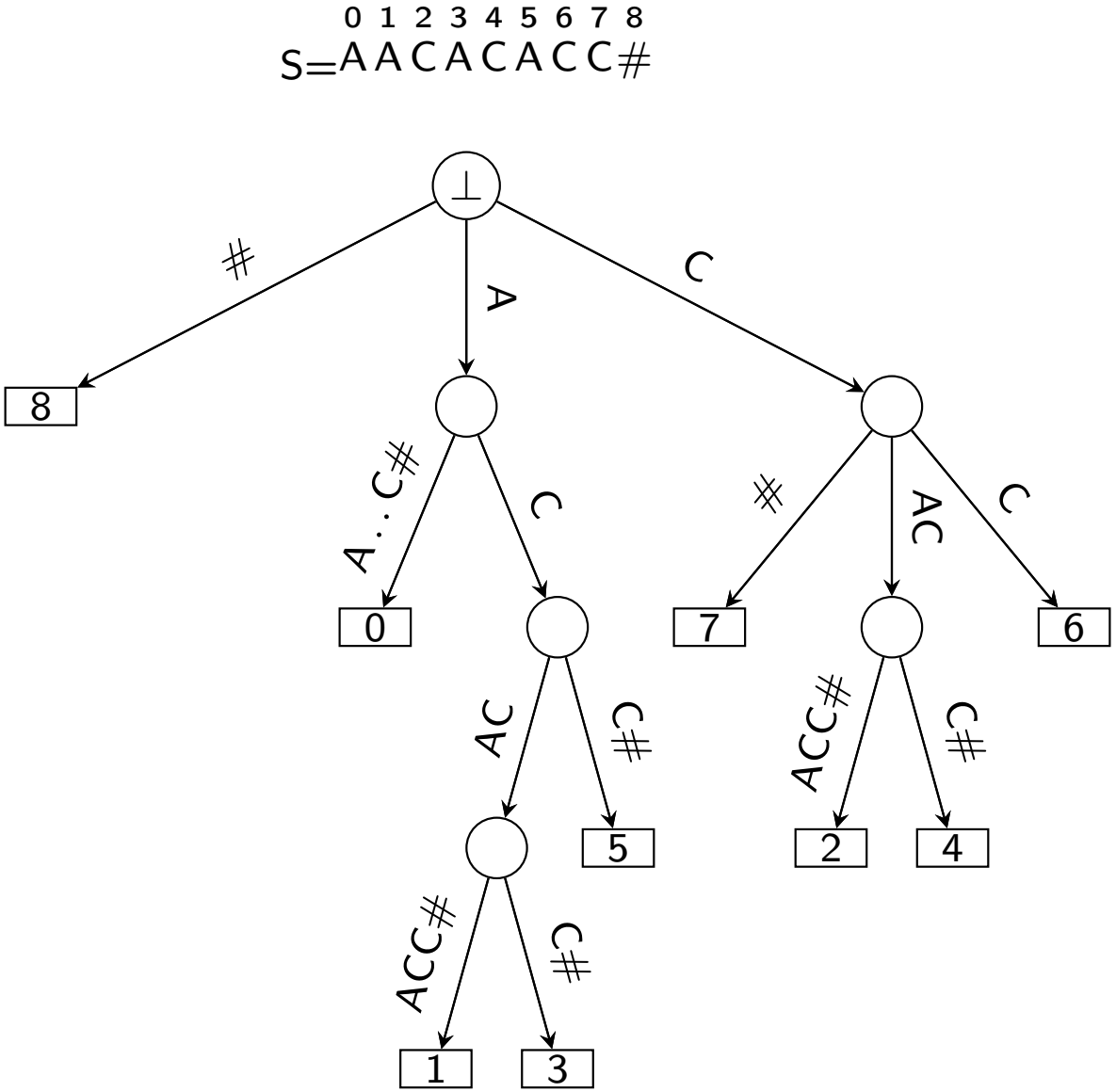
$\rightarrow$ On ne peut pas écrire l'ensemble des mots absents minimaux de chaque facteur de taille m en temps $\mathcal{O}(n)$.

