Les méthodes de séquençage

J-F. Gibrat

Unité Mathématique, Informatique et Génome,
INRA, Jouy-en-Josas

Module 8, 15 novembre 2010

Séquençage

- Plus petit génome (non viral) connu : *Carsonella ruddii* 0.16 Mbp
- Plus grand génome connu : *Amoeba dubia* 160 Gb
- Taille maximale des lectures (Sanger) : 900 bp
- Nécessité de découper le génome en millions de fragments (shotgun sequencing) ▶

Deux cas possibles :

1. On dispose du génome $\Longrightarrow$ alignement (mapping) des lectures
   - RNA-seq, CHIP-seq, SNPs, métagénomique, etc.
2. On ne dispose pas du génome $\Longrightarrow$ assemblage des lectures
   - séquençage *de novo*, re-séquençage, métagénomique, réarrangements chromosomiques, etc.

Séquençage

- Plus petit génome (non viral) connu : *Carsonella ruddii* 0.16 Mbp
- Plus grand génome connu : *Amoeba dubia* 160 Gb
- Taille maximale des lectures (Sanger) : 900 bp
- Nécessité de découper le génome en millions de fragments (shotgun sequencing) ▶

Deux cas possibles :

1. On dispose du génome $\Longrightarrow$ alignement (mapping) des lectures
   - RNA-seq, CHIP-seq, SNPs, métagénomique, etc.
2. On ne dispose pas du génome $\Longrightarrow$ assemblage des lectures
   - séquençage *de novo*, re-séquençage, métagénomique, réarrangements chromosomiques, etc.

Séquençage

- Plus petit génome (non viral) connu : *Carsonella ruddii* 0.16 Mbp
- Plus grand génome connu : *Amoeba dubia* 160 Gb
- Taille maximale des lectures (Sanger) : 900 bp
- Nécessité de découper le génome en millions de fragments (shotgun sequencing) ▶

Deux cas possibles :

1. On dispose du génome $\Longrightarrow$ alignement (mapping) des lectures
   - RNA-seq, CHIP-seq, SNPs, métagénomique, etc.
2. On ne dispose pas du génome $\Longrightarrow$ assemblage des lectures
   - séquençage *de novo*, re-séquençage, métagénomique, réarrangements chromosomiques, etc.

Couverture du génome

- On veut au moins 99% du génome couvert par les reads
- Couverture $C = nl/L$ où n nombre de lectures, l longueur des lectures, L longueur du segment génomique
- Cela requiert une couverture 8x (reads 600-700 bp)▶
- Un génome 1 Mbp avec couverture 8x nécessite 11 000 lectures (700 nt).
- Utilité des « paired ends »▶

Causes de difficultés d'assemblage/mapping

- Données contiennent des erreurs
  - humaines
  - dues aux machines
- Qualité $Q = -10\ log_{10} P_{err}$
- Certaines parties du génome ne peuvent pas être clonées
- Présence de répétitions ▶
  - Important de pouvoir détecter et corriger problèmes dûs aux répétitions

▷ « Finishing »
  - ces différents problèmes font que assembleur peut générer plusieurs centaines de contigs
  - il faut « boucher les trous »
  - « sequence gaps » dans les scaffolds vs « physical gaps » entre les scaffolds

Causes de difficultés d'assemblage/mapping

- Données contiennent des erreurs
  - humaines
  - dues aux machines
- Qualité $Q = -10\ log_{10} P_{err}$
- Certaines parties du génome ne peuvent pas être clonées
- Présence de répétitions ▶
  - Important de pouvoir détecter et corriger problèmes dûs aux répétitions

▷ « Finishing »
  - ces différents problèmes font que assembleur peut générer plusieurs centaines de contigs
  - il faut « boucher les trous »
  - « sequence gaps » dans les scaffolds vs « physical gaps » entre les scaffolds

Causes de difficultés d'assemblage/mapping

- Données contiennent des erreurs
  - humaines
  - dues aux machines
- Qualité $Q = -10\ log_{10} P_{err}$
- Certaines parties du génome ne peuvent pas être clonées
- Présence de répétitions▶
  - Important de pouvoir détecter et corriger problèmes dûs aux répétitions

▷ « Finishing »
  - ces différents problèmes font que assembleur peut générer plusieurs centaines de contigs
  - il faut « boucher les trous »
  - « sequence gaps » dans les scaffolds vs « physical gaps » entre les scaffolds

Causes de difficultés d'assemblage/mapping

- Données contiennent des erreurs
  - humaines
  - dues aux machines
- Qualité $Q = -10\ log_{10} P_{err}$
- Certaines parties du génome ne peuvent pas être clonées
- Présence de répétitions▶
  - Important de pouvoir détecter et corriger problèmes dûs aux répétitions

▷ « Finishing »
  - ces différents problèmes font que assembleur peut générer plusieurs centaines de contigs
  - il faut « boucher les trous »
  - « sequence gaps » dans les scaffolds vs « physical gaps » entre les scaffolds

Différentes méthodes de séquençage

- Méthode Sanger : séquençage par synthèse (clonage)
- Méthode 454 : séquençage par synthèse (amplification PCR)
- Méthode Solexa : séquençage par synthèse (amplification PCR)
- Méthode SOLID : séquençage par ligation (amplification PCR)
- 3-20 μg ADN
- *Méthode Pacific Bioscience : séquençage molécule unique*
- *<1 μg ADN*

Différentes méthodes de séquençage

- Méthode Sanger : séquençage par synthèse (clonage)
- Méthode 454 : séquençage par synthèse (amplification PCR)
- Méthode Solexa : séquençage par synthèse (amplification PCR)
- Méthode SOLID : séquençage par ligation (amplification PCR)
- 3-20 μg ADN
- *Méthode Pacific Bioscience : séquençage molécule unique*
- *<1 μg ADN*

Propriétés des méthodes de séquençage

Vendor:	Roche			Illumina			ABI		
Technology:	454			Solexa GA			SOLiD		
Platform:	GS20	FLX	Ti	I	II	IIx	1	2	3
Reads: (M)	0.5	0.5	1.25	28	100	250	40	115	320
Fragment									
Read length:	100	200	400	35	50	100	25	35	50
Run time: (d)	0.25	0.3	0.4	3	3	5	6	5	8
Yield: (Gb)	0.05	0.1	0.5	1	5	25	1	4	16
Paired-end									
Read length:		200	400	2×35	2×50	2×100	2×25	2×35	2×50
Insert: (kb)		3.5	3.5	0.2	0.2	0.2	3	3	2
Run time: (d)		0.3	0.4	6	10	10	12	10	16
Yield: (Gb)		0.1	0.5	2	9	50	2	8	32

Illumina Hiseq 2000

- Nombre de lectures : 1 milliard
- Longueur lecture : 100 nt
- Production : 100 Gb
- Temps : $\sim$4 jours

- Lecture pairée : 2 x 100 nt
- Production : 200 Gb
- Temps : $\sim$8 jours
- Prix : $\sim$9 000 €

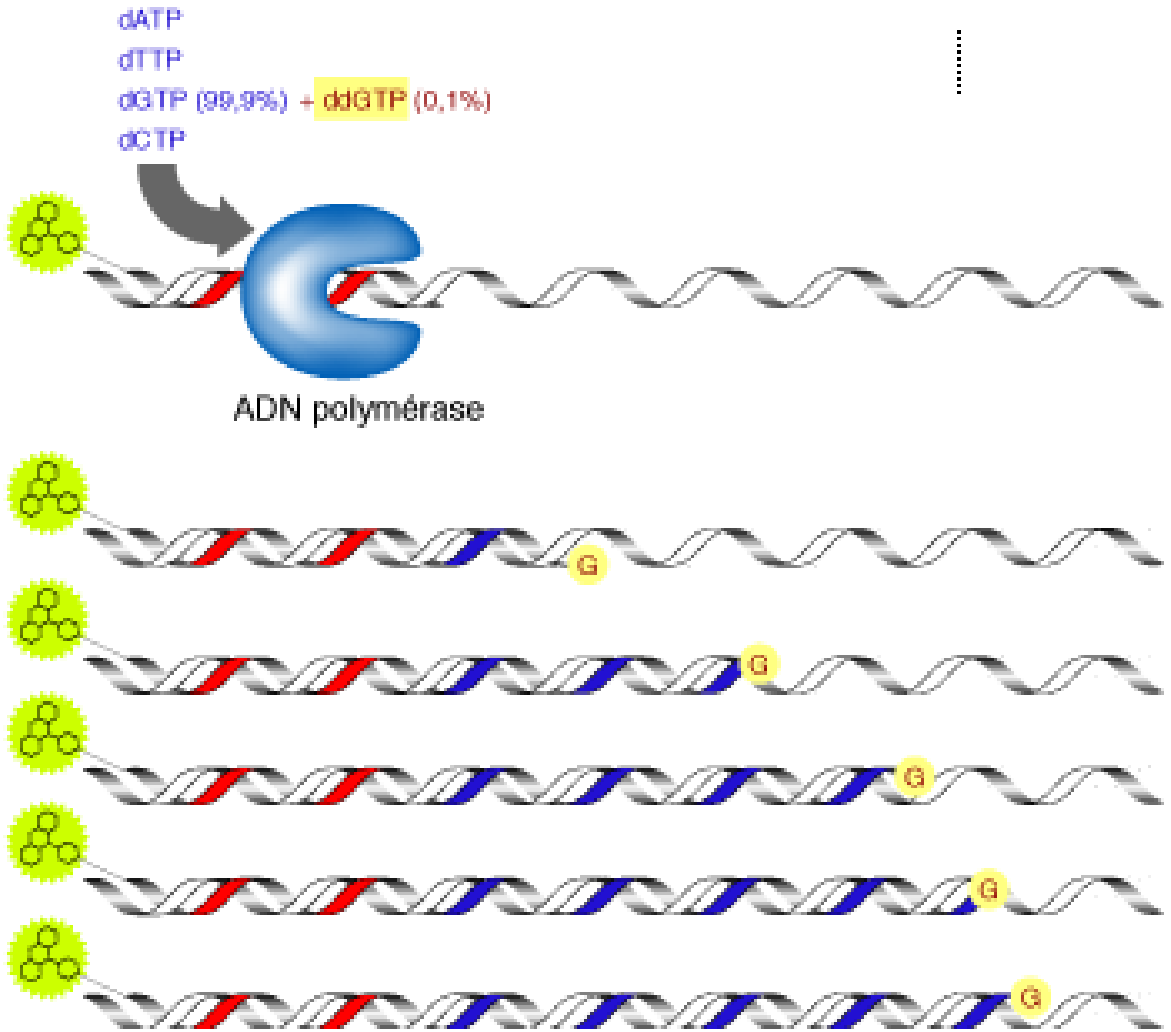
Principe des méthodes de séquençage

- Méthode Sanger : lecture $\sim$ 800 bp ; Prod. < 0.01Gb ▶
- Méthode 454 : lecture : $\sim$ 400 bp ; Prod. $\sim$0.5 Gb ▶
- Méthode Solexa : lecture $\sim$ 100 bp ; Prod. $\sim$200 Gb ▶
- Méthode SOLID : lecture $\sim$ 50 bp ; Prod. $\sim$32 Gb ▶

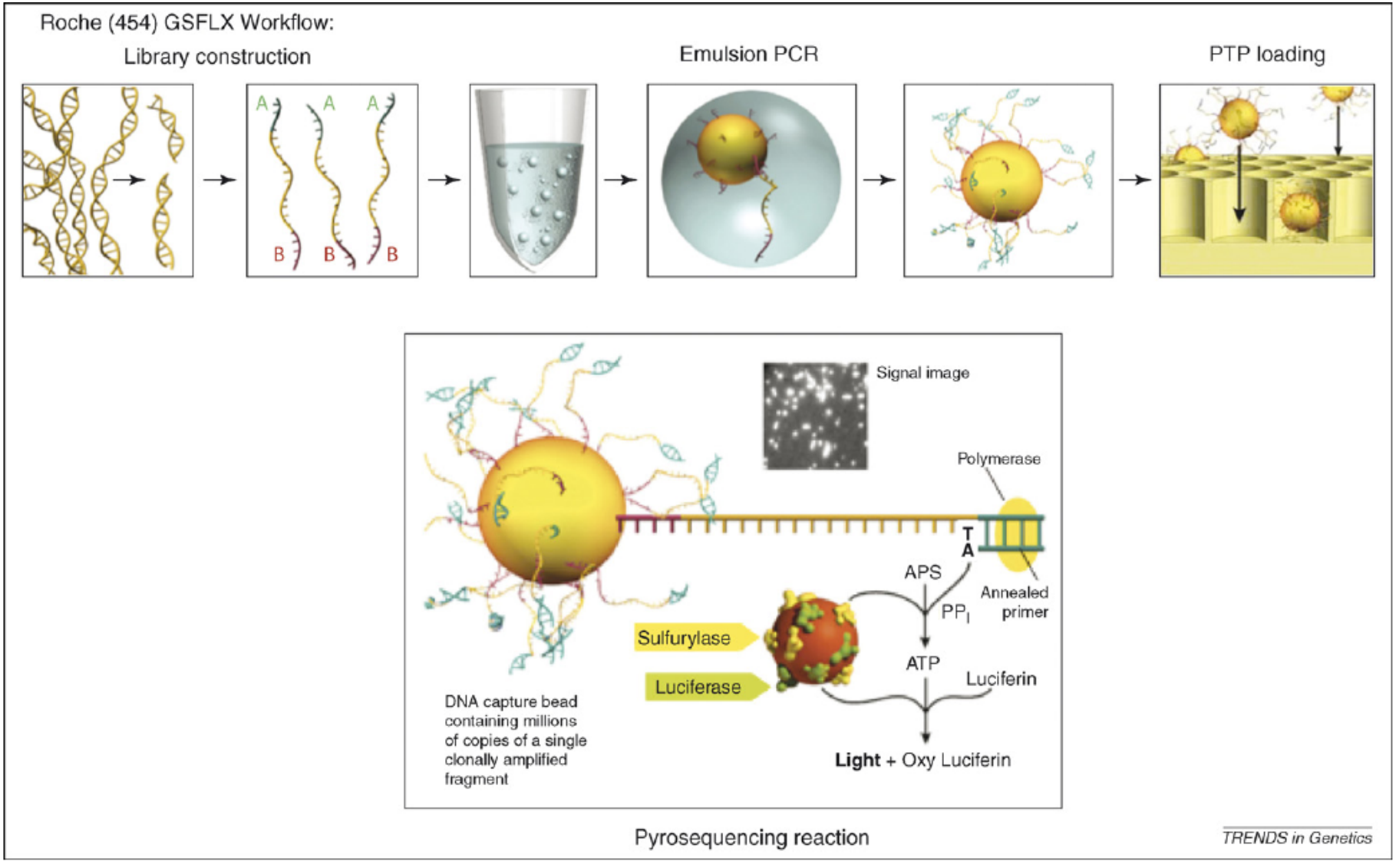
Algorithmes d'assemblage

1. Plus courte super-chaîne commune▶
2. Chemin Hamiltonien dans un graphe▶
3. Chemin Eulérien dans un graphe▶

Principes de la méthode Sanger



Principes de la méthode 454



Principes de la méthode 454

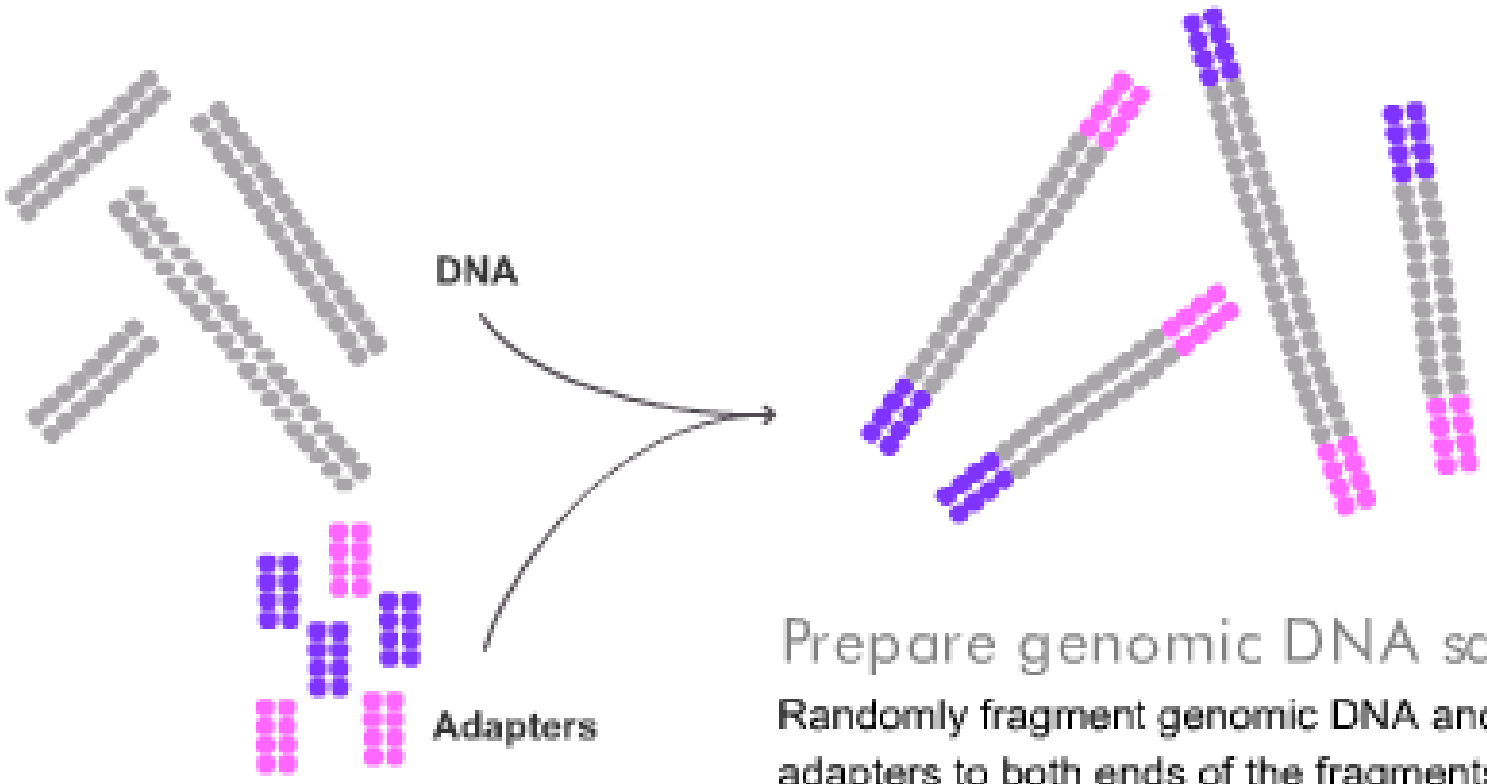
- création de fragments d'ADN par sonication
- séparation des brins + ajout d'un adaptateur B et d'une amorce de PCR A
- dans micelle chaque micro-bille fixe un fragment
- dans micelle amplification du fragment et fixation sur la bille (chaque bille contient plus de 1 million de copies du fragment original).
- la bille est placée dans un micro-puit
- chaque nucléotide est ajouté à son tour. S'il est incorporé il y a émission d'un photon. Signal proportionnel au nombre de photons.

