# Algorithmique du texte GLIN608 - Séquençage et assemblage

Annie Chateau

# Introduction

**Principe du séquençage :** À partir de la molécule d'ADN, retrouver la séquence de paires de bases constituant le message génétique.

Les séquenceurs automatiques ont permis la naissance de la génomique

De nombreuses applications, en santé, en biologie, en paléontologie, et même dans le milieu médico-légal...

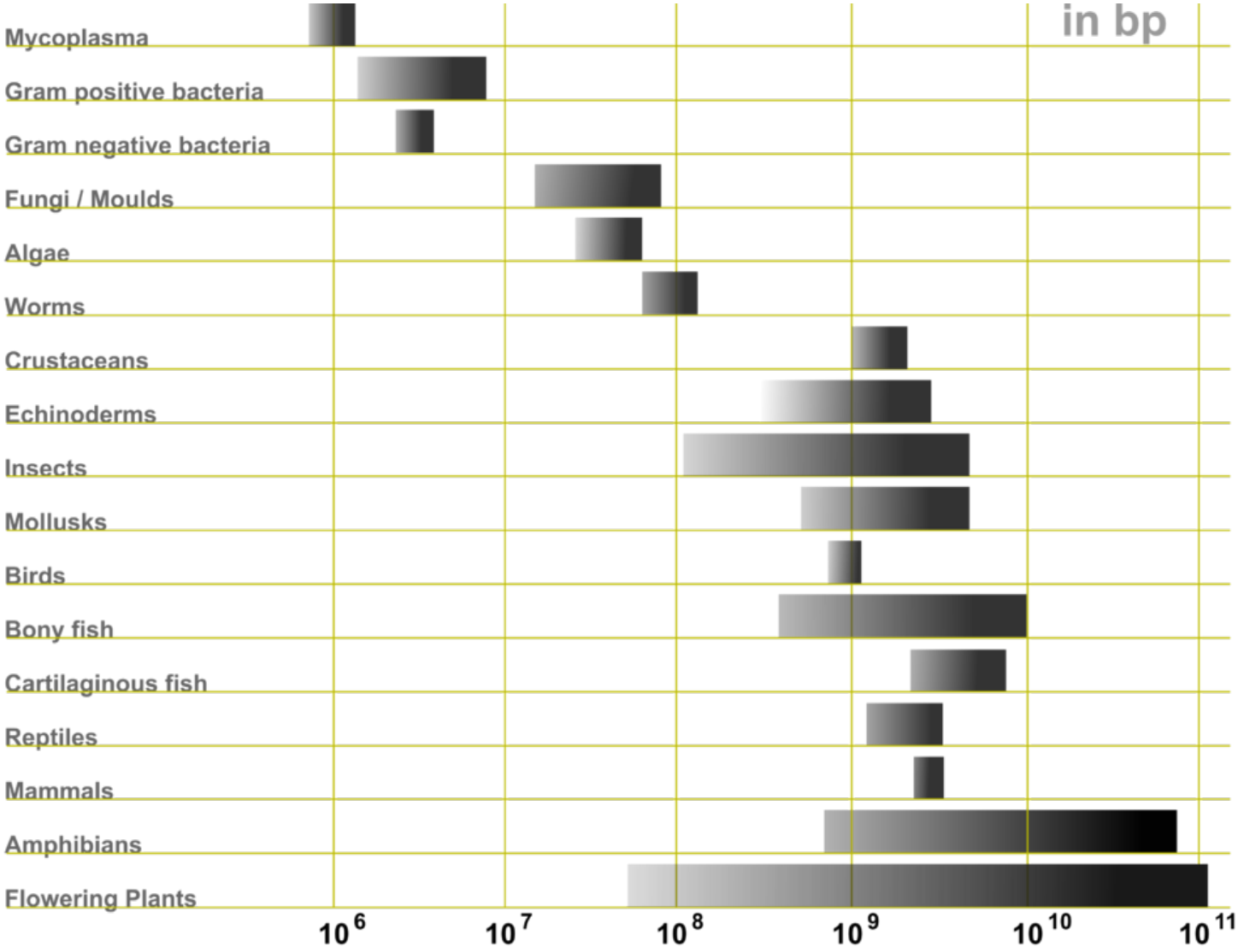
# Introduction

**Taille des génomes :** variable de quelques millions de paires de bases chez les bactéries à plusieurs milliards de paires de bases chez la plupart des animaux et des plantes.

Mais les séquenceurs automatiques ne peuvent proposer que des séquences de quelques centaines de paires de base à la fois...

**Problème :** On se retrouve avec un gigantesque PUZZLE à assembler.

# Introduction



# Les pionniers

- 1973 : Première publication d'une séquence de 24 bp

5'--TGGAATTGTGAGCGGATAACAATT--3'

3'--ACCTTAACACTCGCCTATTGTTAA--5'

# Les pionniers

- 1973 : Première publication d'une séquence de 24 bp

  5'--TGGAATTGTGAGCGGATAACAATT--3'

  3'--ACCTTAACACTCGCCTATTGTTAA--5'

- 1977-80 : Méthode de séquençage Sanger – Prix Nobel pour Wally Gilbert et Fred Sanger

# Les pionniers

- 1973 : Première publication d'une séquence de 24 bp

  5'--TGGAATTGTGAGCGGATAACAATT--3'

  3'--ACCTTAACACTCGCCTATTGTTAA--5'

- 1977-80 : Méthode de séquençage Sanger – Prix Nobel pour Wally Gilbert et Fred Sanger
- 1983 : Développement de la technique de PCR (Réaction en chaîne par polymérase)

# Les pionniers

- 1973 : Première publication d'une séquence de 24 bp

```
5'--TGGAATTGTGAGCGGATAACAATT--3'
3'--ACCTTAACACTCGCCTATTGTTAA--5'
```