Théoreme

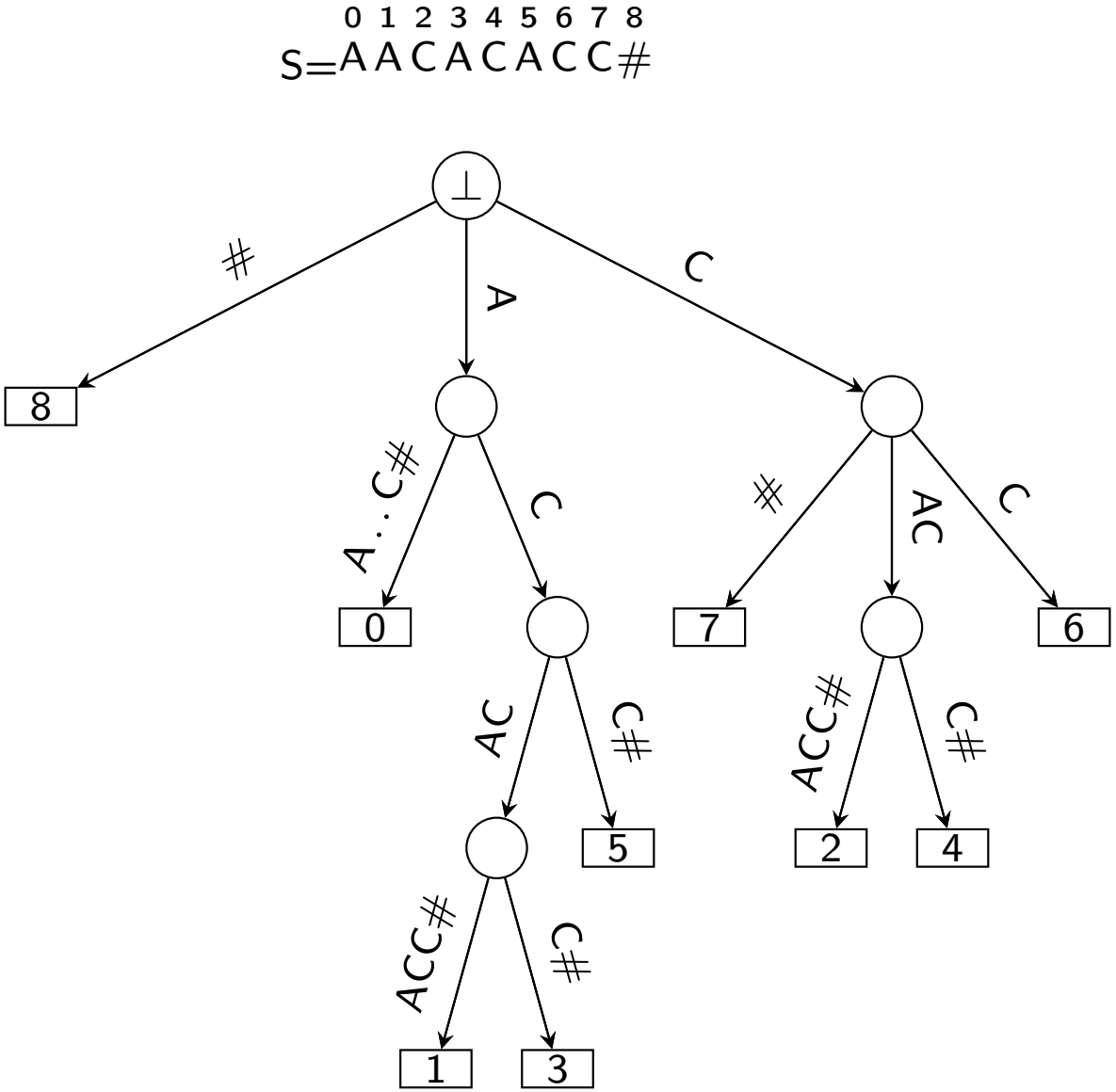
La borne haute de $\sum_{i=0}^{n-m-1} |M(y[i\,..\,i+m-1]) \triangle M(y[i+1\,..\,i+m])|$ est $\mathcal{O}(n)$.

$\rightarrow$ Il nous faut une structure dynamique pour passer d'un ensemble à un autre.