Erreurs du 454

- estimation incorrecte de la longueur lors répétition nucléotide
- multiples fragments attachés sur la même bille
- base identique à un homopolymère proche est insérée à une localisation non adjacente.
- mesure de la qualité du 454 pas aussi fiable que celle de Sanger

Principe du Solexa : fixation d'adaptateurs

1

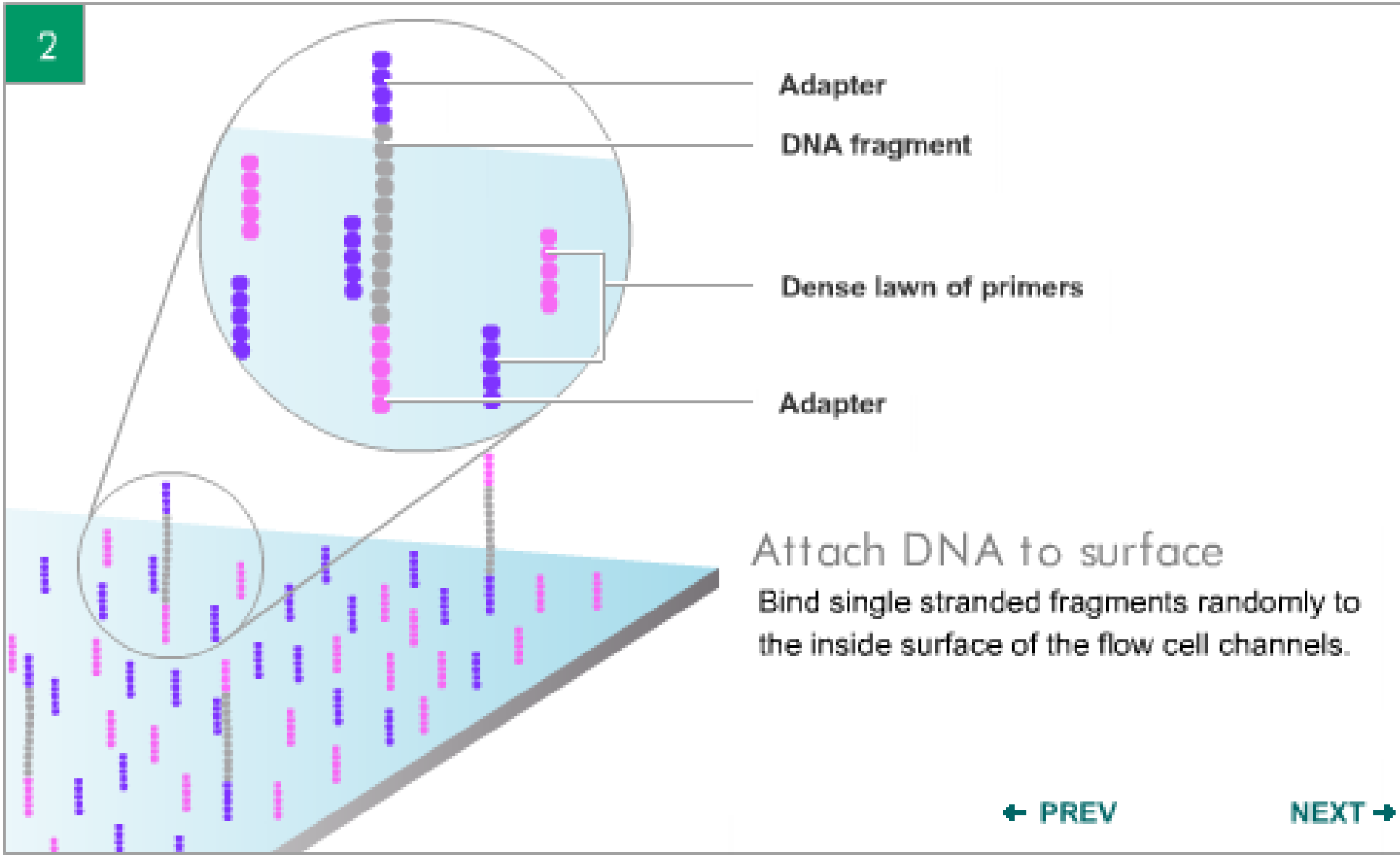


Prepare genomic DNA sample

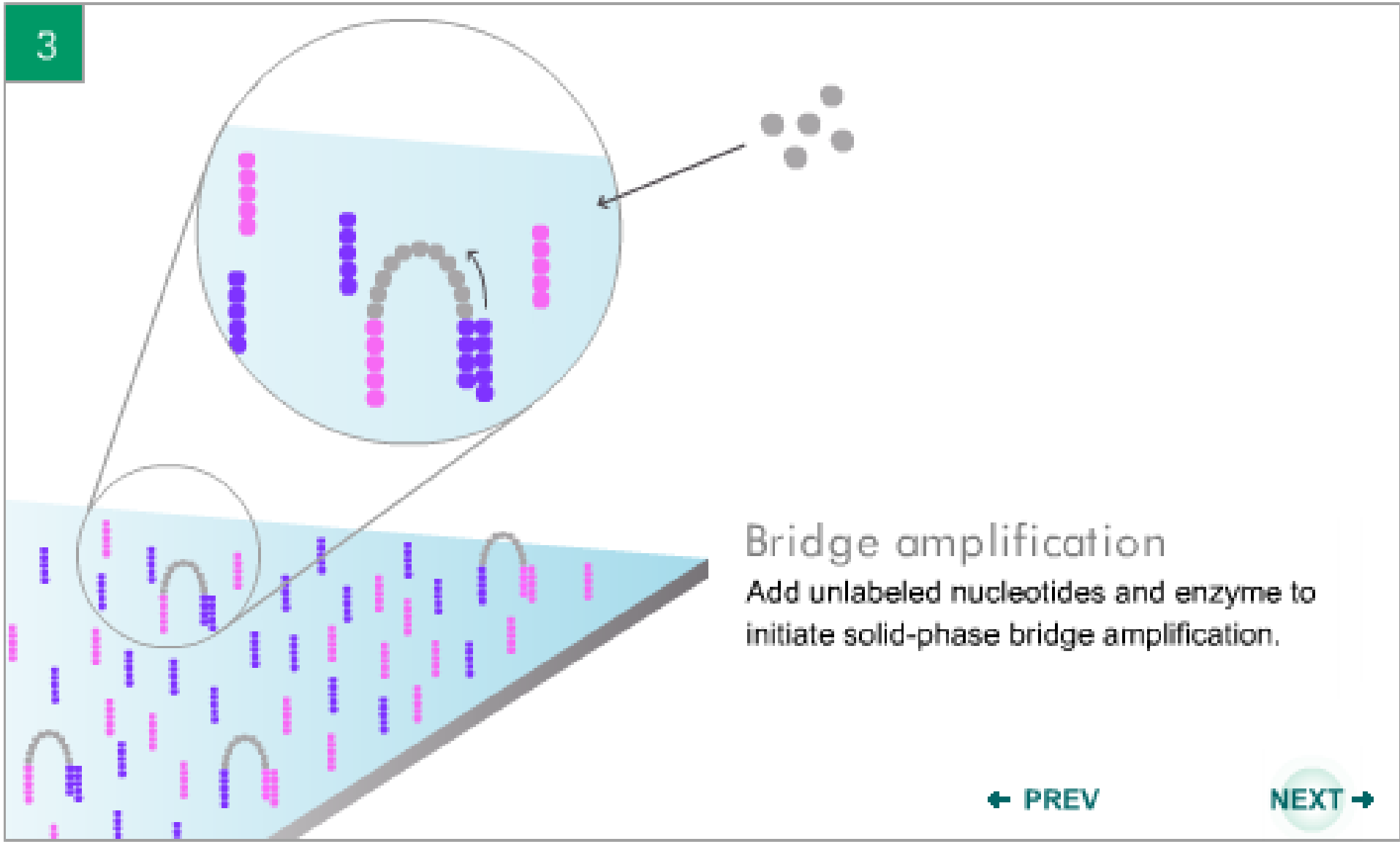
Randomly fragment genomic DNA and ligate adapters to both ends of the fragments

← PREV NEXT →

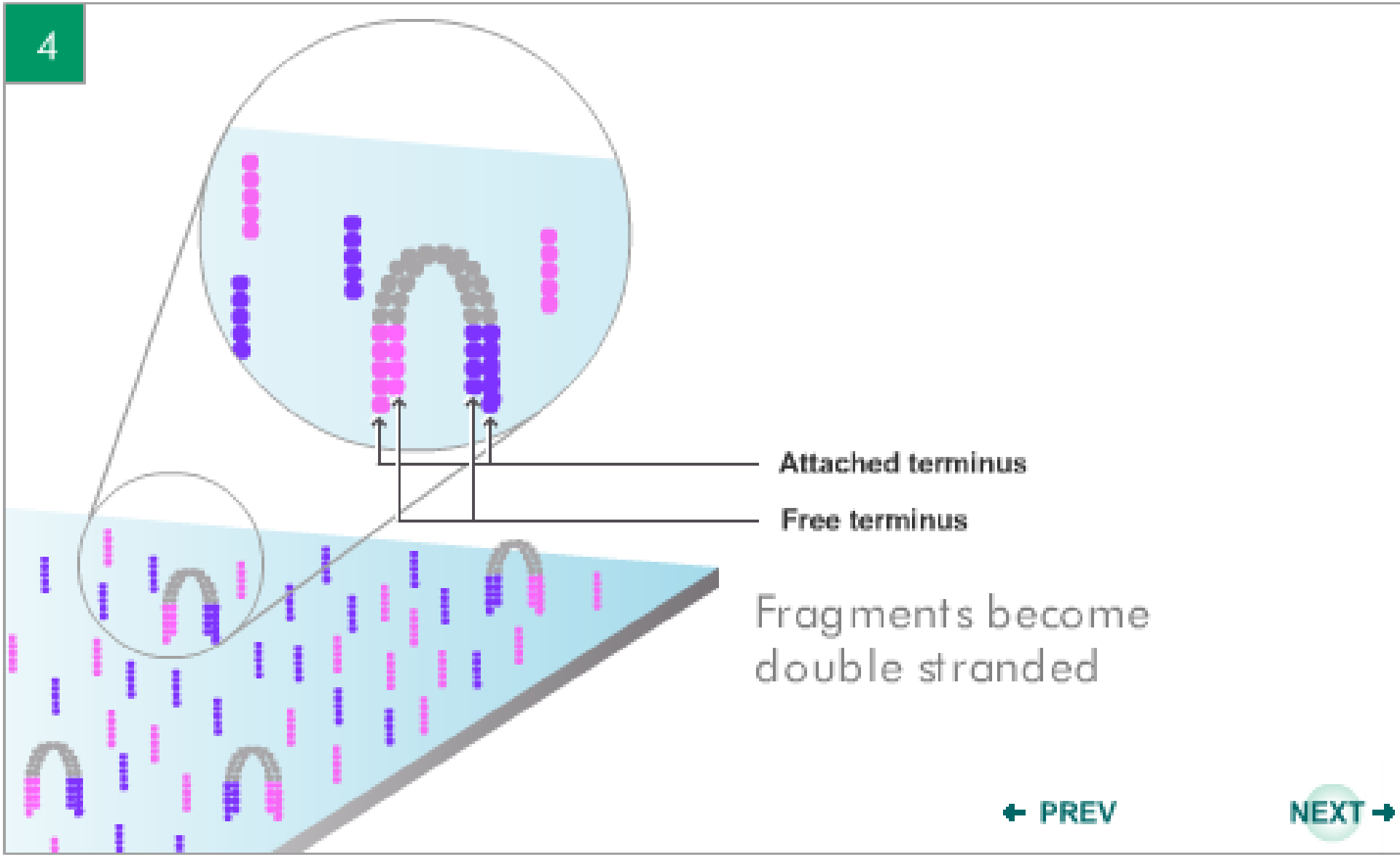
Principe du Solexa : attachement à la surface