- 1977-80 : Méthode de séquençage Sanger – Prix Nobel pour Wally Gilbert et Fred Sanger
- 1983 : Développement de la technique de PCR (Réaction en chaîne par polymérase)
- 1987 : 1er séquenceur automatique : Applied Biosystems Prism 373

# Les pionniers

- 1973 : Première publication d'une séquence de 24 bp

  5'--TGGAATTGTGAGCGGATAACAATT--3'

  3'--ACCTTAACACTCGCCTATTGTTAA--5'

- 1977-80 : Méthode de séquençage Sanger – Prix Nobel pour Wally Gilbert et Fred Sanger
- 1983 : Développement de la technique de PCR (Réaction en chaîne par polymérase)
- 1987 : 1er séquenceur automatique : Applied Biosystems Prism 373

- 1998 : Génome de *C. elegans* sequencé

# Les pionniers

- 1973 : Première publication d'une séquence de 24 bp

  5'--TGGAATTGTGAGCGGATAACAATT--3'

  3'--ACCTTAACACTCGCCTATTGTTAA--5'

- 1977-80 : Méthode de séquençage Sanger – Prix Nobel pour Wally Gilbert et Fred Sanger
- 1983 : Développement de la technique de PCR (Réaction en chaîne par polymérase)
- 1987 : 1er séquenceur automatique : Applied Biosystems Prism 373

- 1998 : Génome de *C. elegans* sequencé

- 2000 : Génome Humain séquencé

# Next Generation Sequencing (NGS)

- 2005 : 1er système de NGS : 454 (Roche) Life Sciences
- 2006 : 1er séquenceur de chez Solexa (Illumina)
- 2007 : 1er séquenceur de chez Applied Biosystems : SOLiD
- 2011 : 1er séquenceur de chez Ion Torrent : PGM

# Next Next Generation Sequencing (NNGS)

- 2009 : 1er séquenceur "monomolécule" : Helicos Genetic Analyser System
- 2011 : 1er séquenceur de longues lectures de chez Pacific Biosciences : PacBio RS Systems
- 2012 : Oxford Nanopore Technologies sort son séquenceur à ultralongue molécule

# Shotgun

On casse l'ADN en millions de fragments aléatoires qui sont "lus" par le séquenceur, après amplification (PCR = Polymérase Chain Reaction)

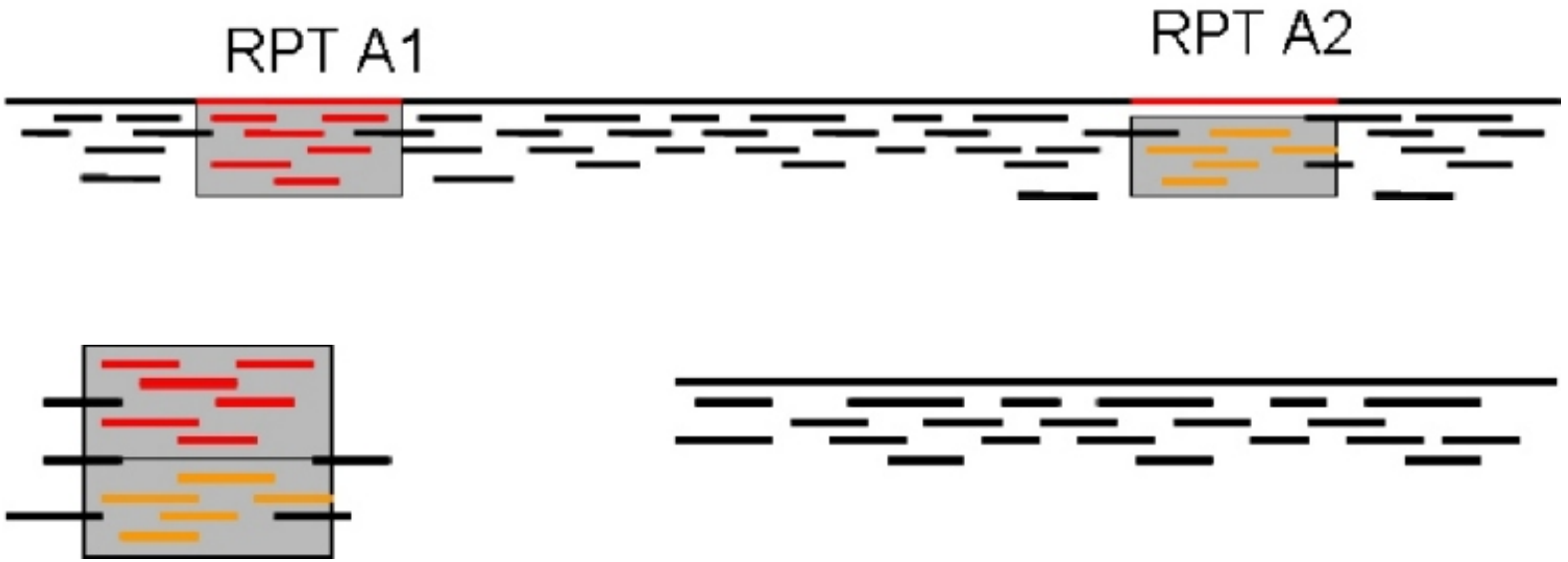
Ensuite, la séquence totale est reconstituée grâce aux chevauchements recherchés par des méthodes informatiques.

# Shotgun : limitations

- Erreurs de lecture (recours à des méthodes inexactes, mais moins de précision sur le génome final)
- Impossibilité de séquencer correctement certaines zones du génome
- Problèmes dus à la présence de répétitions $\Rightarrow$ fausse la recherche de chevauchements optimaux et produit des erreurs à l'assemblage.

Une solution : Masquer les répétitions, mais coûteux en ressources...

# Shotgun : limitations



# Applications

- Production de génome

# Applications

- Production de génome
- Profil d'expression des transcrits

# Applications

- Production de génome
- Profil d'expression des transcrits
- Relation entre les facteurs de transcription

# Applications

- Production de génome
- Profil d'expression des transcrits
- Relation entre les facteurs de transcription
- Variations structurelles

# Applications

- Production de génome
- Profil d'expression des transcrits
- Relation entre les facteurs de transcription
- Variations structurelles
- Métagenomique
- ...