L'arbre des suffixes : une structure plus couteuse en espace mais dynamique



L'arbre des suffixes : une structure plus couteuse en espace mais dynamique



LCP	SA
0	8
0	0
1	1
4	3
2	5
0	7
1	2
3	4
1	6

L'arbre des suffixes pour une fenêtre glissante

Construction dynamique de l'arbre des suffixes

- Weiner en 1973 propose une construction de gauche à droite
- McCreight en 1976 propose une construction de droite à gauche
- Ukkonen en 1995 simplifie l'algorithme de Weiner

L'arbre des suffixes pour une fenêtre glissante

Construction dynamique de l'arbre des suffixes

- Weiner en 1973 propose une construction de gauche à droite
- McCreight en 1976 propose une construction de droite à gauche
- Ukkonen en 1995 simplifie l'algorithme de Weiner

L'arbre des suffixes pour une fenêtre glissante, Senft 2005

- Enlever la lettre la plus à gauche,
- Mettre à jour les étiquettes des branches

Les mots absents minimaux pour une fenêtre glissante

Nous avons adapté l'algorithme de Senft (2005), aux mots absent minimaux.

Les mots absents minimaux pour une fenêtre glissante

Nous avons adapté l'algorithme de Senft (2005), aux mots absent minimaux.

- Ajout sur l'arbre de l'information contenue dans la BWT.

Les mots absents minimaux pour une fenêtre glissante

Nous avons adapté l'algorithme de Senft (2005), aux mots absent minimaux.

- Ajout sur l'arbre de l'information contenue dans la BWT.
- Ajout sur l'arbre des mots absent minimaux

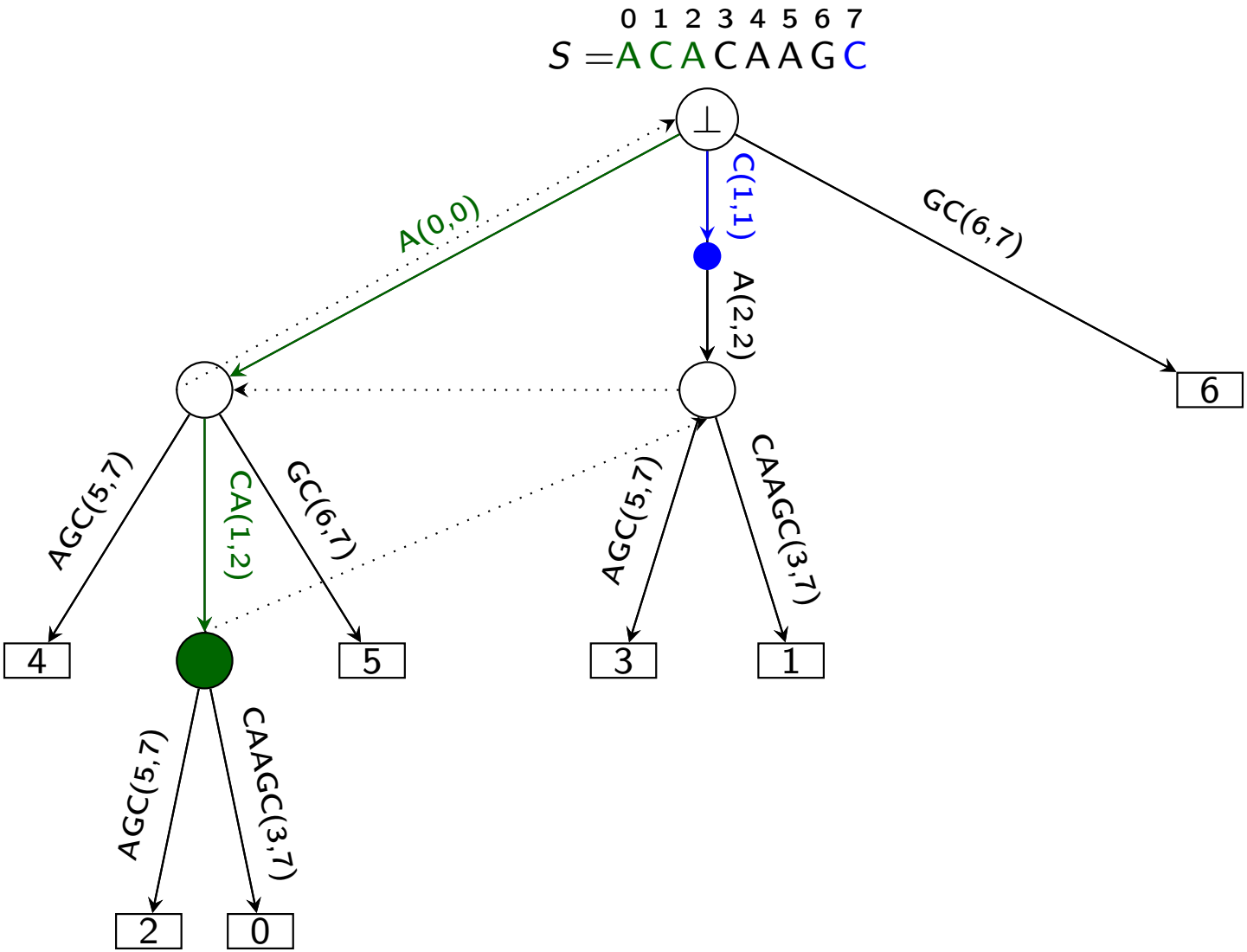
Les mots absents minimaux pour une fenêtre glissante