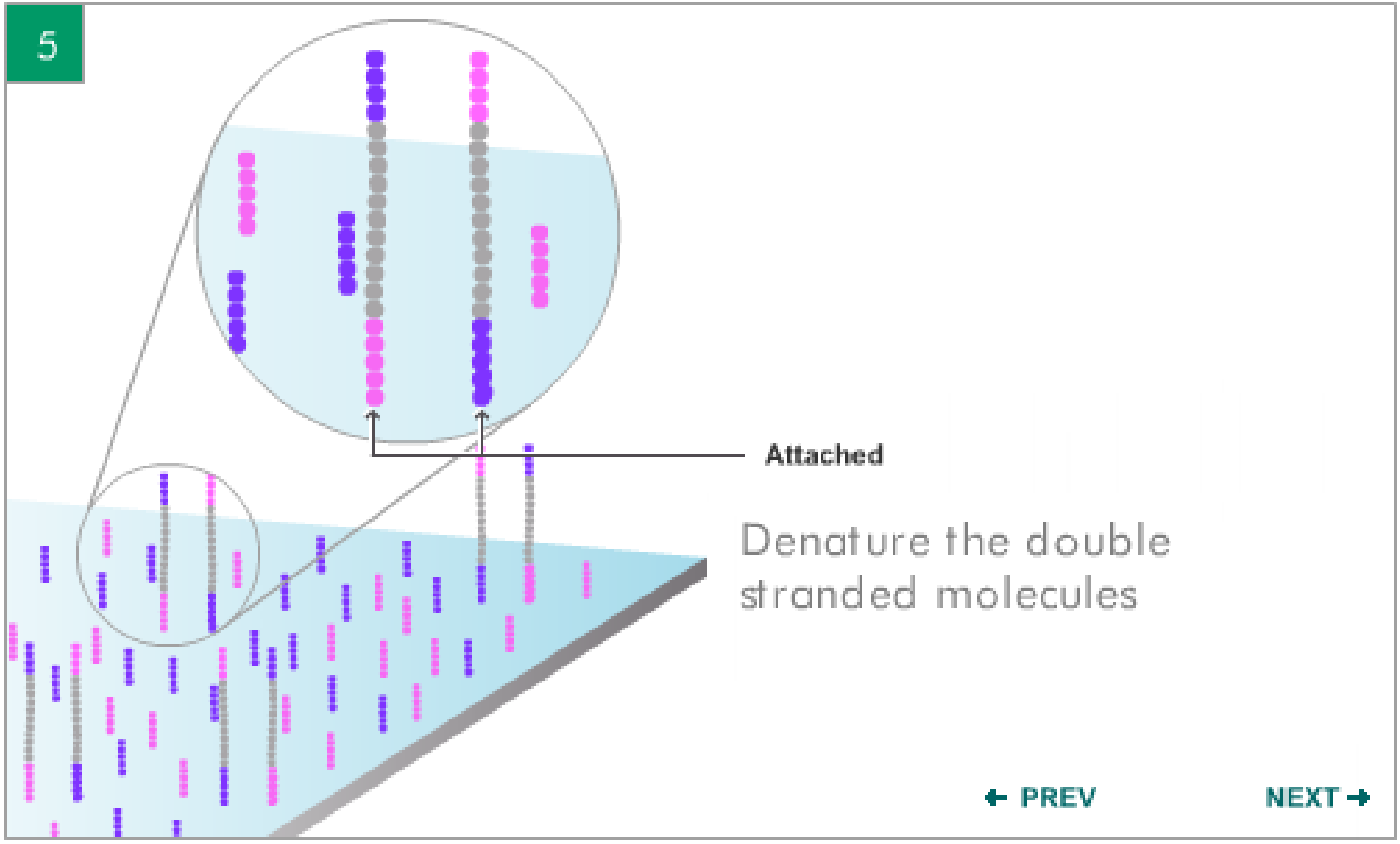
Principe du Solexa : amplification



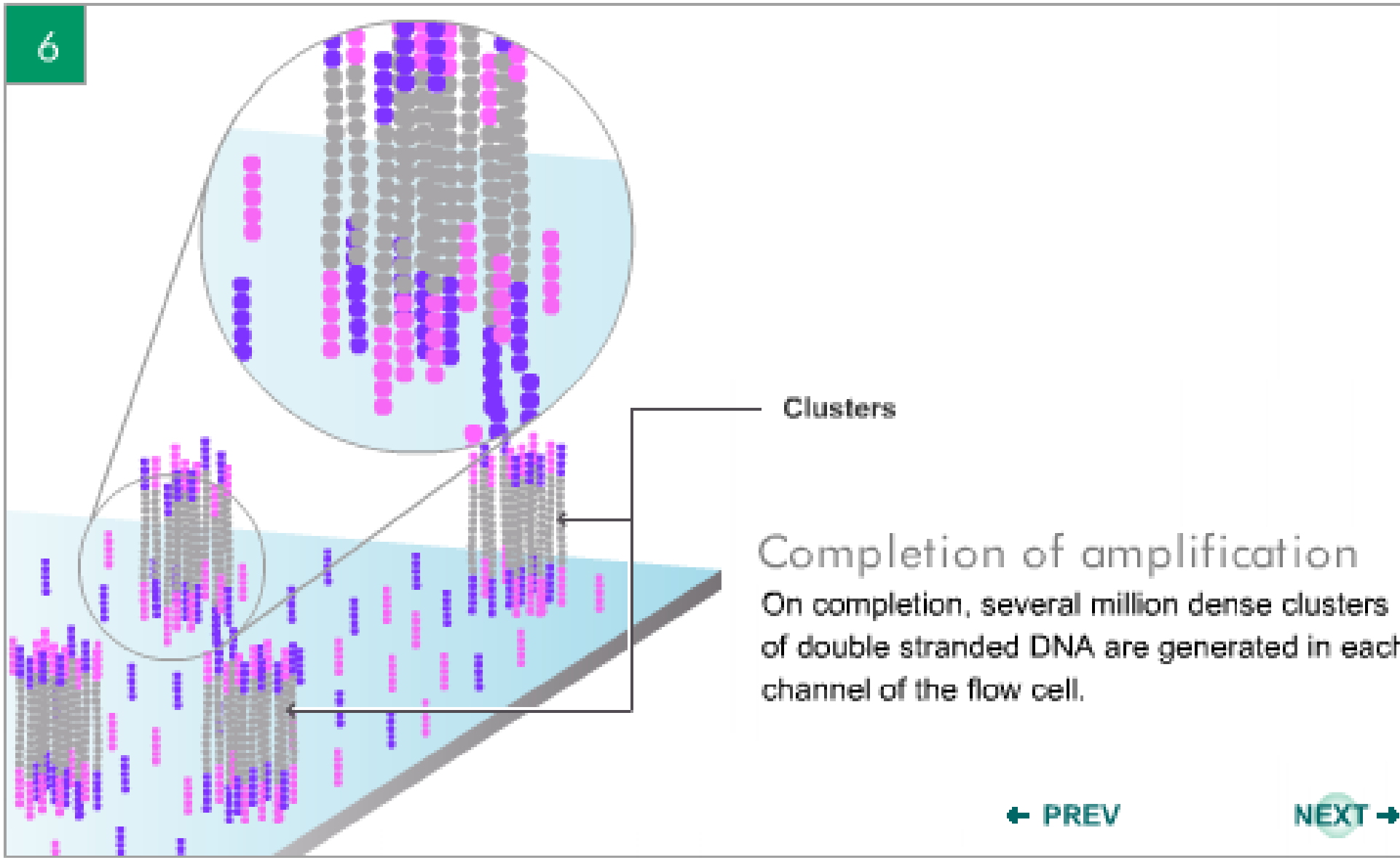
Principe du Solexa : amplification



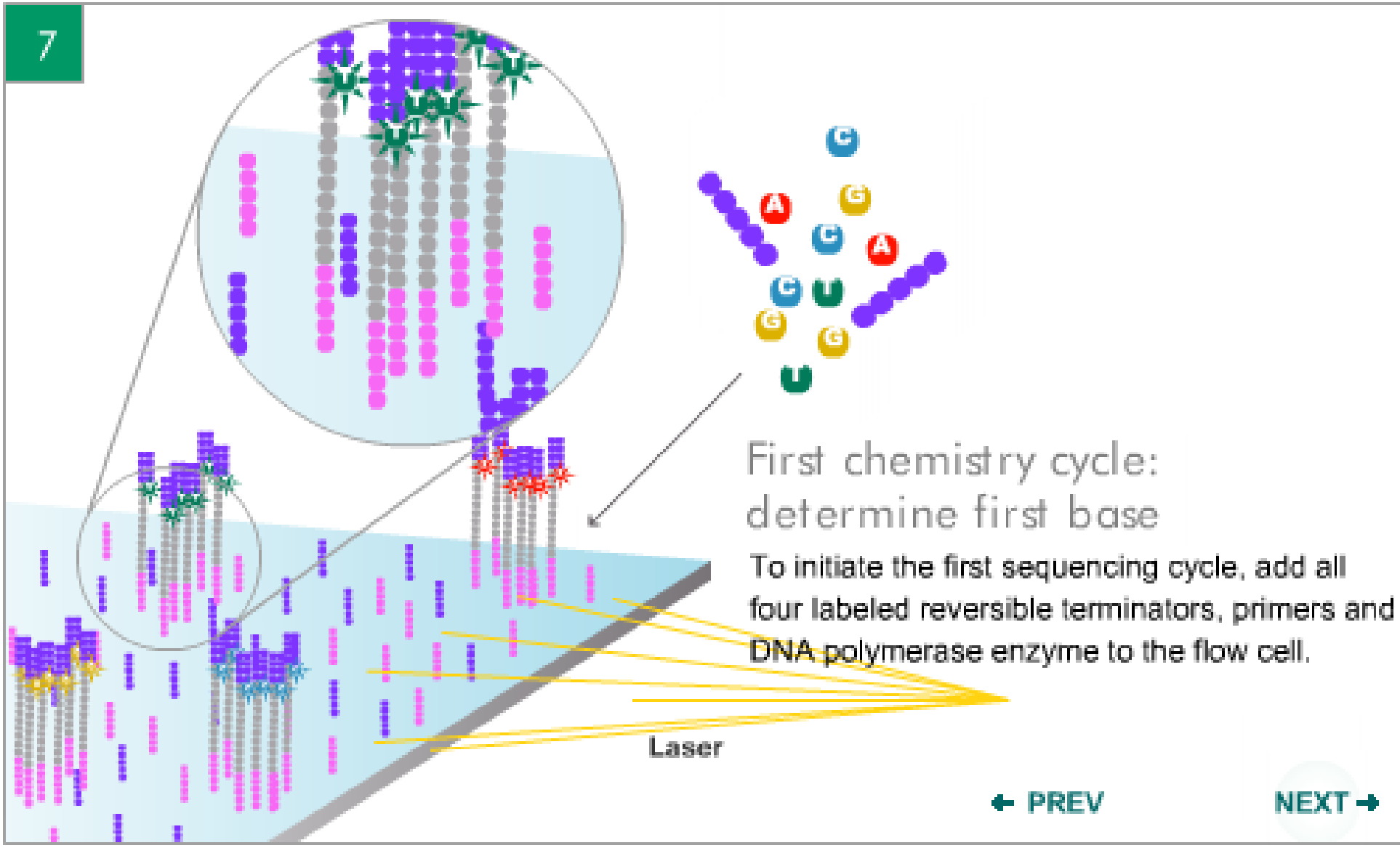
Principe du Solexa : dénaturation



Principe du Solexa : groupes fragments amplifiés



Principe du Solexa : extension 1ère base



Principe du Solexa : lecture

8

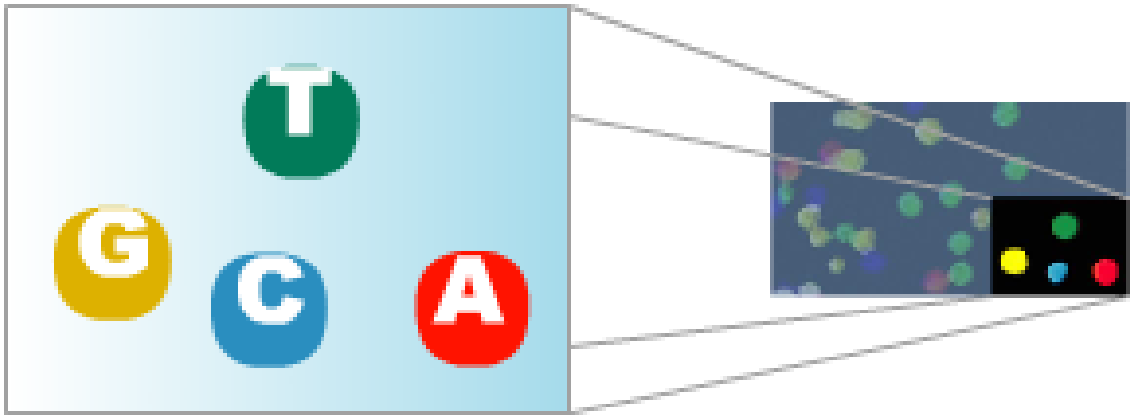


Image of first chemistry cycle

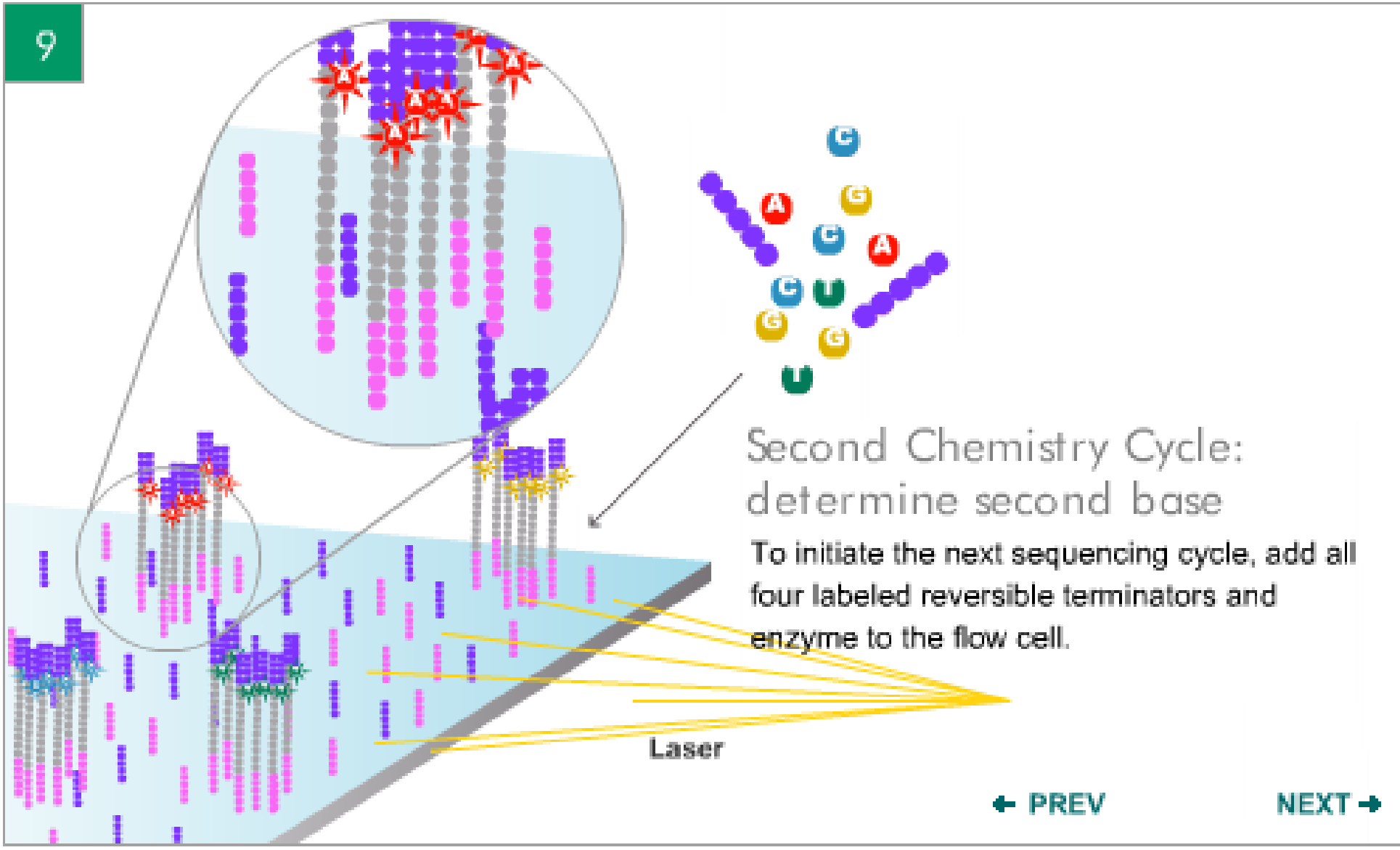
After laser excitation, capture the image of emitted fluorescence from each cluster on the flow cell. Record the identity of the first base for each cluster.

Before initiating the next chemistry cycle

The blocked 3' terminus and the fluorophore from each incorporated base are removed.

← PREV NEXT →

Principe du Solexa : extension 2ème base



Principe du Solexa : lecture

10

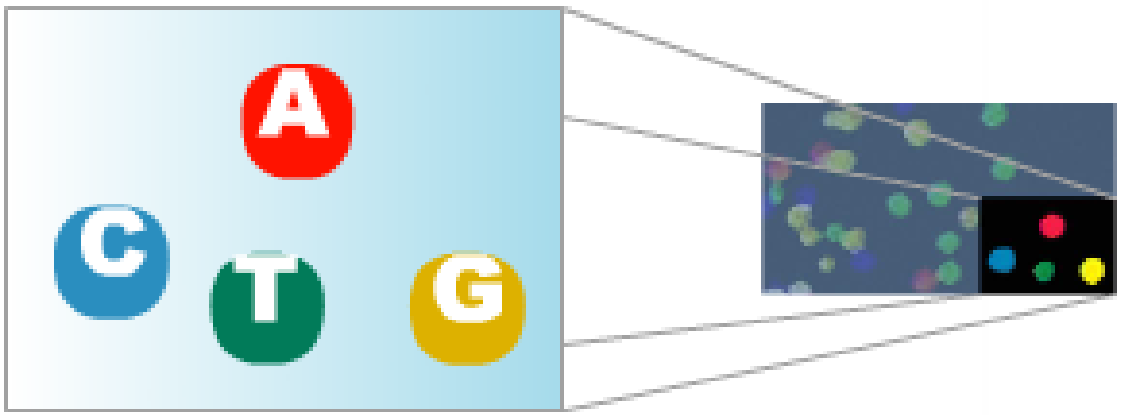


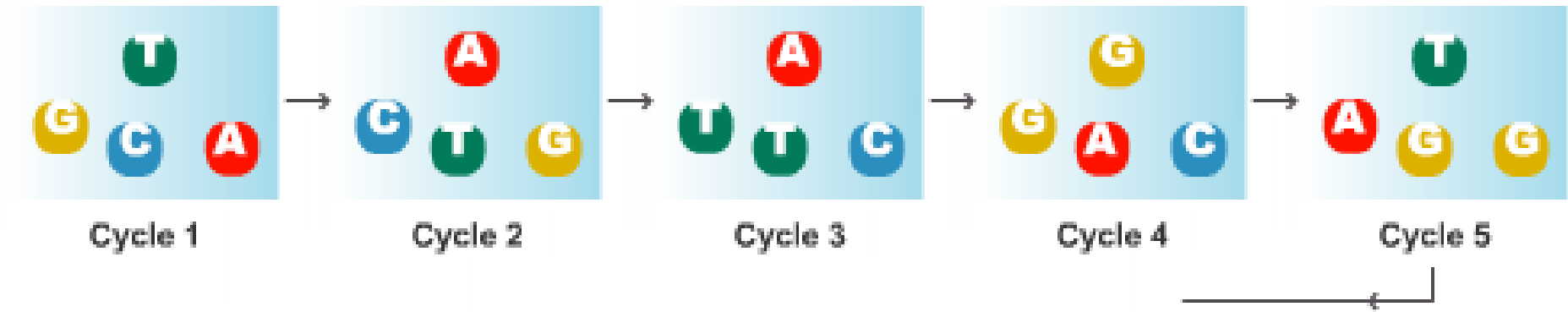
Image of second chemistry cycle is captured by the instrument

After laser excitation, collect the image data as before. Record the identity of the second base for each cluster.

PREV NEXT

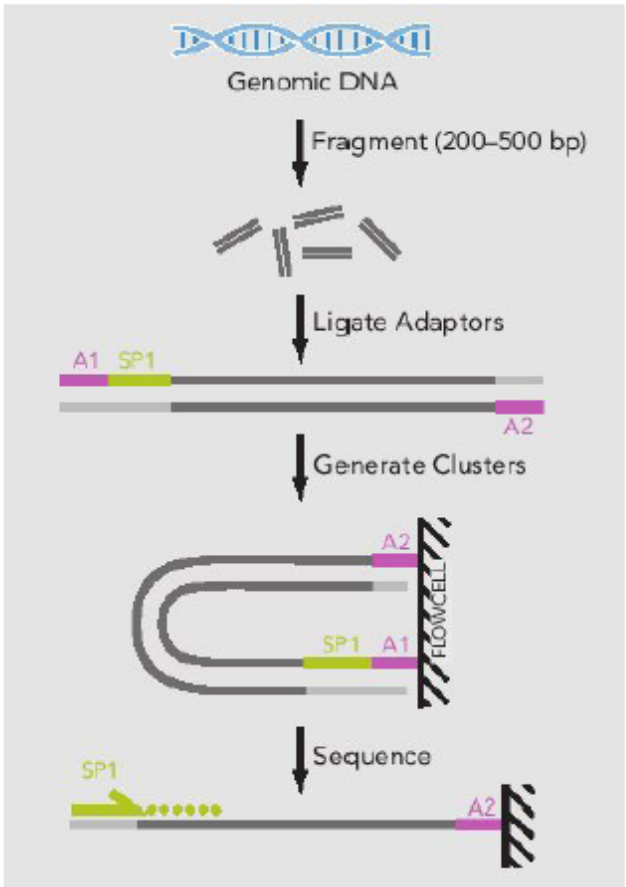
Principe du Solexa : interprétation

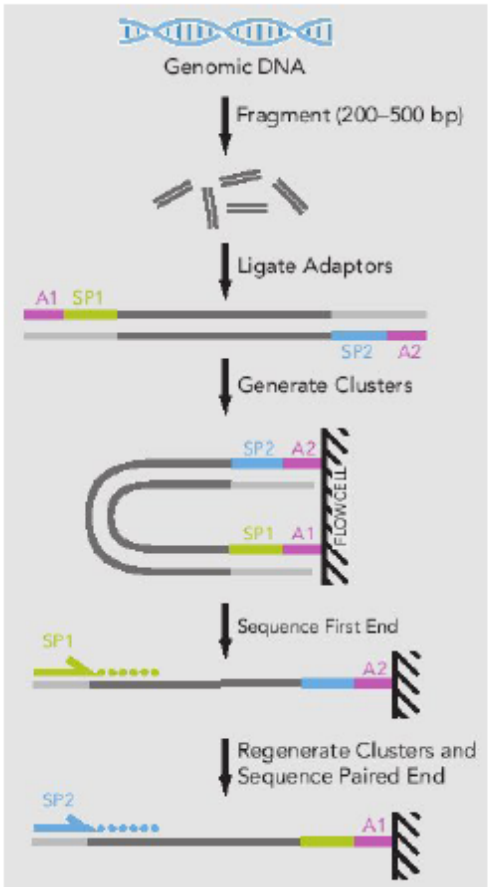
11

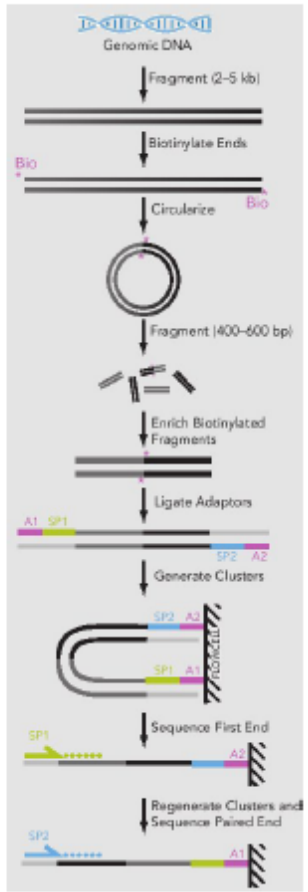


← PREV NEXT →

Single end, paired end, mate pair

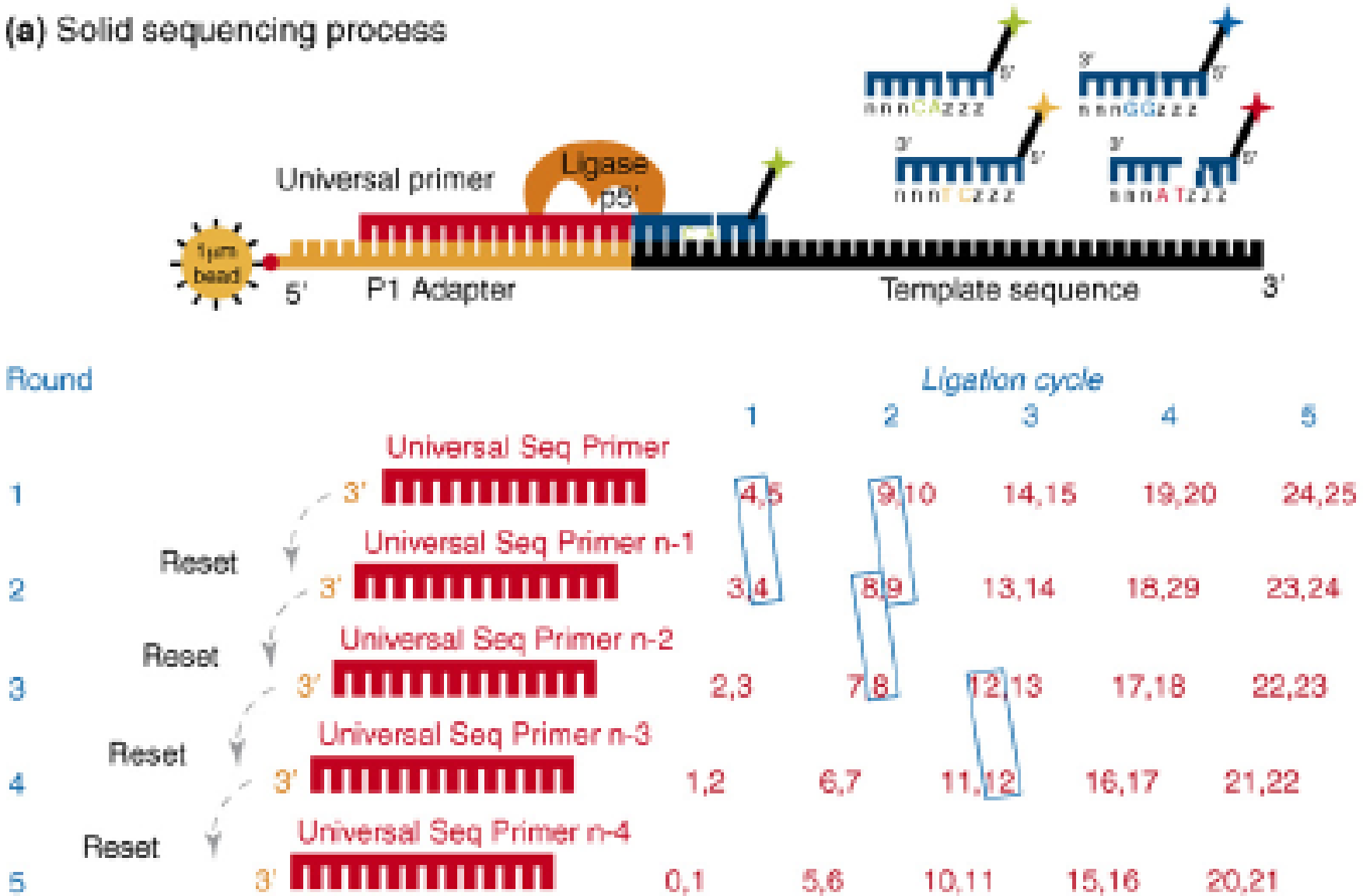






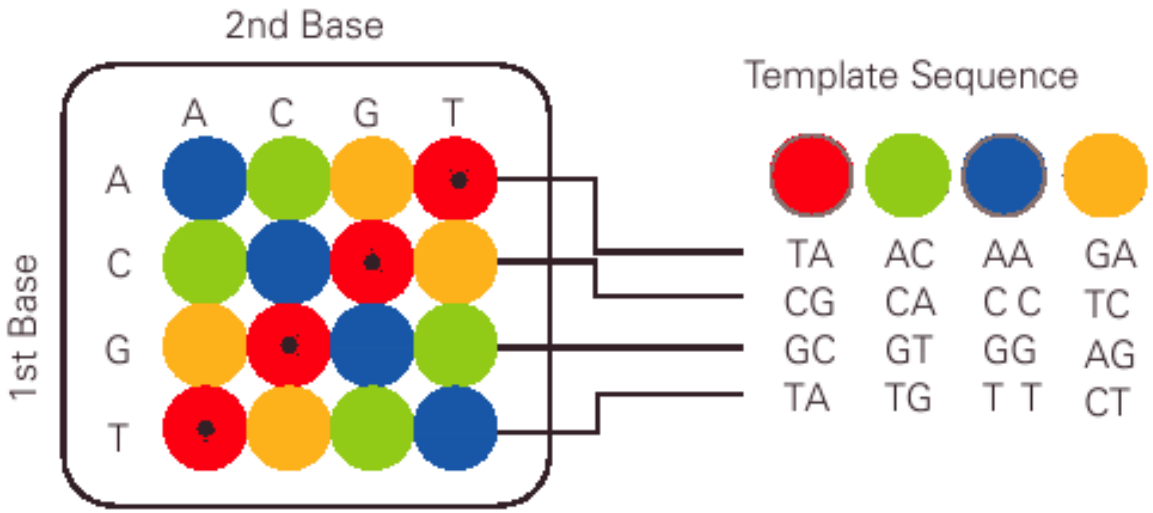
AP1/AP2: flow cell adapators; SP1/SP2: sequencing primers