# Types de données

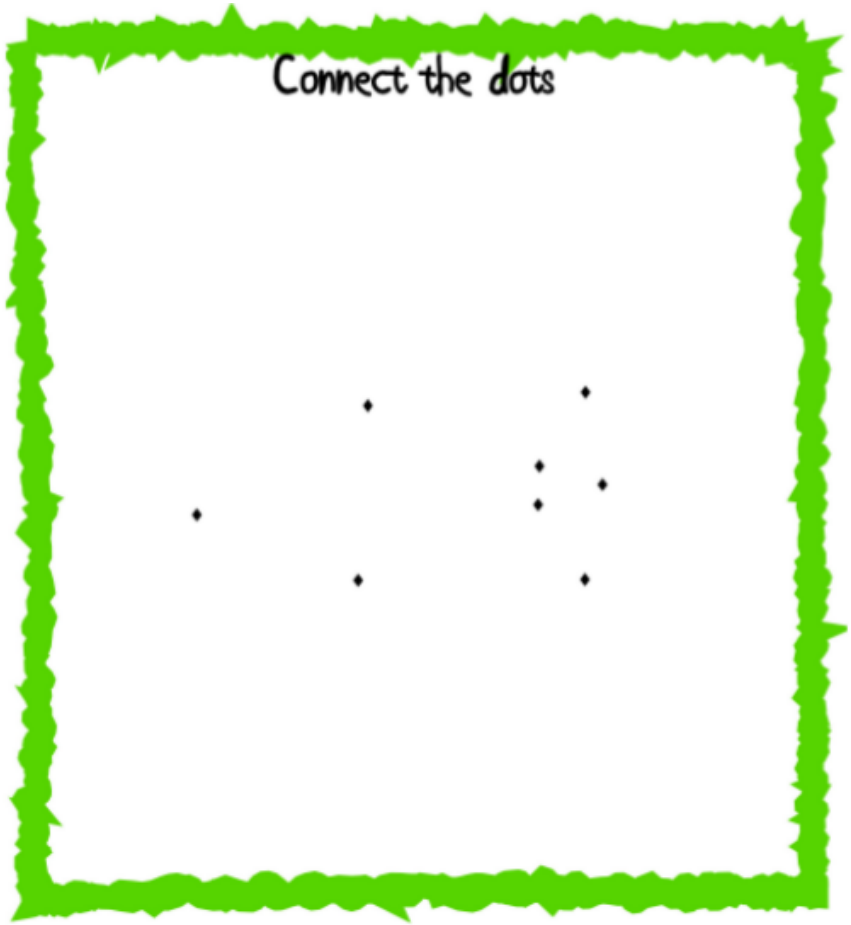
| | Capacity | Speed | Read Length | Homopolymers | Cost/run | Amplification |
|---|---|---|---|---|---|---|
| **454 Roche** | 35-700 Mb | 10-23 hours | 400-700 bp | - | 5.000 € | Yes |
| **SOLiD** | 90-180 Gb | 7-12 days | 75 bp | + | 5.000 € | Yes |
| **Illumina** | 6-600 Gb | 2-14 days | 100-250 bp | + | 10.000-20.000 € | Yes |
| **Ion Torrent** | 20 Mb- 1Gb | 4,5 hours | 200 bp | - | 1.000-2.000 € | Yes |
| **Helicos** | 35 Gb | 8 days | 35 bp | + | 20.000 € | No |
| **PacBio** | 1Gb | 30 minutes | 3000 bp | + | 600-800 € | No |

d'après Andy Vierstraete

# Le problème de l'assemblage

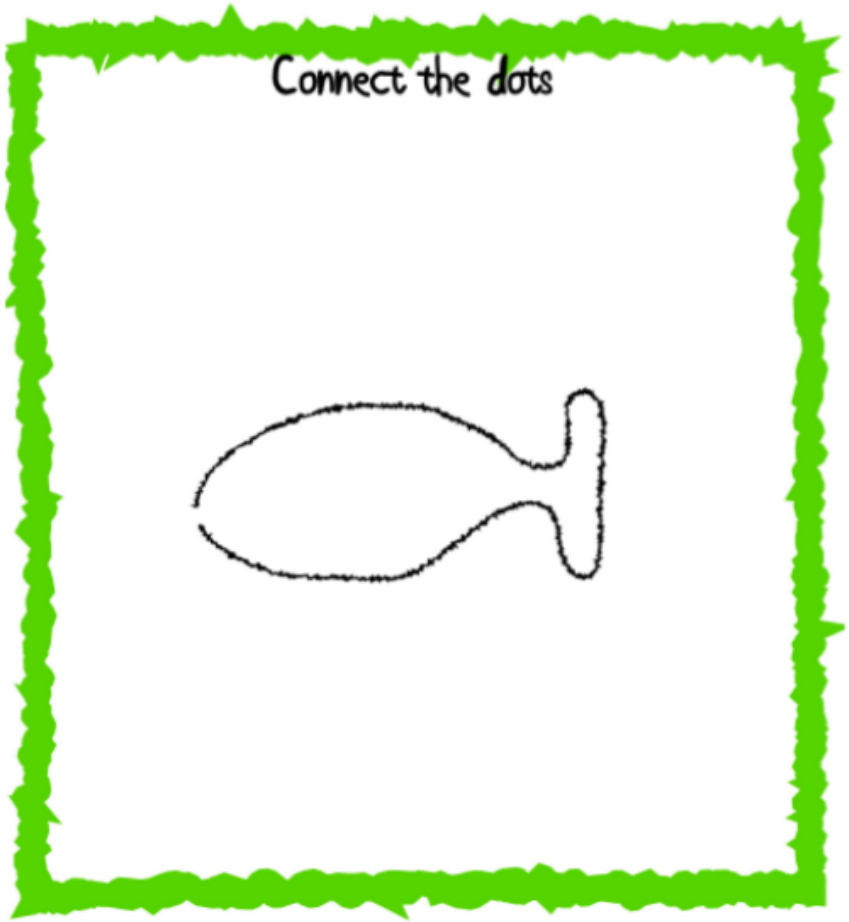
- Puzzle gigantesque avec des millions de petites pièces (short reads) de bonne qualité ou des milliers de grosses pièces (long reads) de qualité moins bonne.
- Parfois on a un modèle approximatif (le génome de référence)
- Parfois non (assemblage *de novo*)