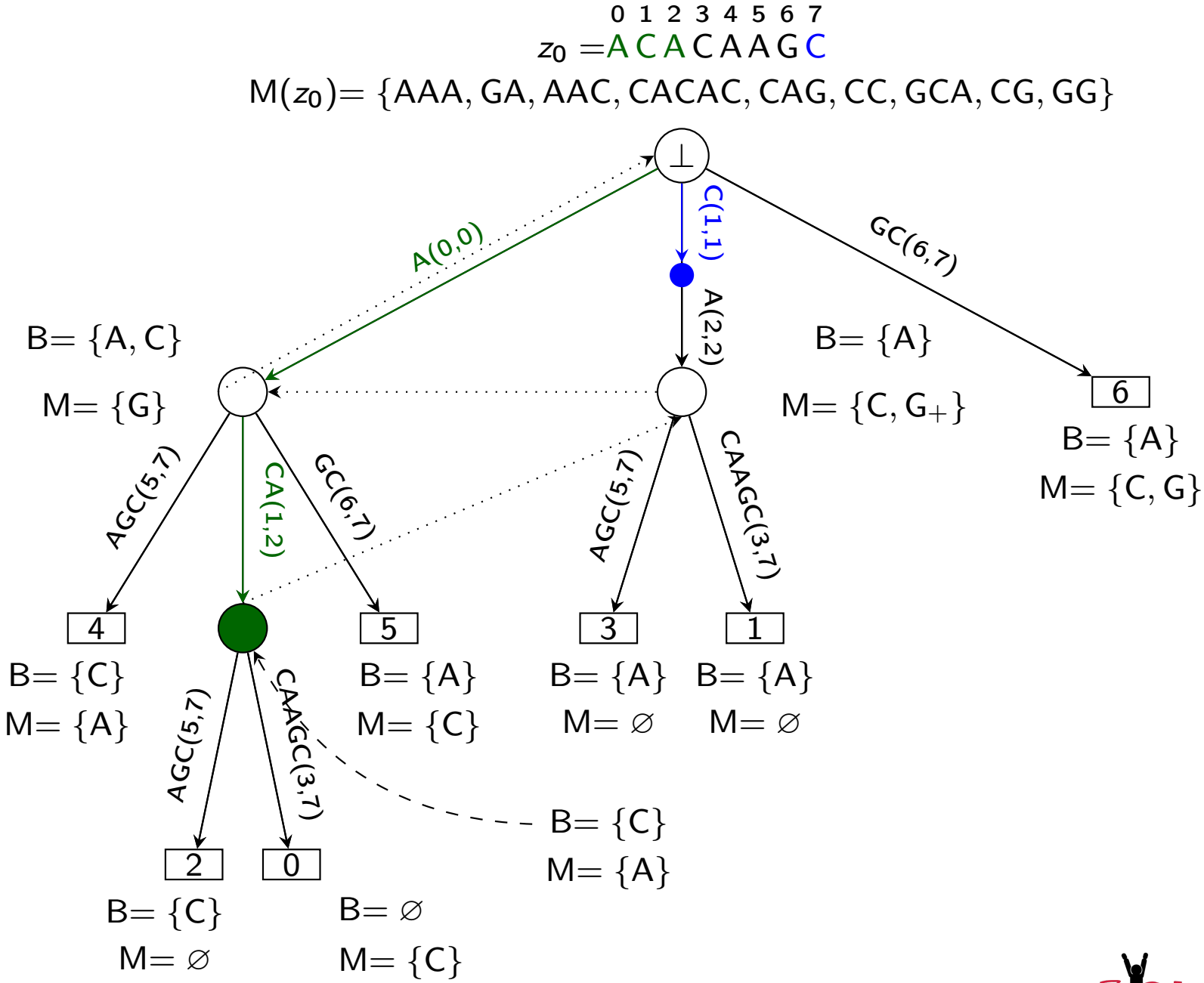
Nous avons adapté l'algorithme de Senft (2005), aux mots absent minimaux.