Principe du SOLiD : procédure de séquençage



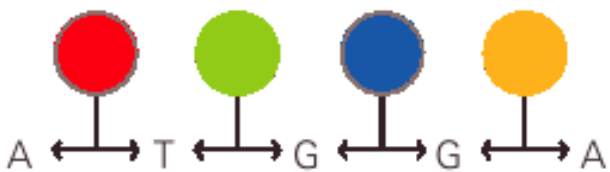
Principe du SOLiD : encodage nucléotides

Possible Dinucleotides Encoded By Each Color

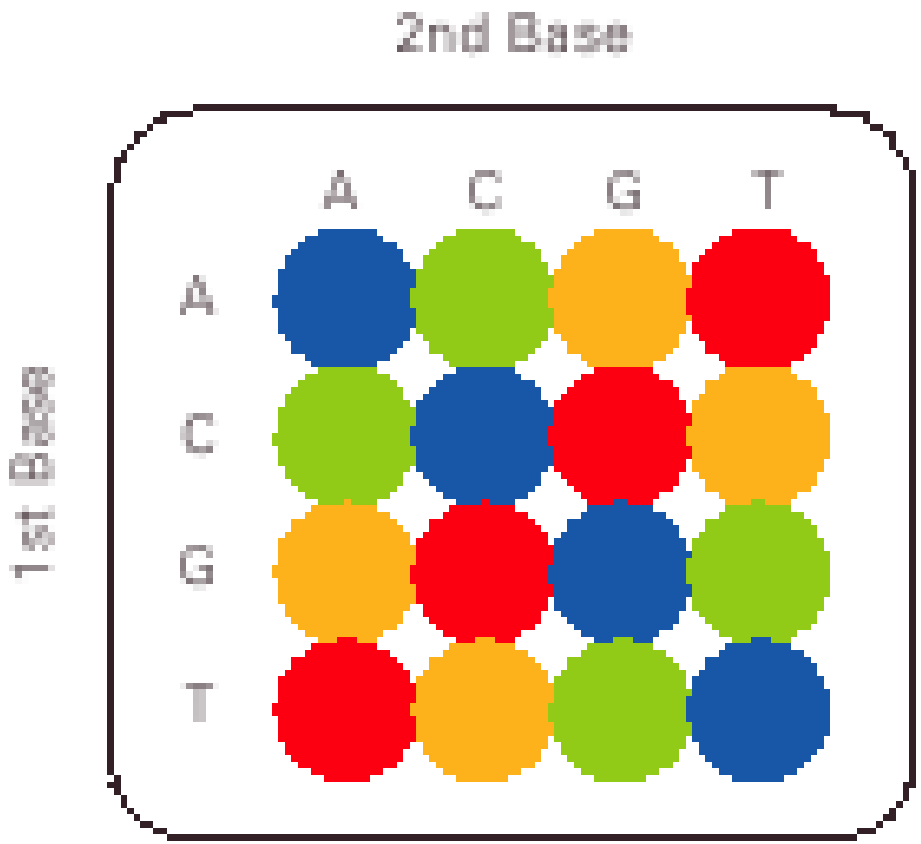


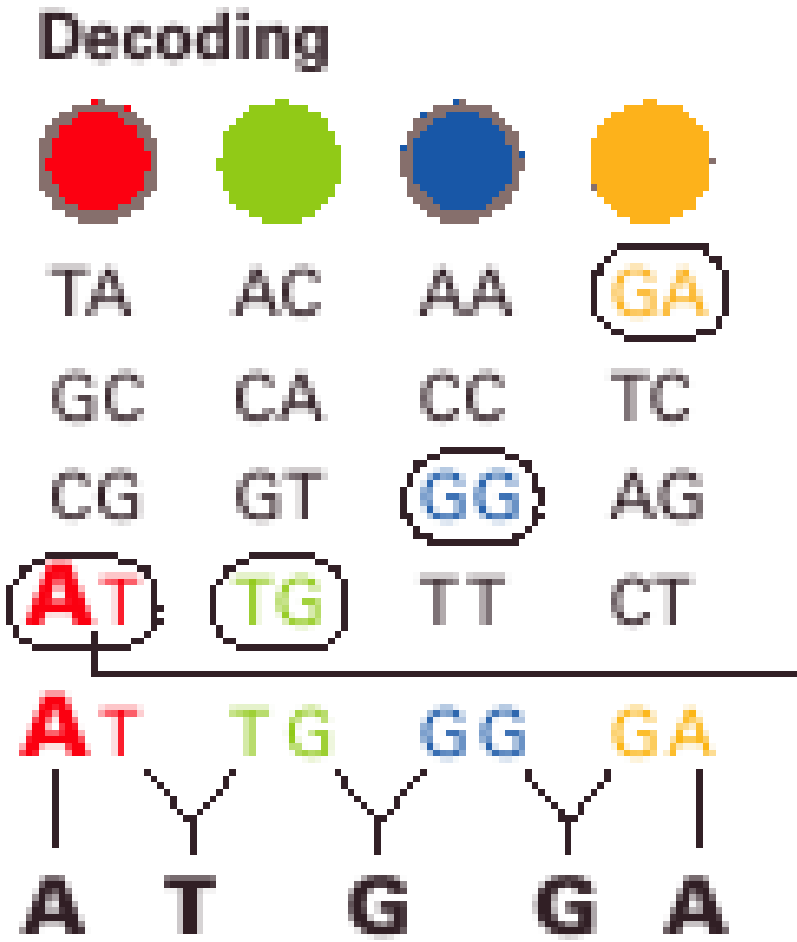
Double Interrogation

With 2 base encoding each base is defined twice



Principe du SOLiD : décodage





Principe du séquençage

ATTAGTGGTCATCCATGGCTATCGCCCGATGAGTGAGG

TCCATGGCTATCG

TCATCCATGGCTA

GGTCATCCATG

GCTATCGCCCGATGAG

CCGATGAGTGAGG

ATTAGTGGTCATCCA

GGCTATCGCCCGA

TAGTGGTCATCCA

ATTAG?????ATCCA

TAGTG???ATCCA

GGTC??CCATG

TCCATGGCTATCG

GCTAT??????ATGAG

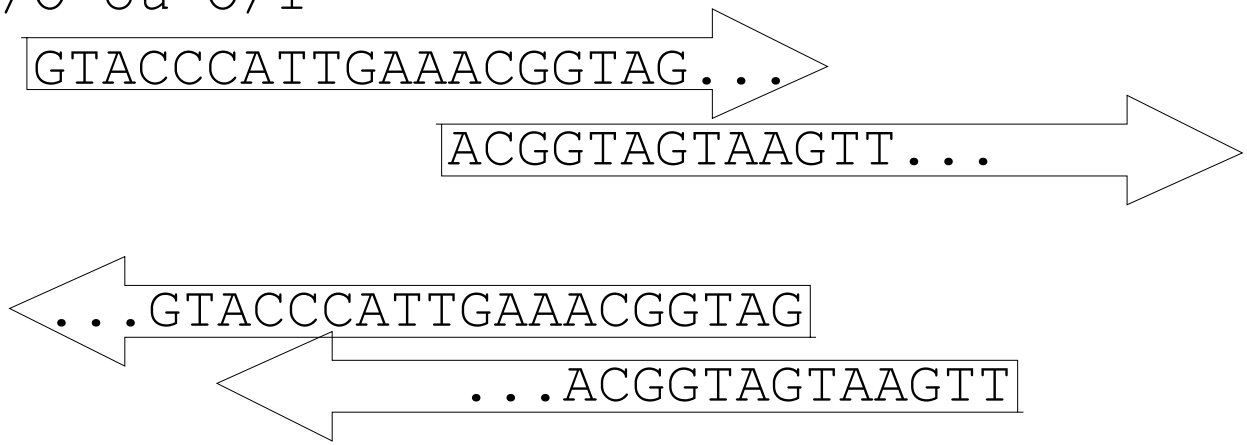
TCATC???GGCTA

GGCTA???CCCGA

CCGAT???TGAGG

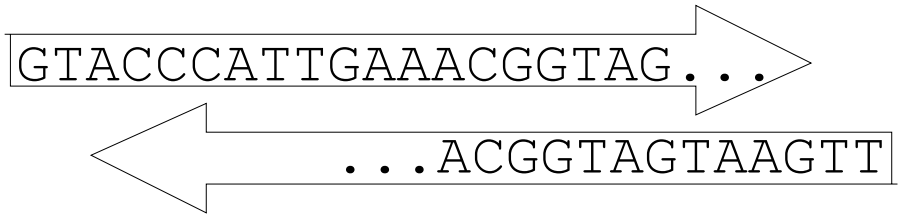
Recouvrements

1) I/O ou O/I

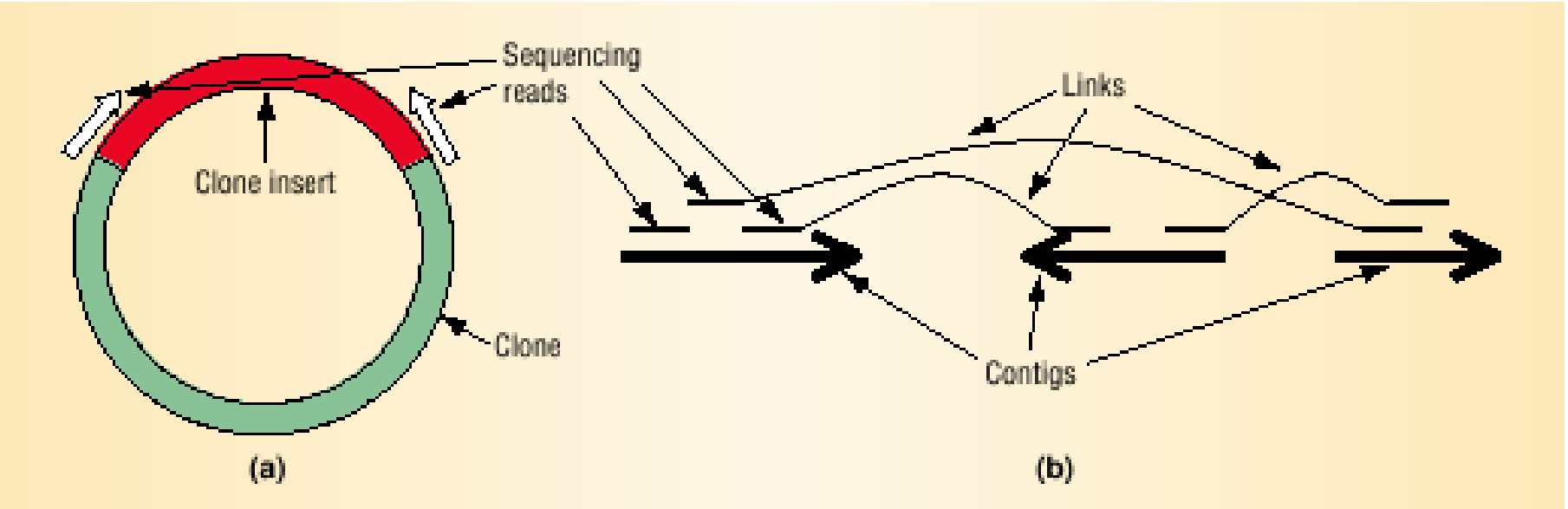


2) O/O

3) I/I



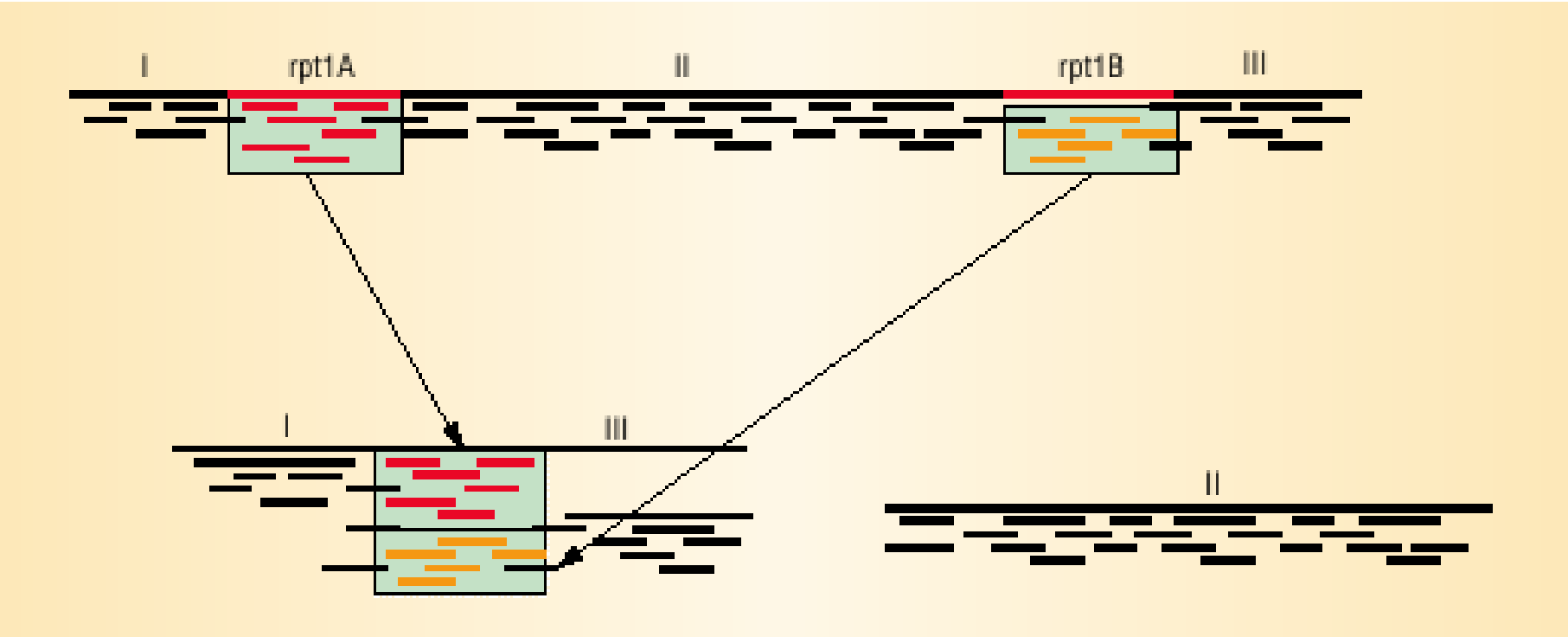
Reads and scaffolds



Répétitions

En bleu répétitions supérieures à la taille d'un read Sanger (900 bp) en rouge répétitions supérieures à la taille d'un read SOLiD (30 bp).

Répétitions



Les répétitions occasionnent des erreurs d'assemblage. ◀

Plus courte super-chaîne commune

- À partir d'un ensemble de chaînes de caractères $\{s_1, s_2, \ldots\}$ trouver la chaîne T minimale telle que toutes les chaînes s_i soient des sous-chaînes de T.
- Exemple : pour AABBA, BABAA, BAAAA, AAABA, ABBAB on a : T = AABBABAAAABA
- Problème NP-complet
- Peut être néanmoins résolu par un algorithme glouton :
  1. calculer toutes les superpositions entre les chaînes et assigner un score : $w_{i,j} = L(s_i) + L(s_j) - L(superposition(s_i, s_j))$
  2. superposer les chaînes itérativement en fonction du score
- ne prend pas en compte information non locale
- nécessite 1Gb de mémoire vive (couverture 8X) pour chaque Mbp assemblé.

Plus courte super-chaîne commune

- À partir d'un ensemble de chaînes de caractères $\{s_1, s_2, \ldots\}$ trouver la chaîne T minimale telle que toutes les chaînes s_i soient des sous-chaînes de T.
- Exemple : pour AABBA, BABAA, BAAAA, AAABA, ABBAB on a : T = AABBABAAAABA
- Problème NP-complet
- Peut être néanmoins résolu par un algorithme glouton :
  1. calculer toutes les superpositions entre les chaînes et assigner un score : $w_{i,j} = L(s_i) + L(s_j) - L(superposition(s_i, s_j))$
  2. superposer les chaînes itérativement en fonction du score
- ne prend pas en compte information non locale
- nécessite 1Gb de mémoire vive (couverture 8X) pour chaque Mbp assemblé.

Plus courte super-chaîne commune

- À partir d'un ensemble de chaînes de caractères $\{s_1, s_2, \ldots\}$ trouver la chaîne T minimale telle que toutes les chaînes s_i soient des sous-chaînes de T.
- Exemple : pour AABBA, BABAA, BAAAA, AAABA, ABBAB on a : T = AABBABAAAABA
- Problème NP-complet
- Peut être néanmoins résolu par un algorithme glouton :▶
  1. calculer toutes les superpositions entre les chaînes et assigner un score : $w_{i,j} = L(s_i) + L(s_j) - L(superposition(s_i, s_j))$
  2. superposer les chaînes itérativement en fonction du score
- ne prend pas en compte information non locale
- nécessite 1Gb de mémoire vive (couverture 8X) pour chaque Mbp assemblé.