# Le problème de l'assemblage

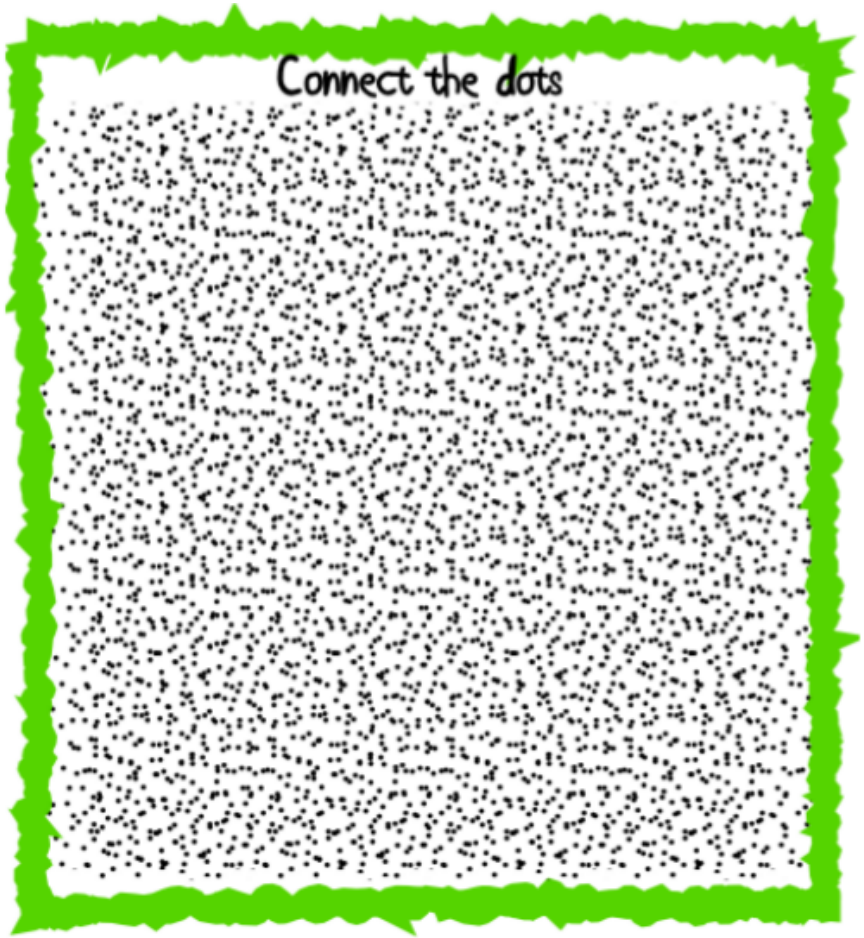



d'après Andy Vierstraete

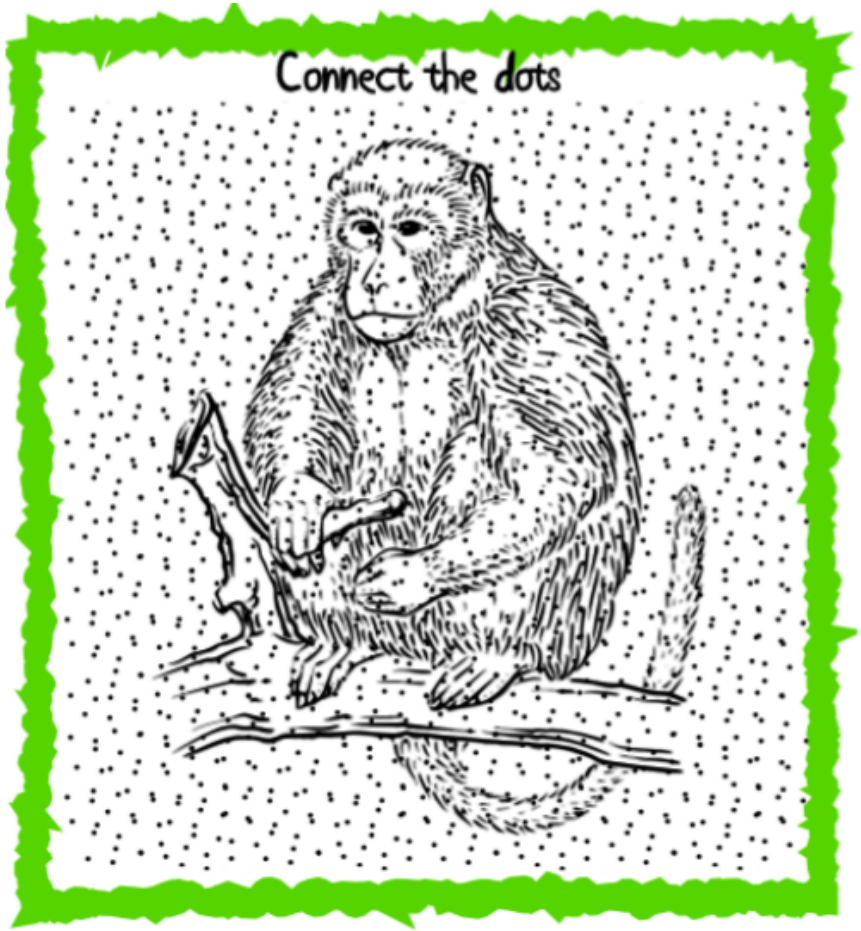
# Le problème de l'assemblage



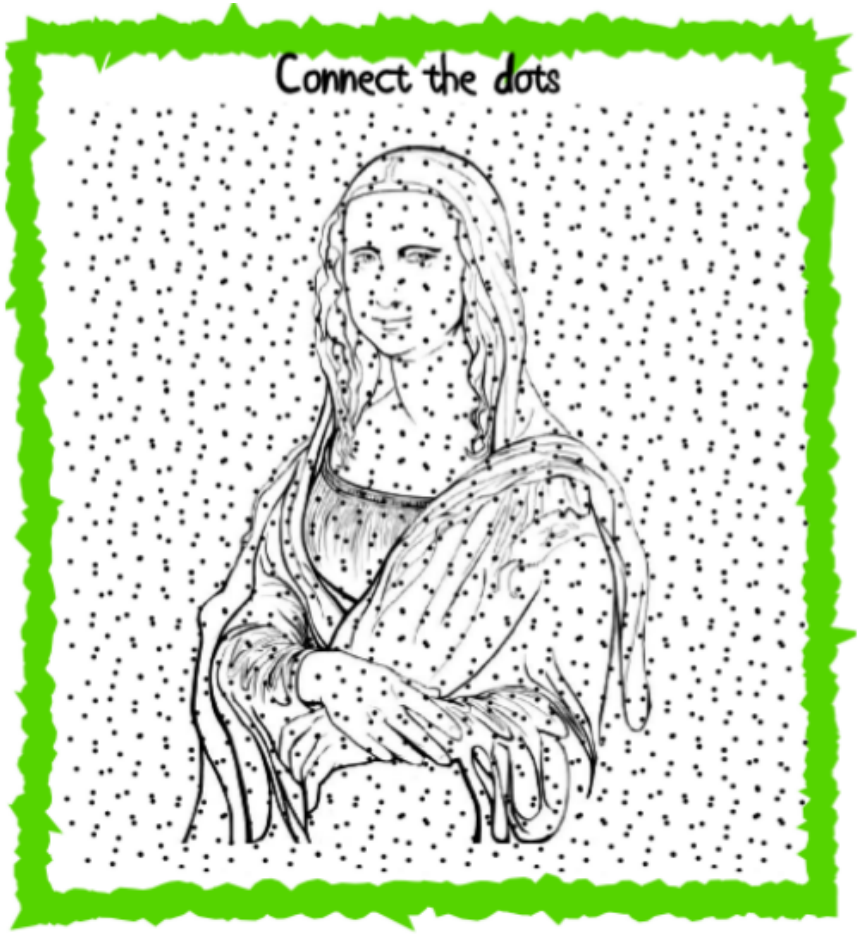
# Le problème de l'assemblage



# Le problème de l'assemblage



# Le problème de l'assemblage



# Un brin de théorie

On note $P$ la profondeur de couverture d'un génome.

$$P = \frac{\sum \text{longueurs des fragments}}{\text{Taille du génome}}$$

**Exemple :** 25Mb séquencés pour un génome de 5Mb donne une profondeur 5, notée **5X**.

# Un brin de théorie

Plus la profondeur est grande, plus on a de reads chevauchants, plus le génome est couvert.

Mais il arrive un moment où il est plus économique de boucher les trous restants de façon ciblée (finition)

Comment déterminer la profondeur optimale ?

# Un brin de théorie

On note :

- $n$ la longueur moyenne des reads
- $L$ la longueur de la séquence cible
- $K$ le nombre de reads

On suppose que $n$ est très petit devant $L$, et que $K = \frac{PL}{n}$ est très grand.

# Un brin de théorie

Alors la théorie de Lander et Waterman indique que l'on peut modéliser l'apparition d'une base dans un nombre donné de reads par une loi de Poisson.

La probabilité d'une base d'apparaître dans $x$ reads est alors $\frac{P^x e^{-P}}{x!}$

En particulier, la probabilité d'une base de n'apparaître dans aucun read est $e^{-P}$.

# Un brin de théorie

Proportion du génome couverte $= 1 - e^{-P}$

Longueur totale des trous $= Le^{-P}$