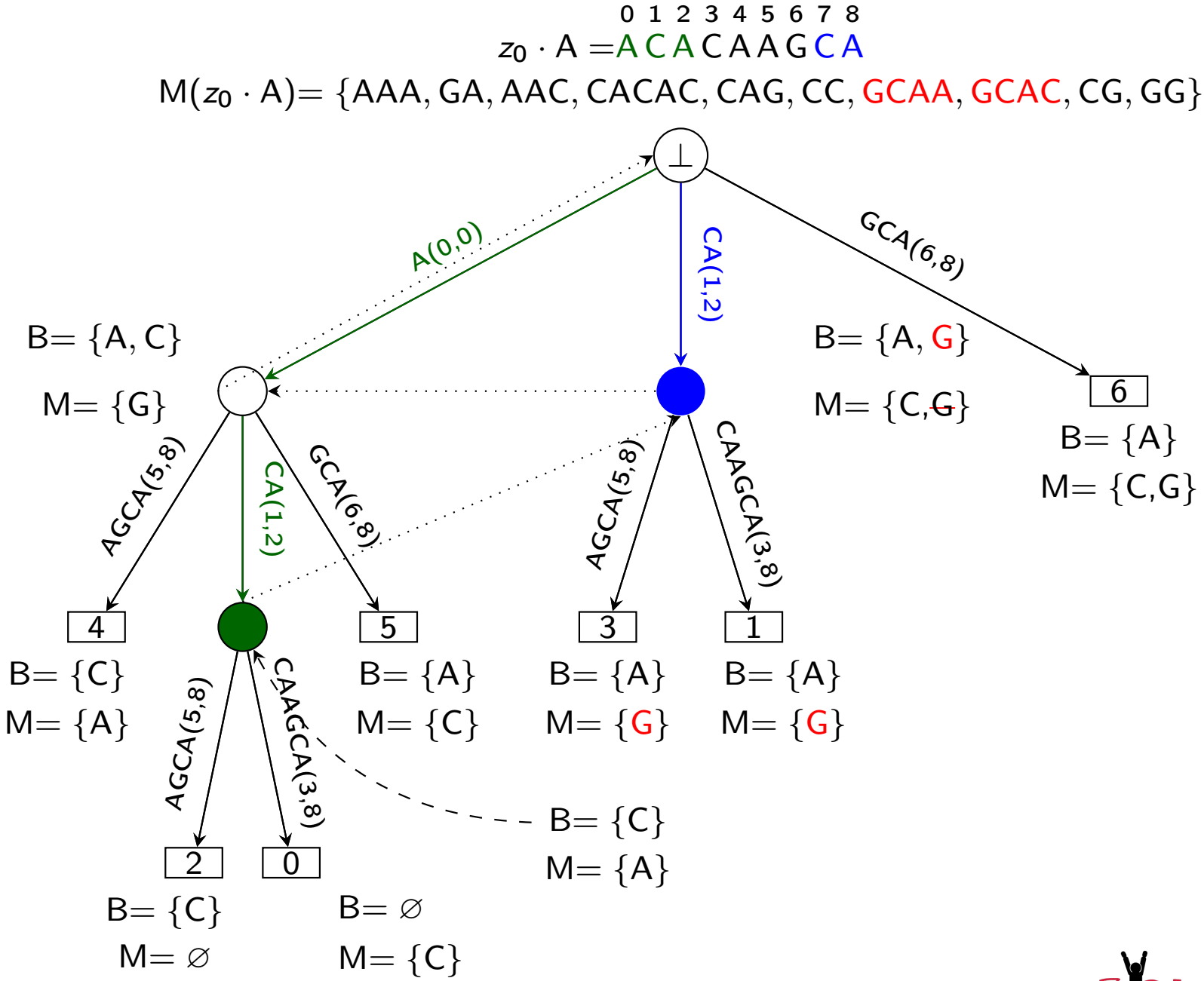
- Ajout sur l'arbre de l'information contenue dans la BWT.
- Ajout sur l'arbre des mots absent minimaux