◀

Plus courte super-chaîne commune

- À partir d'un ensemble de chaînes de caractères $\{s_1, s_2, \ldots\}$ trouver la chaîne T minimale telle que toutes les chaînes s_i soient des sous-chaînes de T.
- Exemple : pour AABBA, BABAA, BAAAA, AAABA, ABBAB on a : T = AABBABAAAABA
- Problème NP-complet
- Peut être néanmoins résolu par un algorithme glouton :▶
  1. calculer toutes les superpositions entre les chaînes et assigner un score : $w_{i,j} = L(s_i) + L(s_j) - L(superposition(s_i, s_j))$
  2. superposer les chaînes itérativement en fonction du score
- ne prend pas en compte information non locale
- nécessite 1Gb de mémoire vive (couverture 8X) pour chaque Mbp assemblé.

◀

Chemin hamiltonien dans un graphe

- Construire un graphe où :
  - les sommets sont les lectures (s_i)
  - les arêtes sont les scores de superposition : $w_{i,j}$
  - l'assemblage est le chemin dans le graphe qui passe par tous les sommets une fois de poids minimal.▶
- Chaque contig génère un chemin dans le graphe
- Étapes de l'algorithme
  1. Générer tous les alignements entre les lectures
  2. Nettoyer le graphe (lectures incluses dans d'autres, lectures transitives, lectures ayant un recouvrement trop faible), etc.
  3. on obtient ensemble de chemins qui ne se coupent pas (contigs).
  4. génération d'une séquence consensus par alignement multiple des lectures d'une chemin donné.
- L'algorithme est NP-complet

◀

Chemin hamiltonien dans un graphe

- Construire un graphe où :
  - les sommets sont les lectures (s_i)
  - les arêtes sont les scores de superposition : $w_{i,j}$
  - l'assemblage est le chemin dans le graphe qui passe par tous les sommets une fois de poids minimal.▶
- Chaque contig génère un chemin dans le graphe
- Étapes de l'algorithme
  1. Générer tous les alignements entre les lectures
  2. Nettoyer le graphe (lectures incluses dans d'autres, lectures transitives, lectures ayant un recouvrement trop faible), etc.
  3. on obtient ensemble de chemins qui ne se coupent pas (contigs).
  4. génération d'une séquence consensus par alignement multiple des lectures d'une chemin donné.
- L'algorithme est NP-complet

◀

Chemin hamiltonien dans un graphe

- Construire un graphe où :
  - les sommets sont les lectures (s_i)
  - les arêtes sont les scores de superposition : $w_{i,j}$
  - l'assemblage est le chemin dans le graphe qui passe par tous les sommets une fois de poids minimal.▶
- Chaque contig génère un chemin dans le graphe
- Étapes de l'algorithme
  1. Générer tous les alignements entre les lectures
  2. Nettoyer le graphe (lectures incluses dans d'autres, lectures transitives, lectures ayant un recouvrement trop faible), etc.
  3. on obtient ensemble de chemins qui ne se coupent pas (contigs).
  4. génération d'une séquence consensus par alignement multiple des lectures d'une chemin donné.
- L'algorithme est NP-complet

◀

Chemin eulérien dans un graphe

- chemin eulérien dans un graphe de de Bruijn de dimension k.
- les sommets du graphe contiennent tous les k-mers sur un alphabet donné V.
- une arête connecte deux sommets si recouvrement k-1 ▶

- diviser toutes les lectures en k-mers recouvrants
- créer un graphe de de Bruijn
- reconstruire la molécule initiale d'ADN == trouver le chemin qui utilise toutes les arêtes.
- l'algorithme est linéaire !

◀

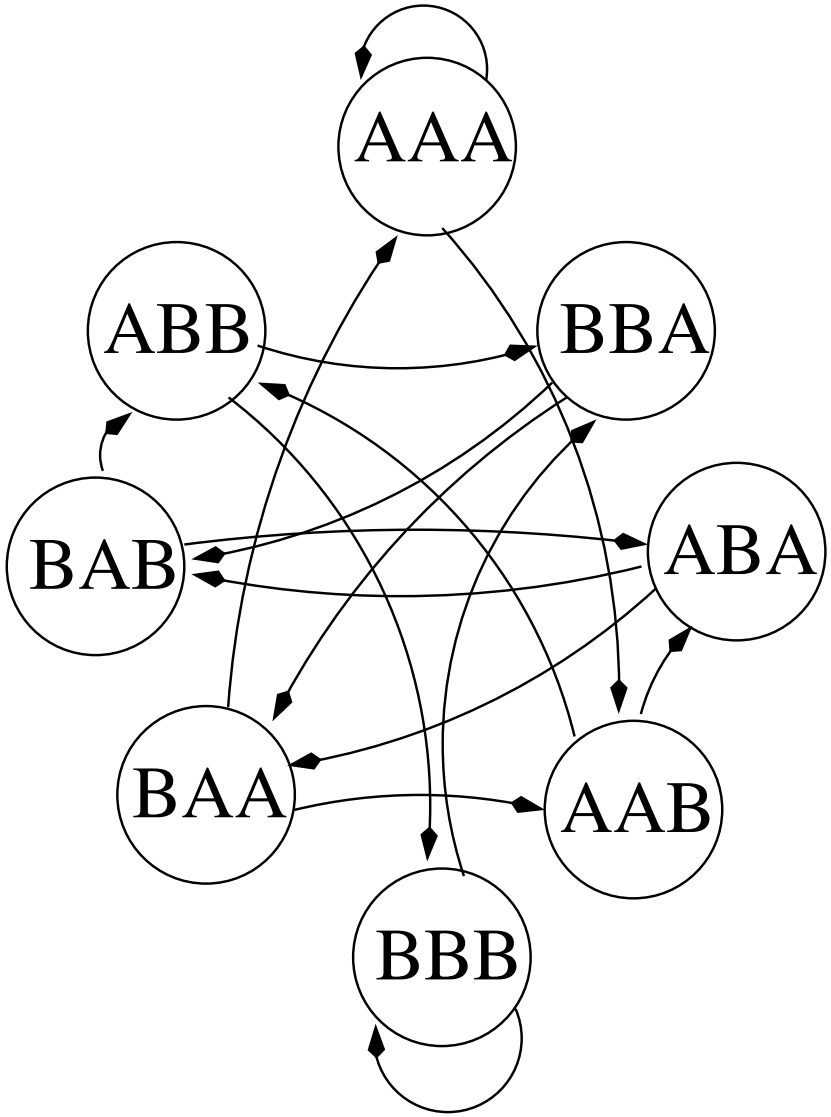
Chemin eulérien dans un graphe

- chemin eulérien dans un graphe de de Bruijn de dimension k.
- les sommets du graphe contiennent tous les k-mers sur un alphabet donné V.
- une arête connecte deux sommets si recouvrement k-1 ▶

- diviser toutes les lectures en k-mers recouvrants
- créer un graphe de de Bruijn
- reconstruire la molécule initiale d'ADN == trouver le chemin qui utilise toutes les arêtes.
- l'algorithme est linéaire !

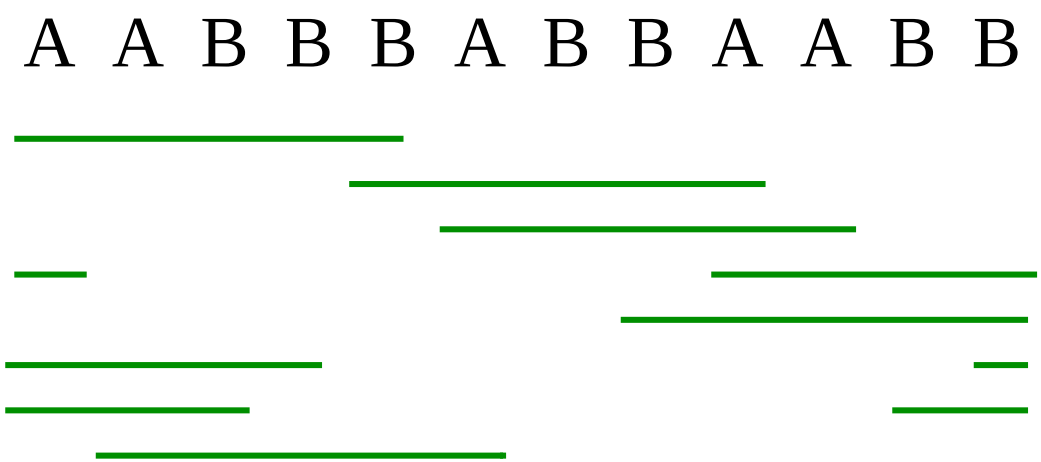
◀

Graphe de de Bruijn

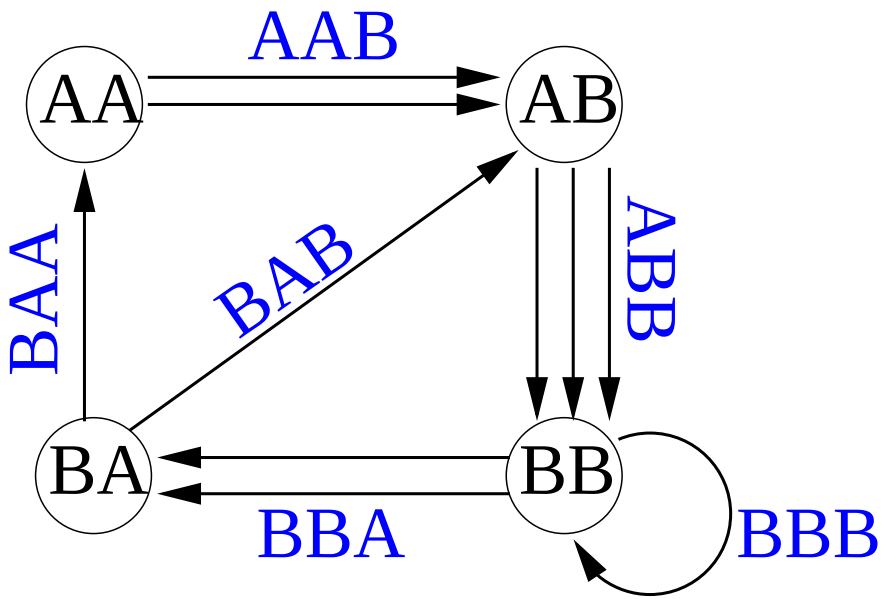


Graph de de Bruijn dimension 3 sur l'alphabet V = A,B

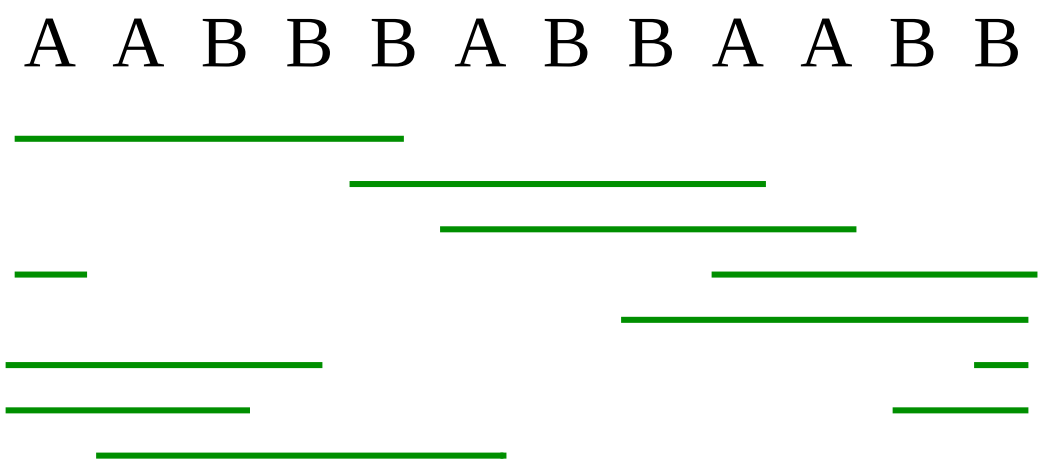
Graphe de de Bruijn : exemple 1



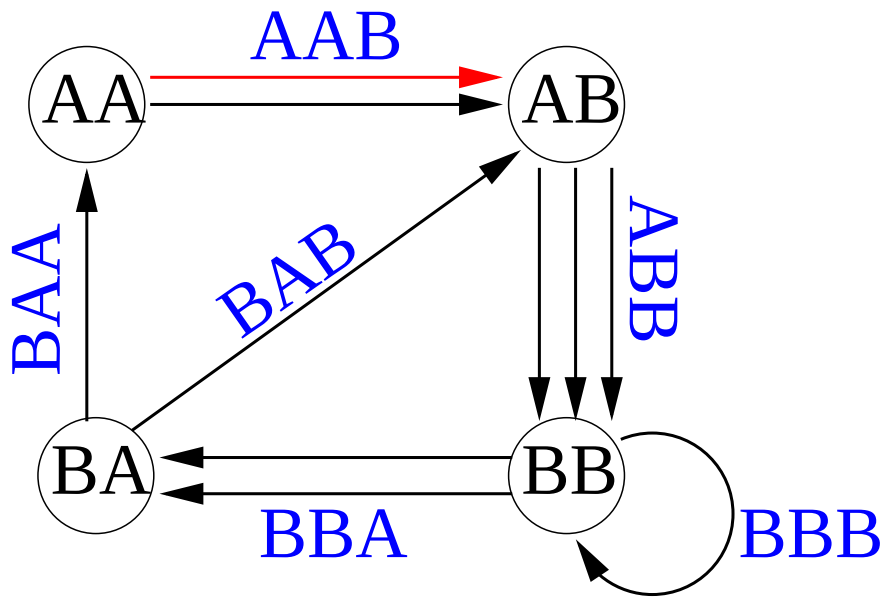
A A B 2
A B B 3
B B B
B B A 2
B A B
B A A



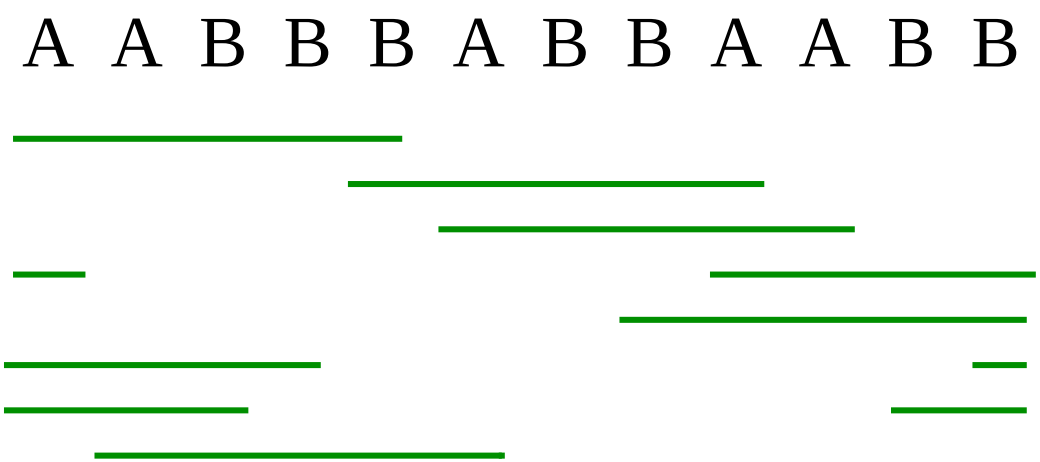
Graphe de de Bruijn : exemple 1



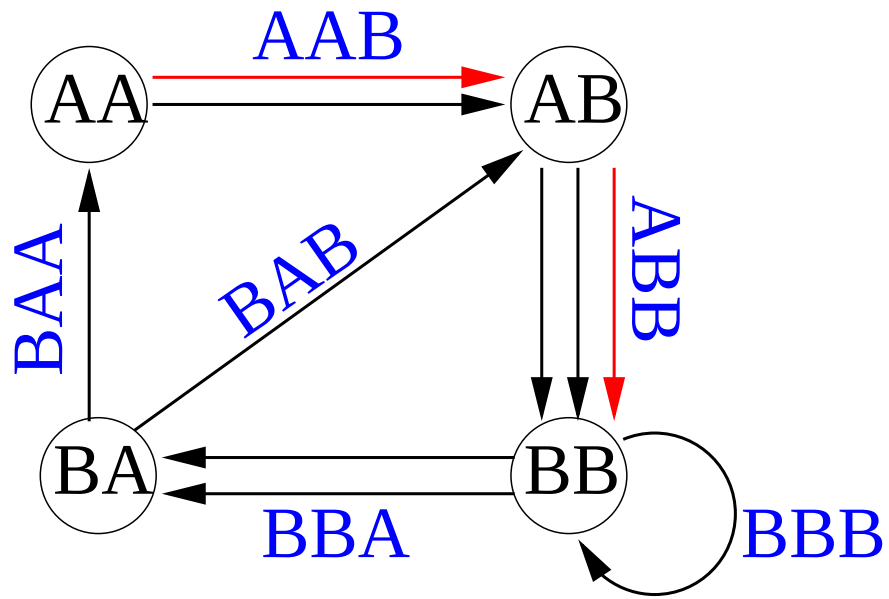
A A B 2
A B B 3
B B B
B B A 2
B A B
B A A