Nombre total de trous $= Ke^{-P}$

Taille moyenne des trous $= \frac{n}{P}$

Taille moyenne des contigs $= \frac{n}{P}e^{P}$

# Un brin de théorie

**Exemples de couvertures pour le séquençage**

On considère le génome humain ($L = 3.10^9$, $n = 1000$)

- $N = \sum n = 9.10^9 \Rightarrow$ **3X** et $K = 9.10^6$
  % d'ADN avec au moins une lecture : $1 - e^{-3} = 95\%$
  Nombre de trous : $9.10^6 \times 0.05 = 450000$ trous
  Longueur moyenne des trous : $1000/3 = 333$ nucléotides
- $N = \sum n = 3.10^10 \Rightarrow$ **10X** et $K = 3.10^7$
  % d'ADN avec au moins une lecture : $1 - e^{-10} > 99.99\%$
  Nombre de trous : $3.10^7 \times 5.10^{-5} = 1500$ trous
  Longueur moyenne des trous : $1000/10 = 100$ nucléotides

# Le problème SSP

Pour étudier le problème de l'assemblage de façon algorithmique : on le formalise comme un problème combinatoire sur des *mots*.

Formellement, l'assemblage se résume au problème suivant : étant donné un ensemble de mots $F = \{F_1, \ldots, F_n\}$ (les fragments), trouver une superséquence $S$ de longueur minimale tel que chaque mot de $F$ est un sous-mot de la superséquence $S$.

C'est le problème SSP (*Shortest Superstring Problem*)

# Le problème SSP

**Exemple**

```
F1 : AATGCC
F2 :    GCCTTACAC
F3 :          ACACTG
F4 :            ACTGAAGG
F5 :                GAAGGTTTA
---------------------------
B  :  AATGCCTTACACTGAAGGTTTA
```

# Le problème SSP

**Problème !**

SSP est un problème NP-complet pour plus de trois mots et un alphabet de plus de deux lettres...