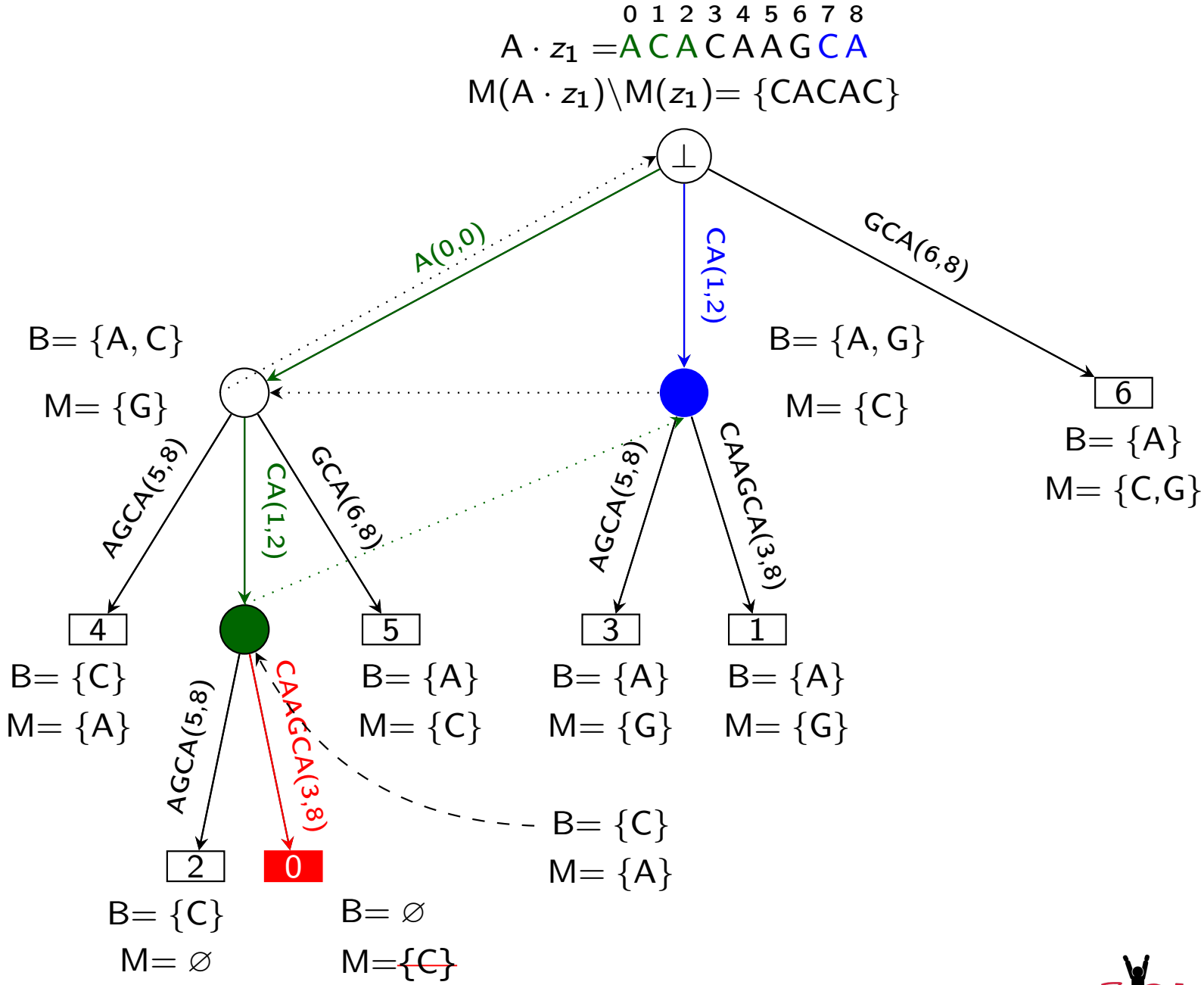
La fonction f est une injection

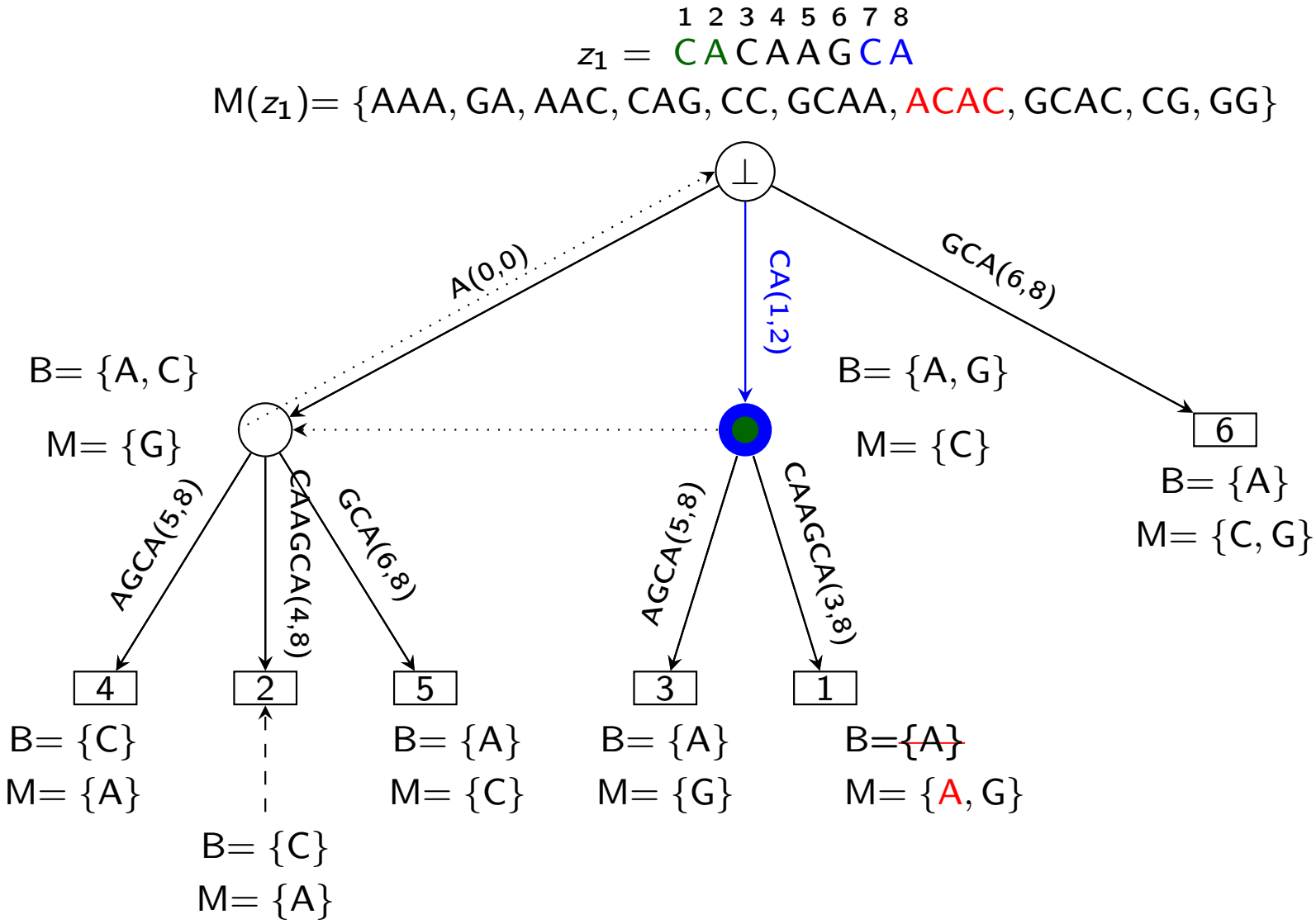
$f : M(z) \to \Sigma(z) \times V(z)$ définit par $f(aub) = (a, v_{ub})$,
où $a \in \Sigma$ et v_{ub} est le noeud correspondant à ub.