Graphe de de Bruijn : exemple 1

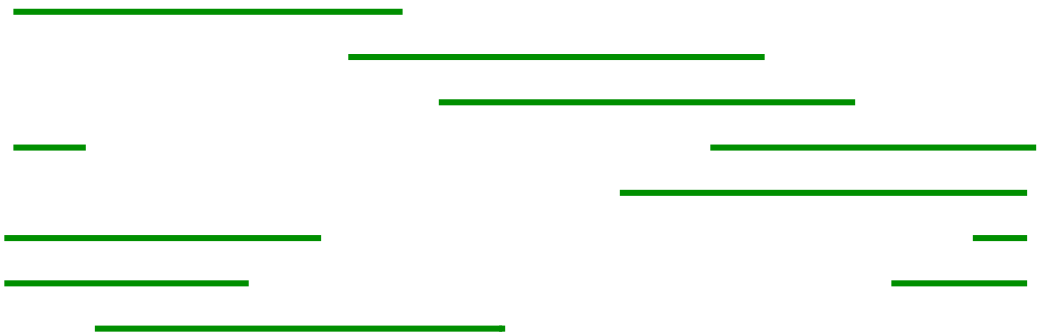


A A B 2
A B B 3
B B B
B B A 2
B A B
B A A

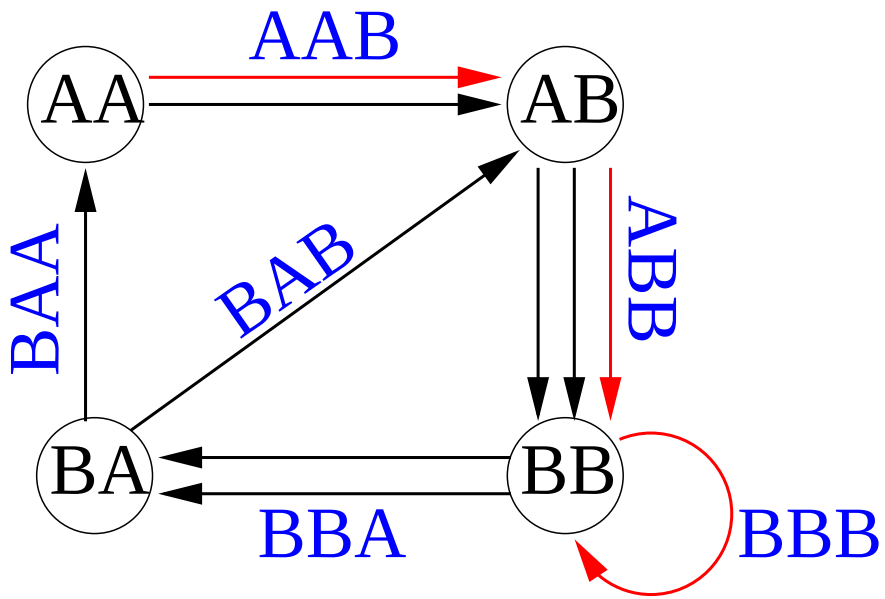


Graphe de de Bruijn : exemple 1

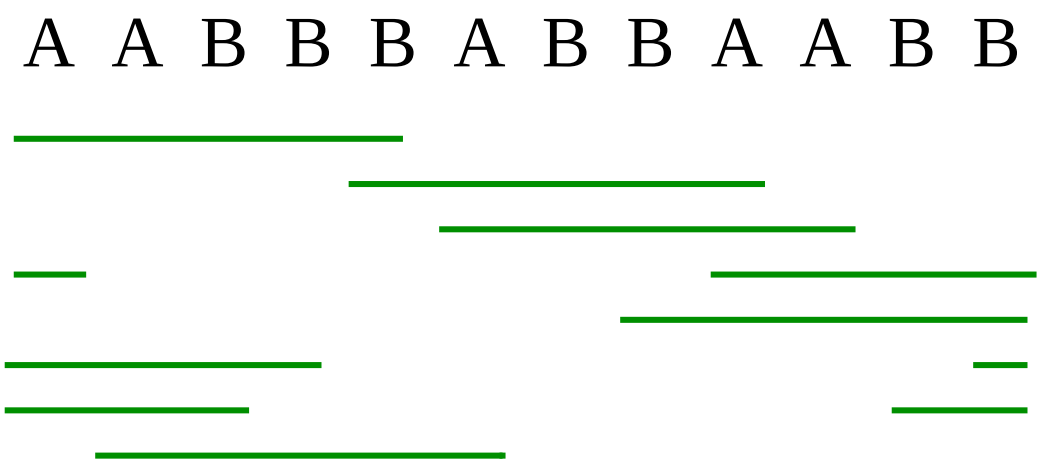
A A B B B A B B A A B B



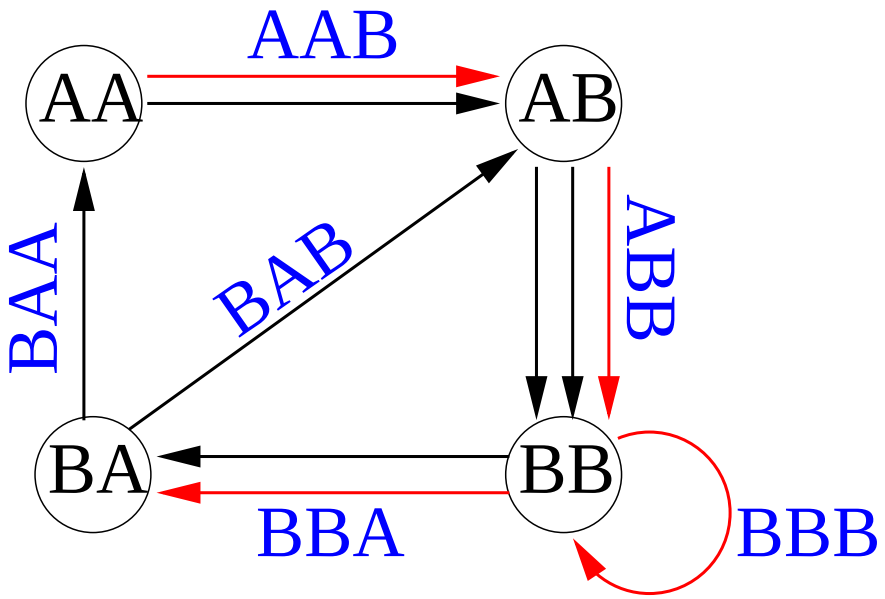
A A B 2
A B B 3
B B B
B B A 2
B A B
B A A



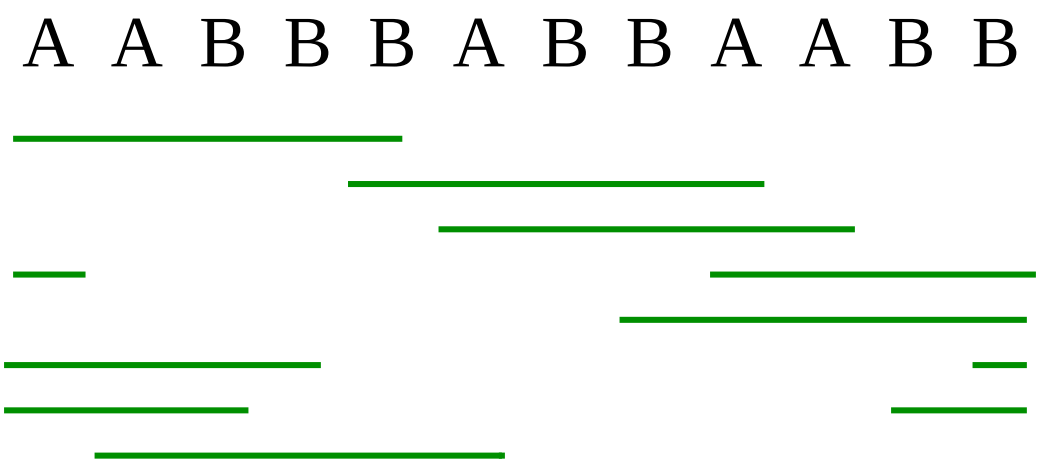
Graphe de de Bruijn : exemple 1



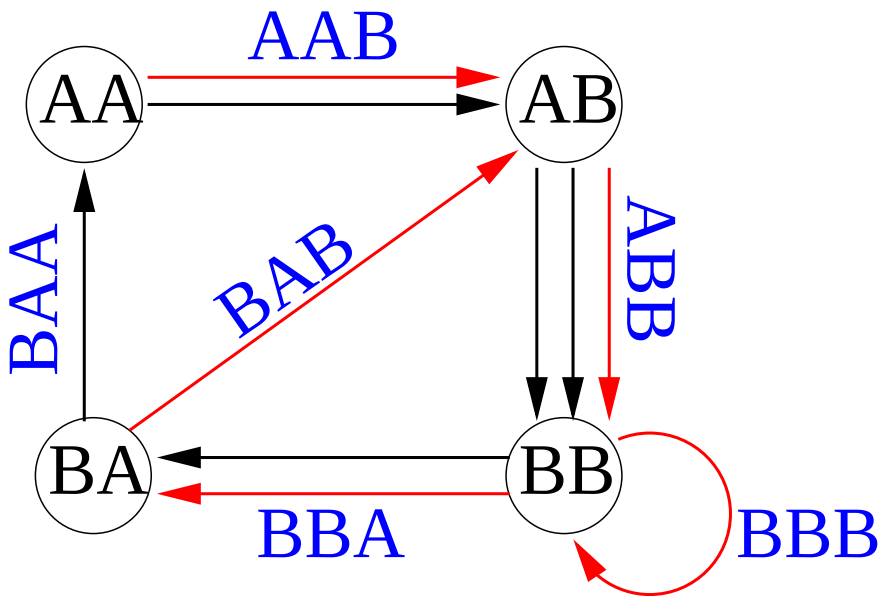
A A B 2
A B B 3
B B B
B B A 2
B A B
B A A



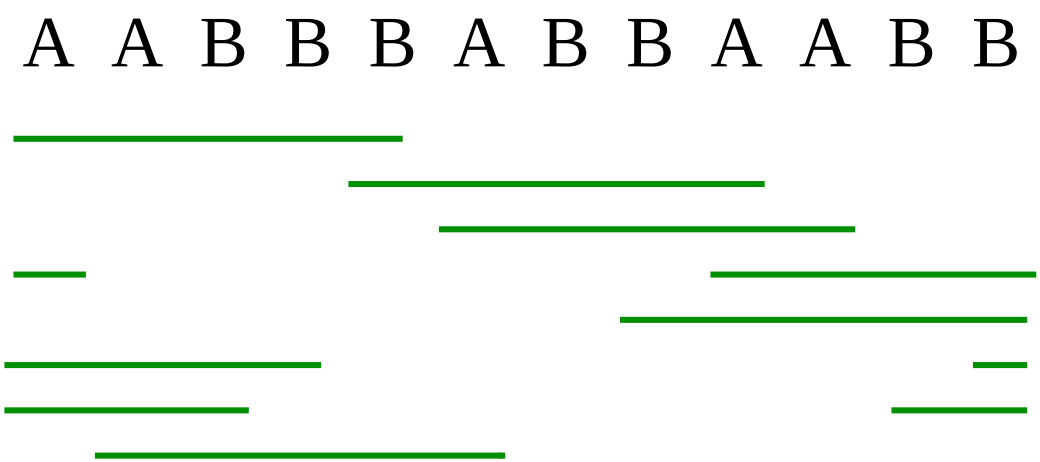
Graphe de de Bruijn : exemple 1



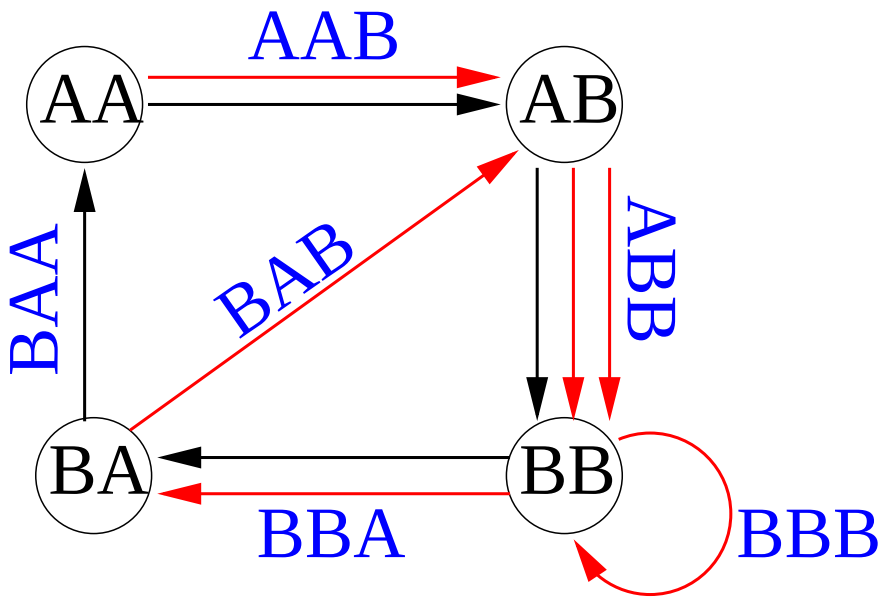
A A B 2
A B B 3
B B B
B B A 2
B A B
B A A



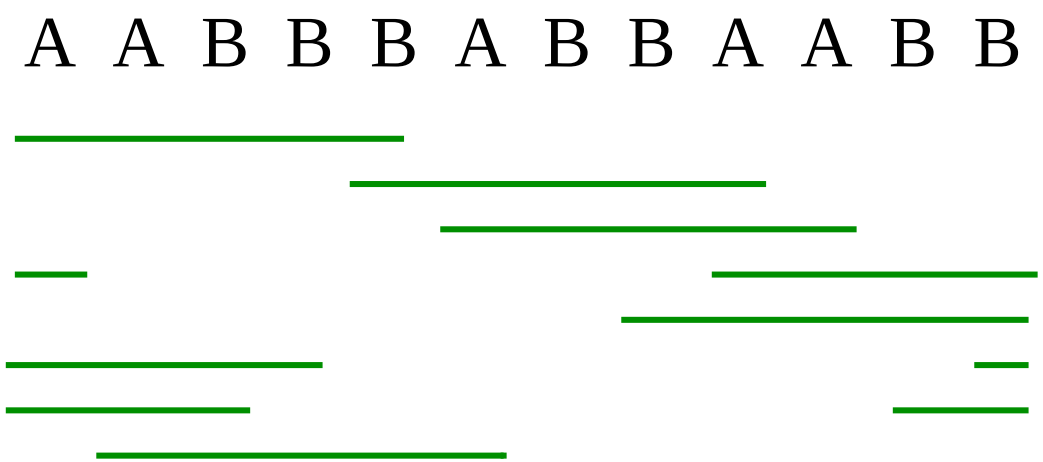
Graphe de de Bruijn : exemple 1



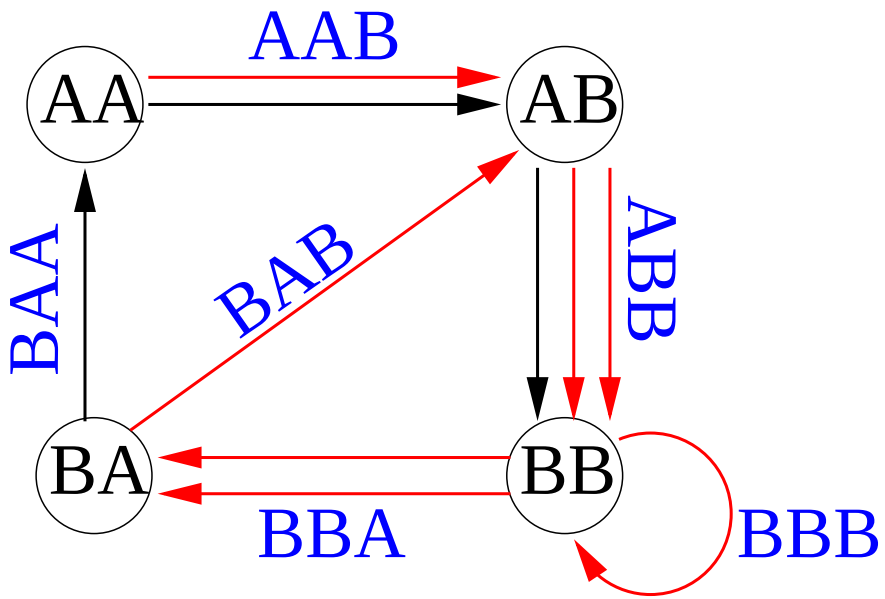
A A B 2
A B B 3
B B B
B B A 2
B A B
B A A



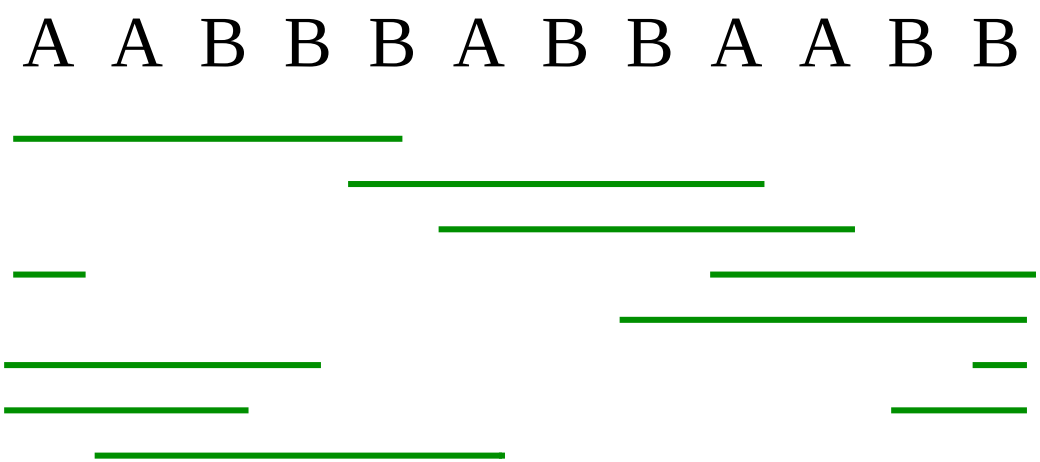
Graphe de de Bruijn : exemple 1



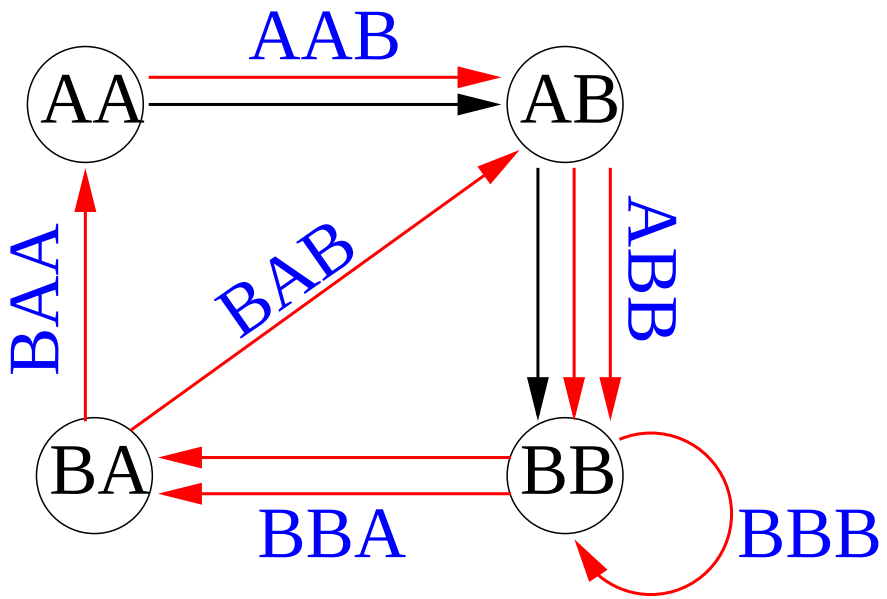
A A B 2
A B B 3
B B B
B B A 2
B A B
B A A