On va devoir donc étudier des heuristiques pour résoudre le problème de façon polynomiale...

# Le problème SSP

Preuve de NP-complétude : En TD...

# Méthode gloutonne

L'heuristique la plus naturelle : la méthode gloutonne.

1. Calculer les scores de chevauchement pour chaque paire de fragment (alignement semi-global)
2. Sélectionner une paire avec le meilleur chevauchement, fusionner les deux fragments, et recommencer

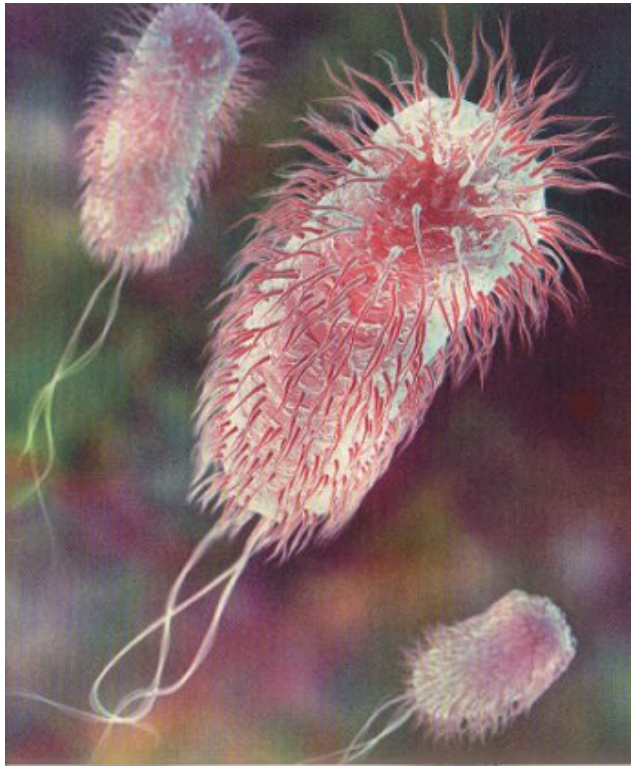
Complexité ?

# Méthode gloutonne

**Exercice :** Appliquer cette méthode à la famille de mots suivantes et retrouver la séquence initiale :

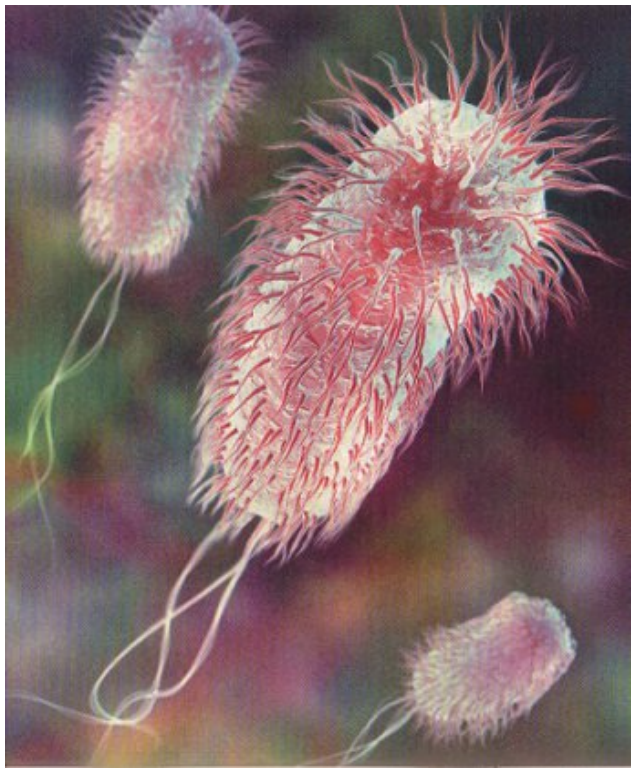
$F_1 =$ ACCTGA
$F_2 =$ TGATTGC
$F_3 =$ GCAGC
$F_4 =$ AGCAA
$F_5 =$ CAATG