0 1 2 3 4 5 6 7
S = A C A C A A G C
⊥
A(0,0)
C(1,1)
A(2,2)
GC(6,7)
AGC(5,7)
CA(1,2)
GC(6,7)
AGC(5,7)
CAAGC(3,7)
AGC(5,7)
CAAGC(3,7)
6
4
5
3
1
2
0


0 1 2 3 4 5 6 7
z_0 = A C A C A A G C
M(z_0)= {AAA, GA, AAC, CACAC, CAG, CC, GCA, CG, GG}
⊥
A(0,0)
C(1,1)
A(2,2)
GC(6,7)
B= {A, C}
M= {G}
B= {A}
M= {C, G+}
6
B= {A}
M= {C, G}
AGC(5,7)
CA(1,2)
GC(6,7)
AGC(5,7)
CAAGC(3,7)
4
B= {C}
M= {A}
5
B= {A}
M= {C}
3
B= {A}
M= ∅
1
B= {A}
M= ∅
AGC(5,7)
CAAGC(3,7)
2
B= {C}
M= ∅
0
B= ∅
M= {C}
B= {C}
M= {A}


0 1 2 3 4 5 6 7 8
$z_0 \cdot A$ = A C A C A A G C A
M($z_0 \cdot A$)= {AAA, GA, AAC, CACAC, CAG, CC, GCAA, GCAC, CG, GG}
⊥
A(0,0)
CA(1,2)
GCA(6,8)
B= {A, C}
M= {G}
B= {A, G}
M= {C,G}
6
B= {A}
M= {C,G}
AGCA(5,8)
CA(1,2)
GCA(6,8)
AGCA(5,8)
CAAGCA(3,8)
4
B= {C}
M= {A}
5
B= {A}
M= {C}
3
B= {A}
M= {G}
1
B= {A}
M= {G}
AGCA(5,8)
CAAGCA(3,8)
B= {C}
M= {A}
2
B= {C}
M= ∅
0
B= ∅
M= {C}


0 1 2 3 4 5 6 7 8
A · z1 =A C A C A A G C A
M(A · z1)\M(z1)= {CACAC}
⊥
A(0,0)
CA(1,2)
GCA(6,8)
B= {A, C}
M= {G}
B= {A, G}
M= {C}
6
B= {A}
M= {C,G}
AGCA(5,8)
CA(1,2)
GCA(6,8)
AGCA(5,8)
CAAGCA(3,8)
4
5
3
1
B= {C}
M= {A}
B= {A}
M= {C}
B= {A}
M= {G}
B= {A}
M= {G}
AGCA(5,8)
CAAGCA(3,8)
2
0
B= {C}
M= {A}
B= {C}
M= ∅
B= ∅
M={C}


1 2 3 4 5 6 7 8
z_1 = C A C A A G C A
M(z_1)= {AAA, GA, AAC, CAG, CC, GCAA, ACAC, GCAC, CG, GG}
⊥
A(0,0)
CA(1,2)
GCA(6,8)
B= {A, C}
M= {G}
B= {A, G}
M= {C}
6
B= {A}
M= {C, G}
AGCA(5,8)
CAAGCA(4,8)
GCA(6,8)
AGCA(5,8)
CAAGCA(3,8)
4
2
5
3
1
B= {C}
M= {A}
B= {C}
M= {A}
B= {A}
M= {C}
B= {A}
M= {G}
B= {A}
M= {A, G}