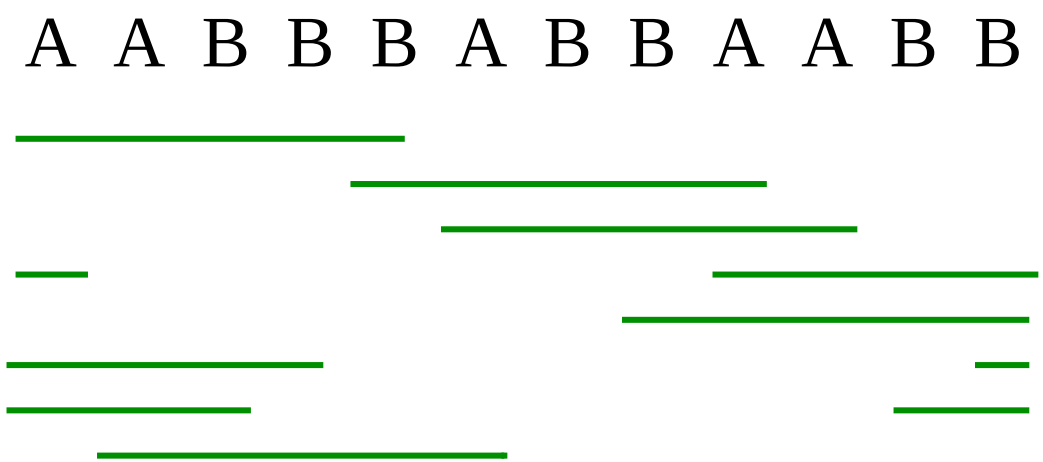
Graphe de de Bruijn : exemple 1



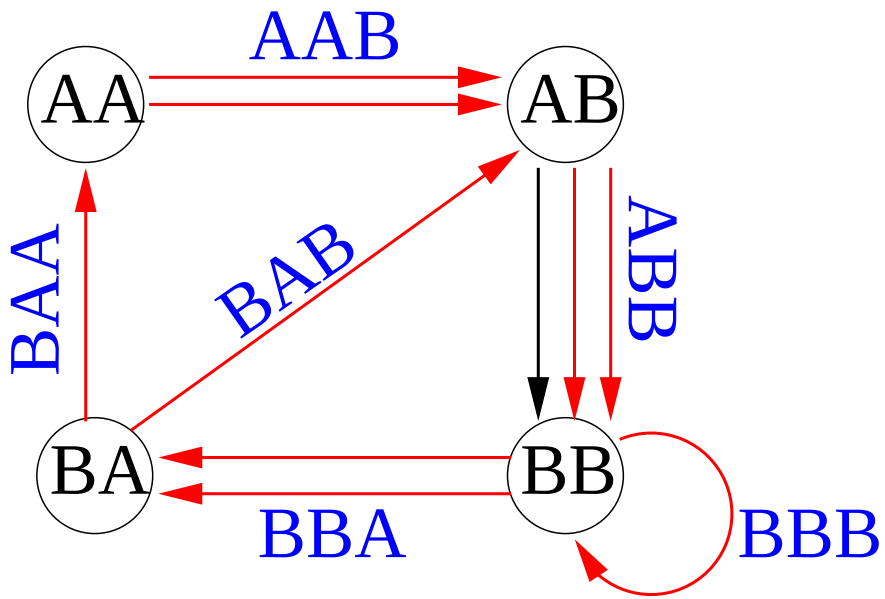
A A B 2
A B B 3
B B B
B B A 2
B A B
B A A



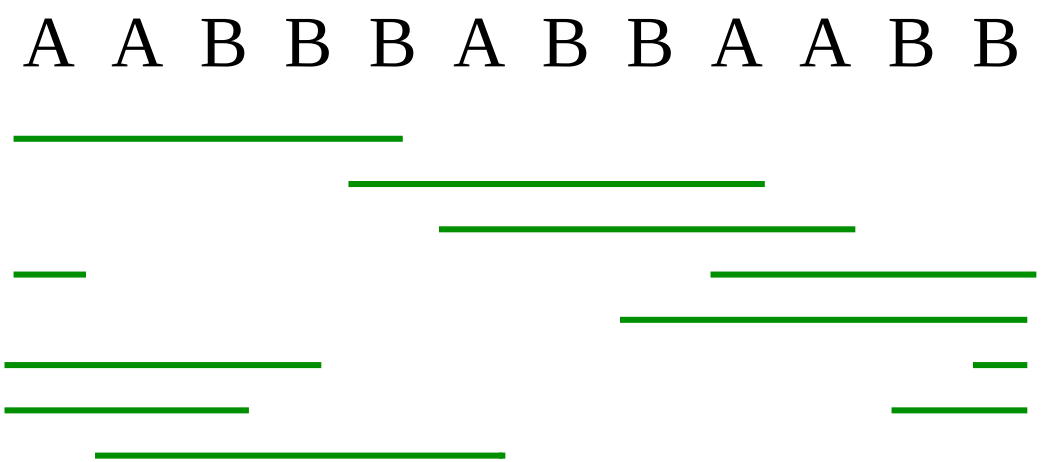
Graphe de de Bruijn : exemple 1



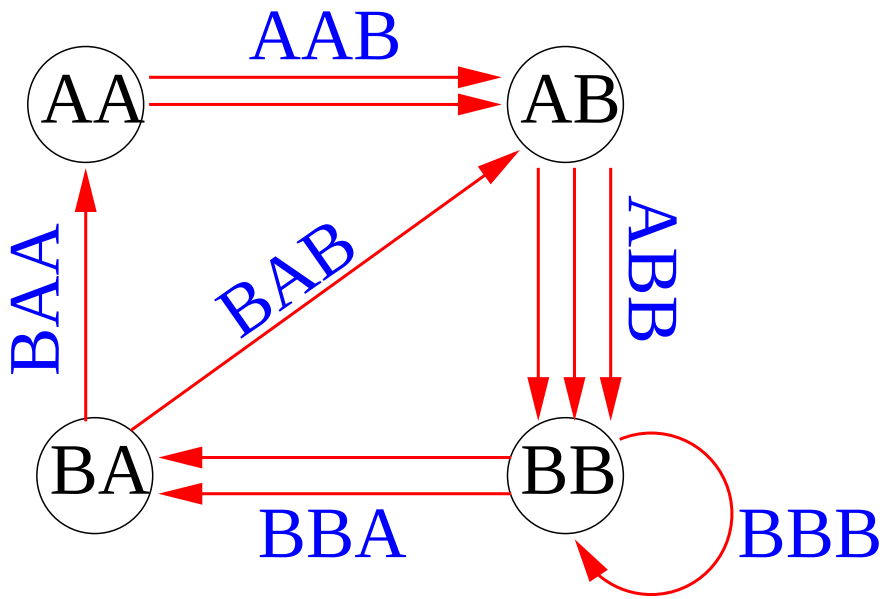
A A B 2
A B B 3
B B B
B B A 2
B A B
B A A



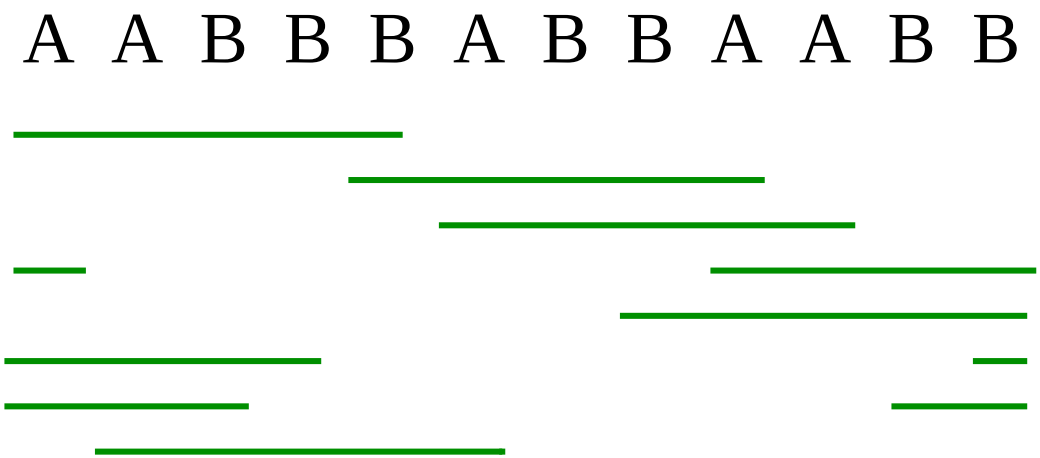
Graphe de de Bruijn : exemple 1



A A B 2
A B B 3
B B B
B B A 2
B A B
B A A

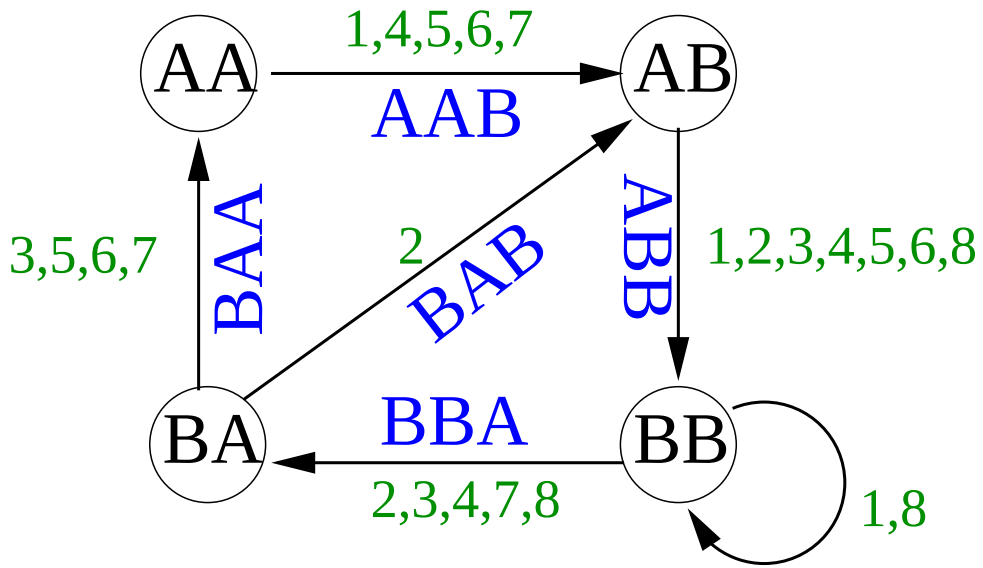


Graphe de de Bruijn : exemple 2



read 1
A A B
A B B
B B B

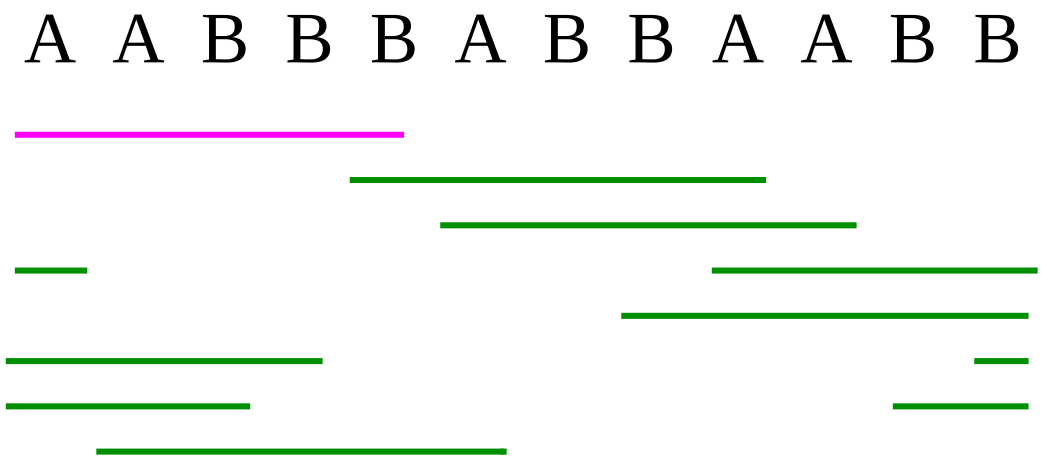
read 2
B A B
A B B
B B A

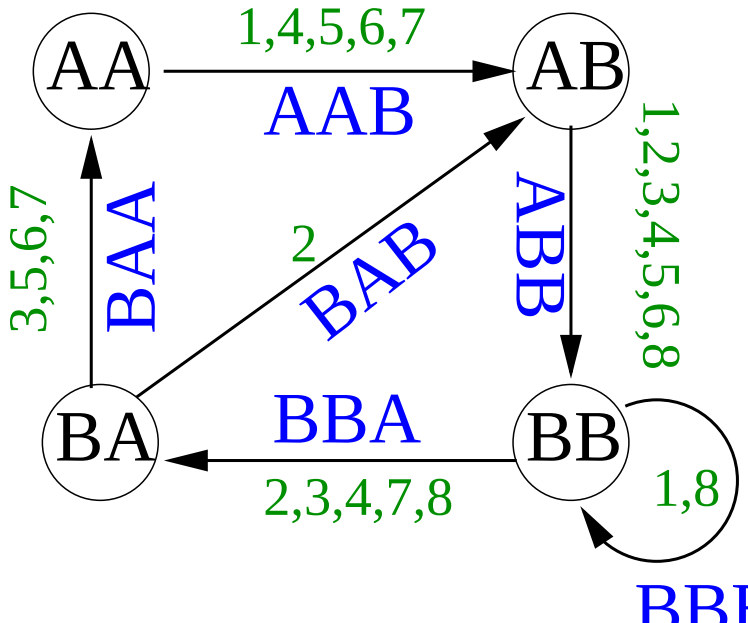


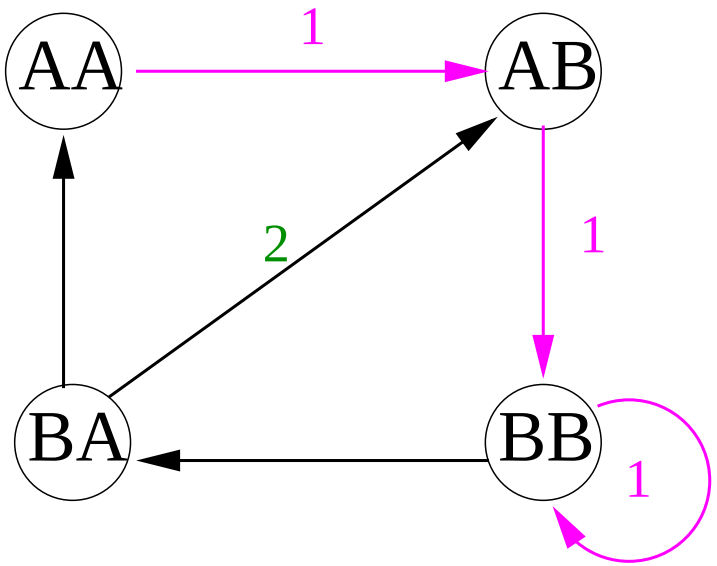
etc.