# Méthode gloutonne



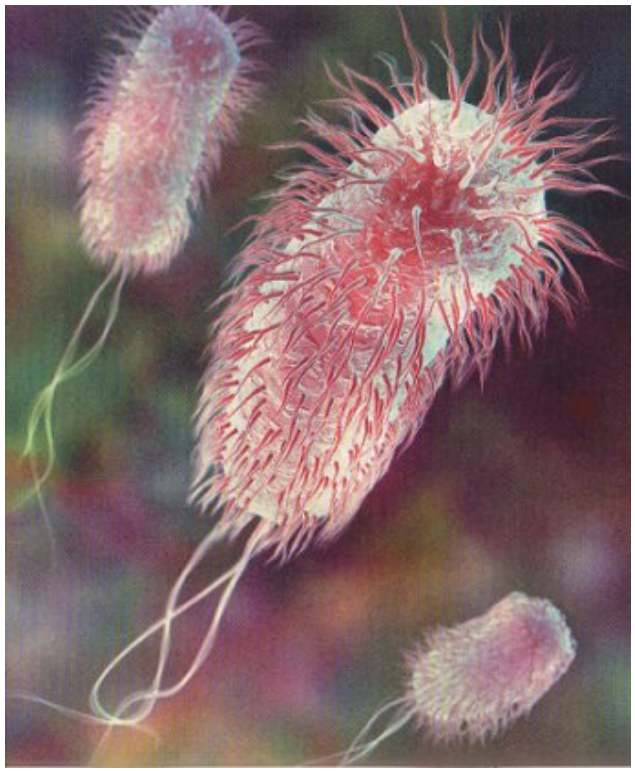
# Méthode gloutonne

- Facile à implémenter



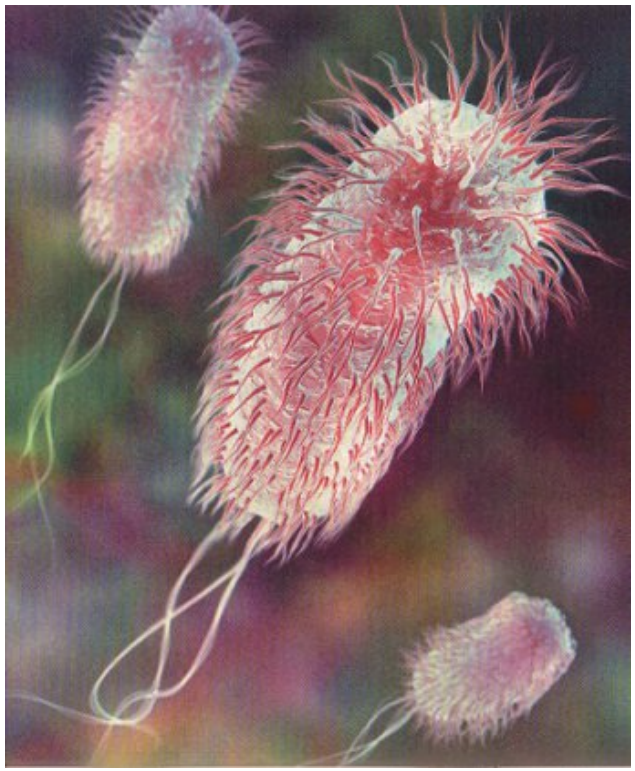
# Méthode gloutonne

- Facile à implémenter
- Résultat assez satisfaisant mais...



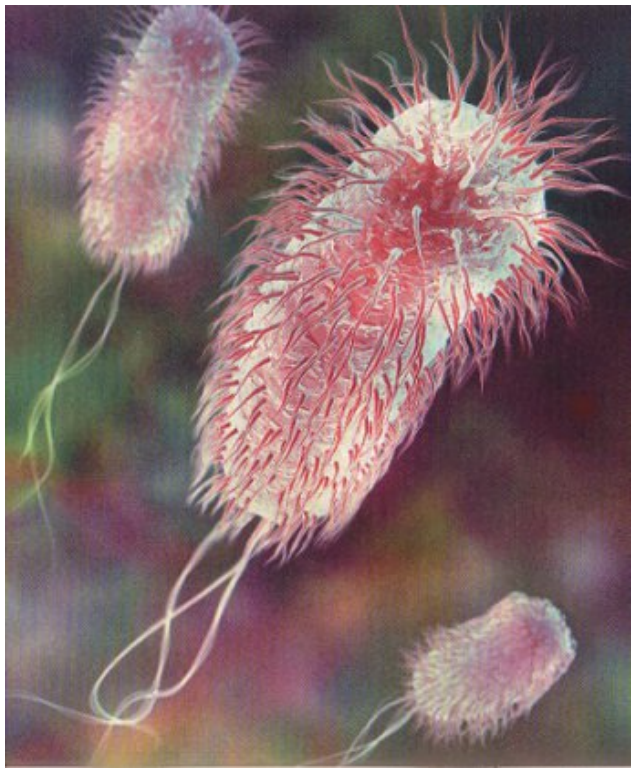
# Méthode gloutonne

- Facile à implémenter
- Résultat assez satisfaisant mais...
- Prend énormément de mémoire : applicable seulement à des petits génomes



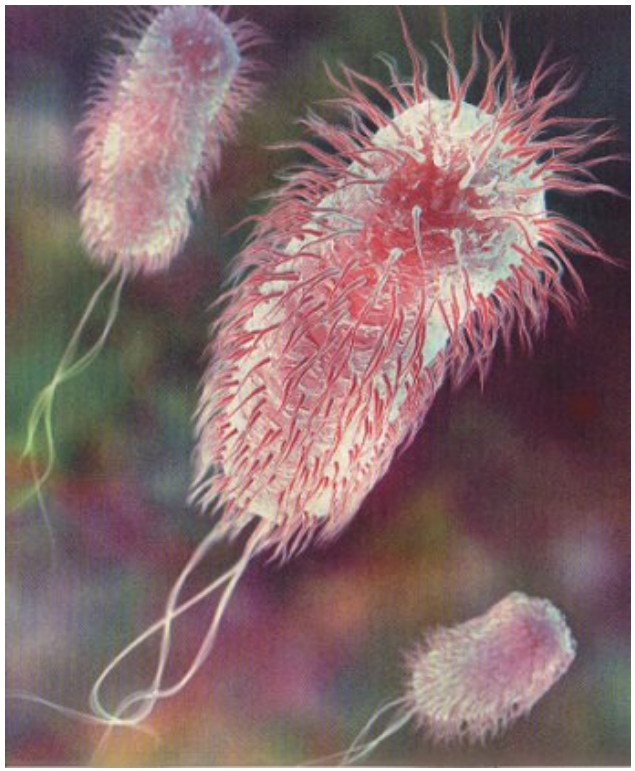
# Méthode gloutonne

- Facile à implémenter
- Résultat assez satisfaisant mais...
- Prend énormément de mémoire : applicable seulement à des petits génomes
- Critère local pas adapté pour tenir compte des répétitions



# Méthode gloutonne

- Facile à implémenter
- Résultat assez satisfaisant mais...
- Prend énormément de mémoire : applicable seulement à des petits génomes
- Critère local pas adapté pour tenir compte des répétitions
- $\Rightarrow$ approches basées sur des graphes



# Overlap-Layout-Consensus

Dans cette méthode, on se sert du fait que le calcul des chevauchements optimaux permet de construire un graphe sur lequel on peut travailler.