read 8
A B B
B B B
B B A

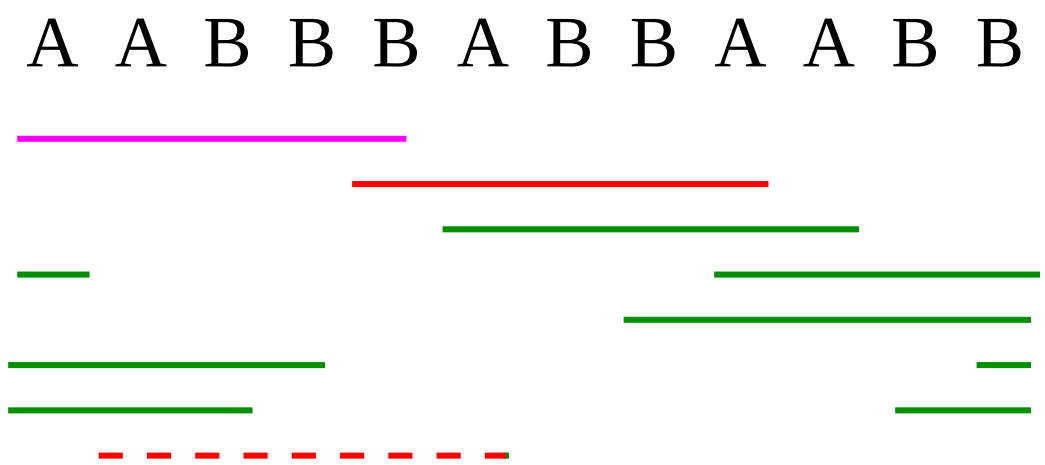
Graphe de de Bruijn : exemple 2

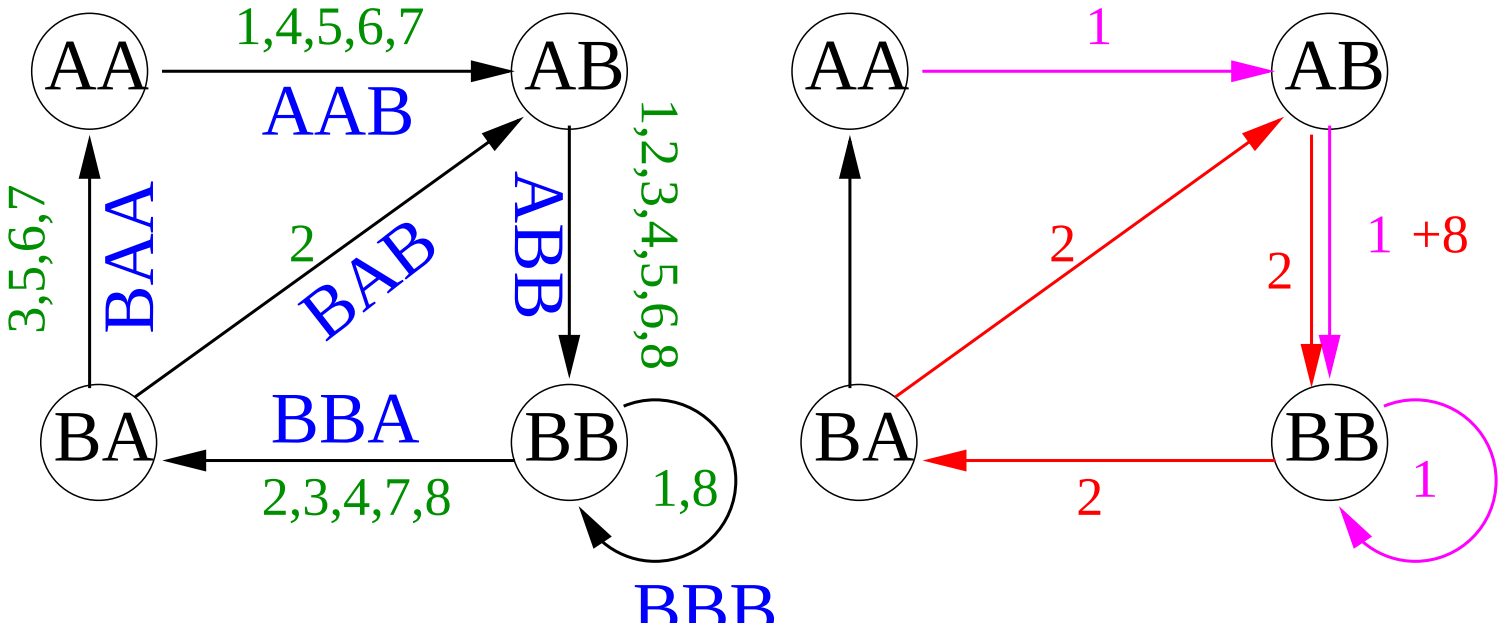




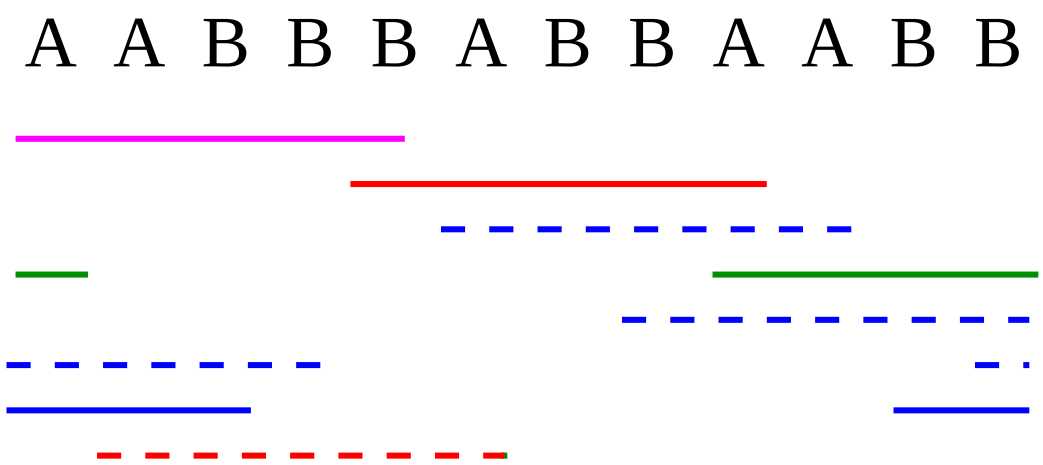


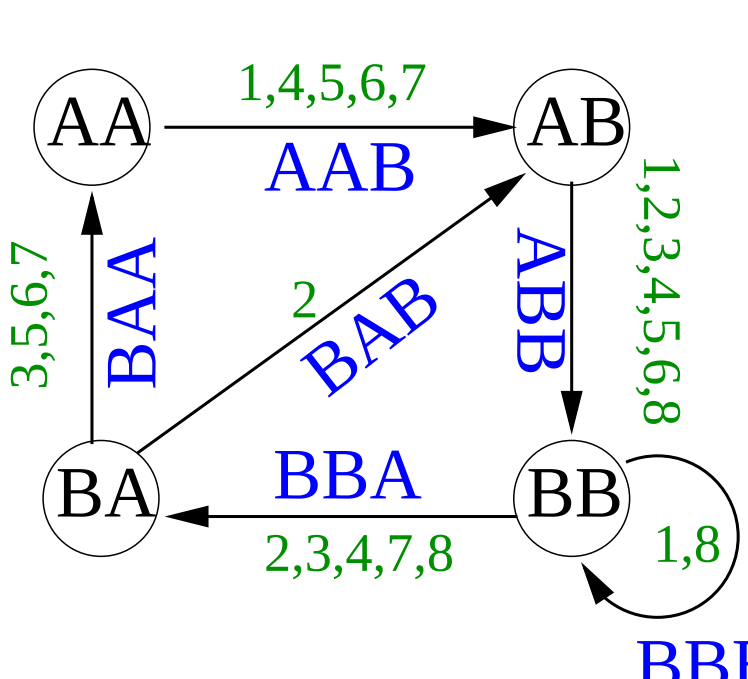
Graphe de de Bruijn : exemple 2

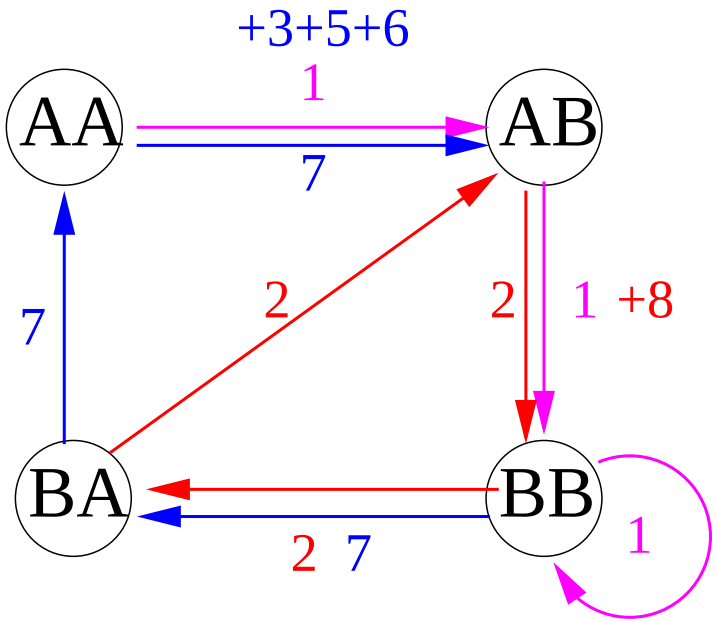




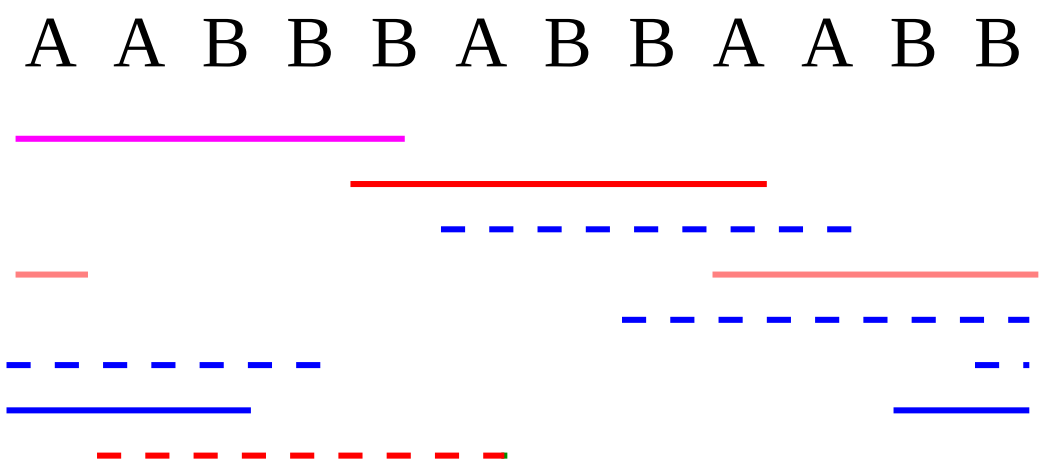
Graphe de de Bruijn : exemple 2

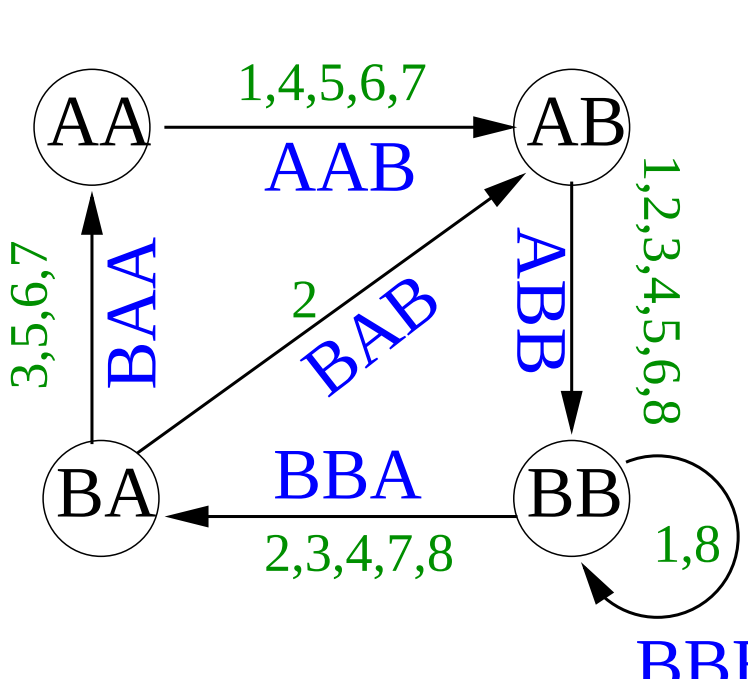


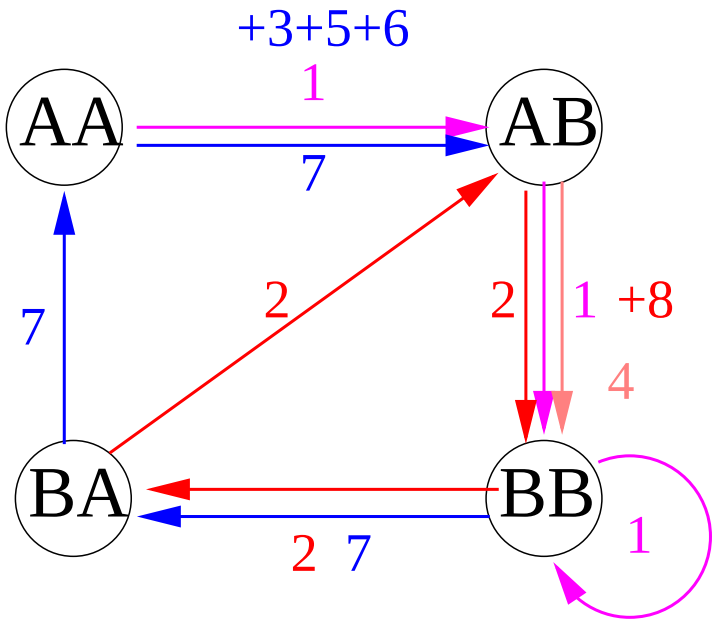




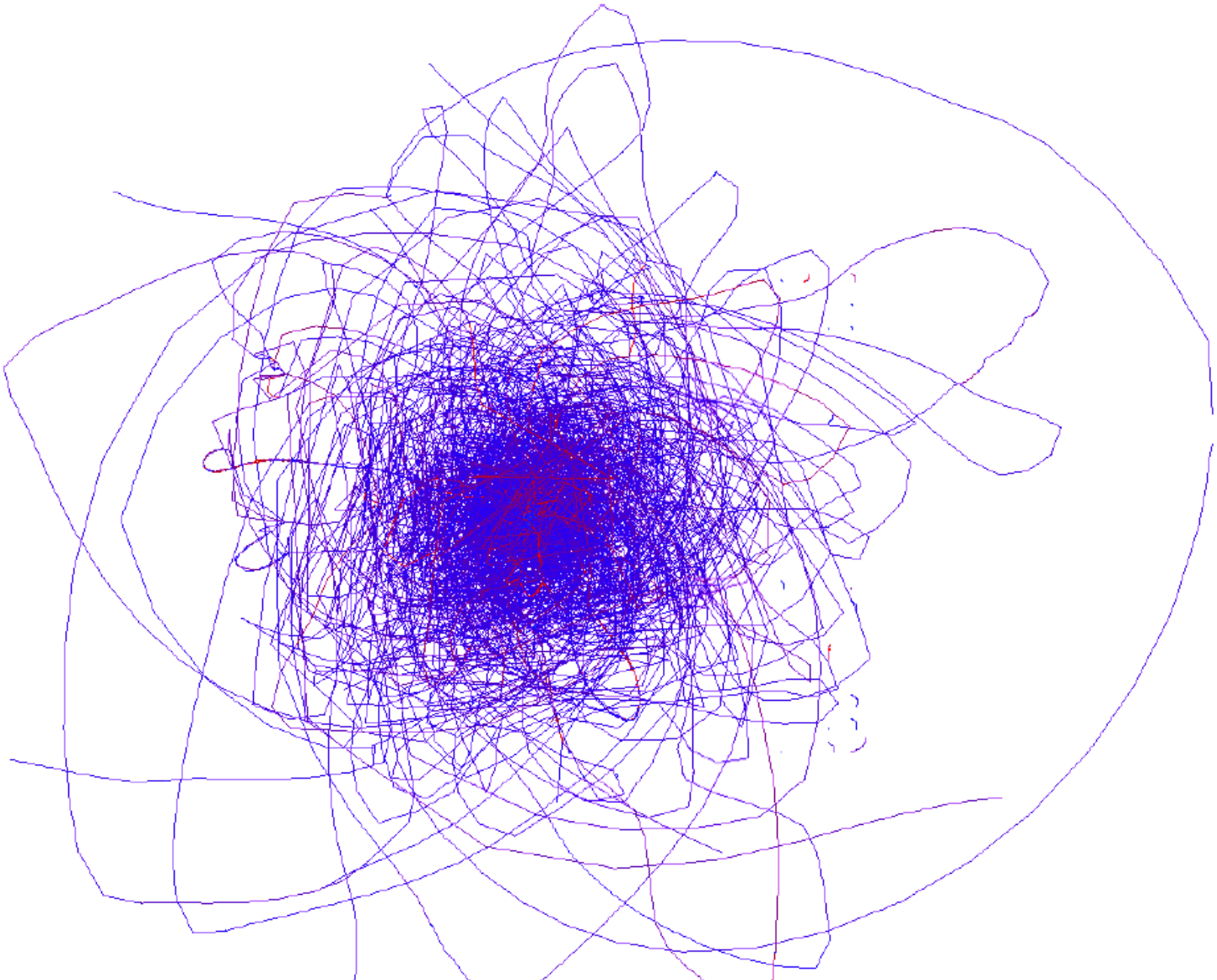
Graphe de de Bruijn : exemple 2





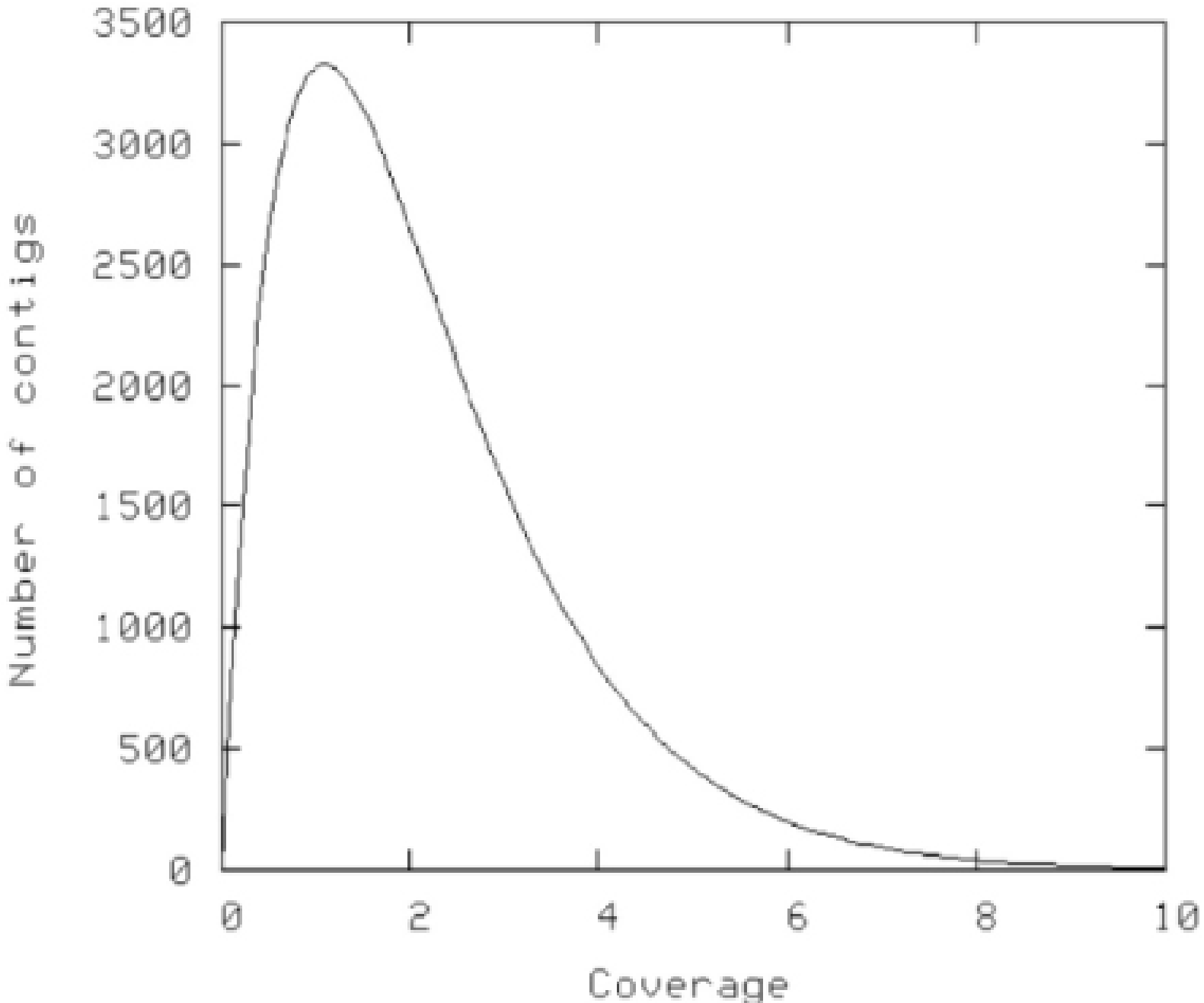


Graphe de de Bruijn : cas réel

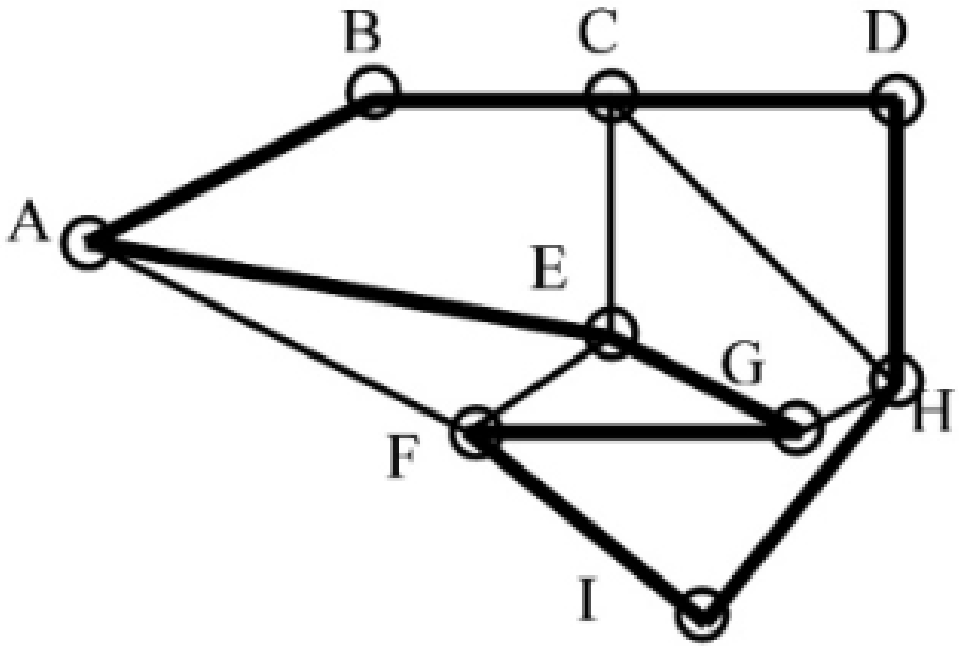


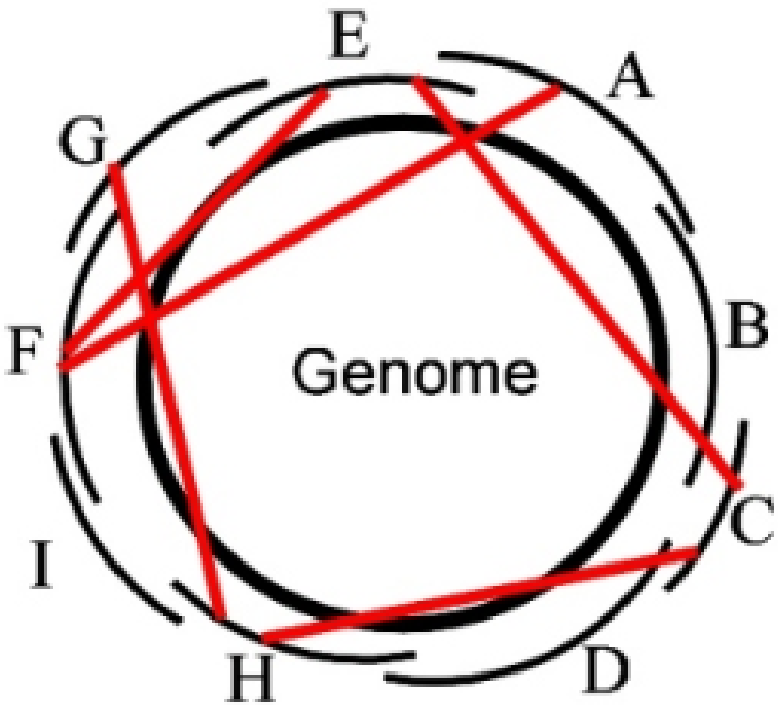
- Graph de de Bruijn de *Streptococcus suis*

Courbe de Waterman-Lander (génome 1Mbp)



Graphe hamiltonien





Algorithme glouton