**Overlap :** trouver chaque paire de lectures qui se chevauchent ;

**Layout :** organiser ces lectures chevauchantes en une séquence contiguë (chemin hamiltonien) ;

**Consensus :** corriger les erreurs et générer une séquence consensus.

# Overlap-Layout-Consensus

Definition
On définit $overlap(s_i, s_j)$ la longueur du plus long préfixe de $s_j$ qui correspond à un suffixe de $s_i$.

# Overlap-Layout-Consensus

On construit un graphe avec $n$ sommets qui représentent les $n$ chaînes $s_1, s_2, \ldots, s_n$.

On insère les arcs portant un poids $overlap(s_i, s_j)$ entre les sommets $s_i$ et $s_j$.

**Problème :** Trouver le chemin de poids maximal qui visite chaque sommet exactement une fois.

C'est une variante du problème du voyageur de commerce, qui est NP-complet !

# Overlap

## Phase Overlap

1. Calcul des chevauchements optimaux sur les reads : alignement semi-global
2. Construction progressive du graphe : on ajoute les sommets (reads) et les arêtes (chevauchements optimaux de score acceptable), pondérées par les scores.
3. On enlève du graphe (ou plutôt on ne met pas dans le graphe) les reads entièrement contenus

**Remarque :** Le graphe est orienté.

# Layout

## TSP

*Solution exactes*

Algorithme naïf : tester les $(n-1)!$ tours possibles.....

Programmation dynamique : Plus rapide mais complexité en espace $\mathcal{O}((n-1)2^{n-2})$...

Branch and bound

# Layout

## TSP

*Quelques heuristiques :*

- Approche gloutonne
- Le plus proche voisin
- 2-interchange
- $k$-opt
- Lin-Kernighan
- Programmation linéaire (Concorde) : jusqu'à 65000 sommets en quelques heures...

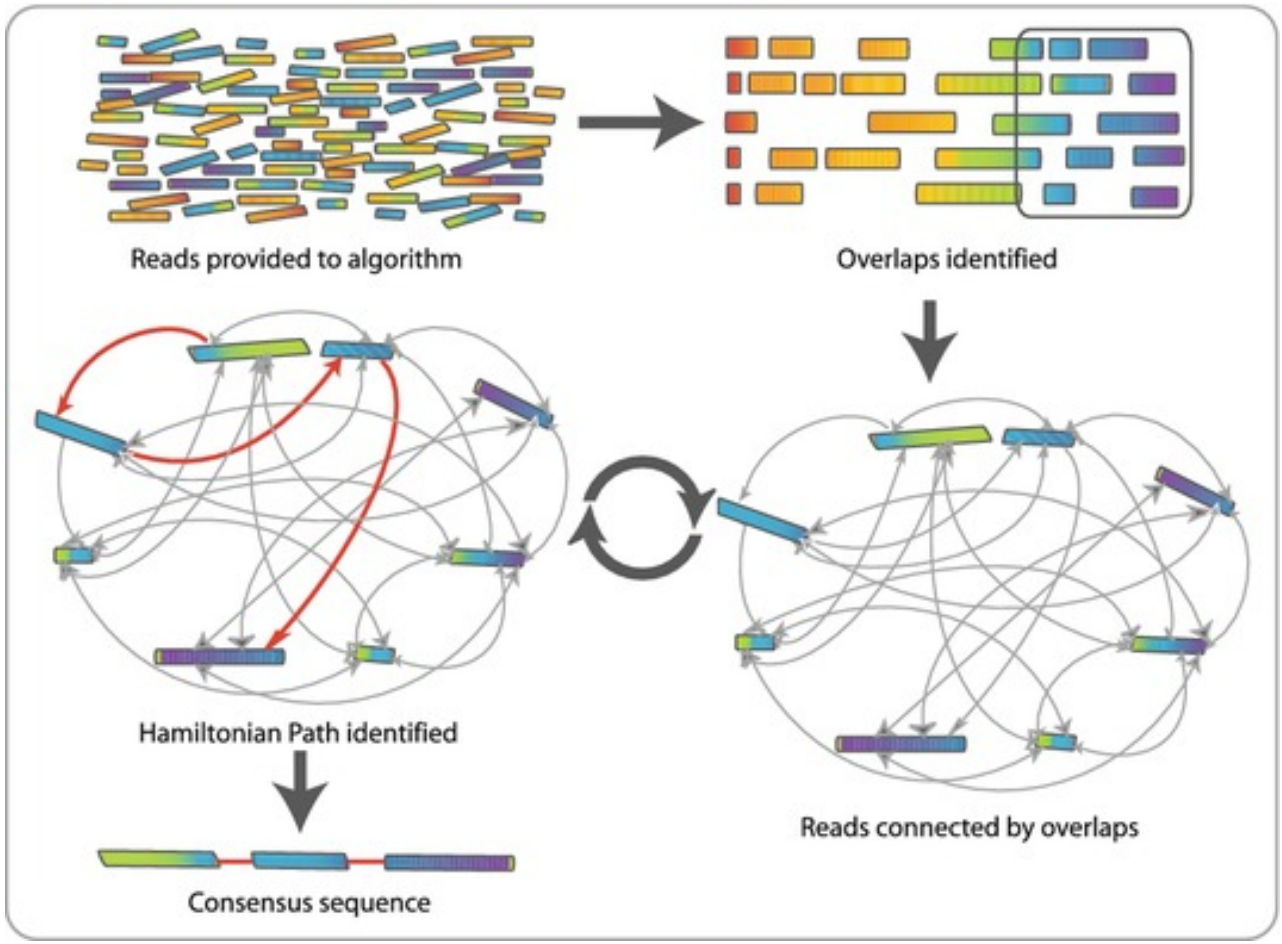
Assemblage : Cas asymétrique, on se ramène au cas symétrique en dupliquant les sommets.

# Consensus

## Phase Consensus

On réalise cette phase en réalisant un alignement multiple par profil.

# Overlap Layout Consensus




d'après J.Commins et al. *Computational Biology Methods and Their Application to the Comparative Genomics of Endocellular Symbiotic Bacteria of Insects* Bio.Proc.Online 2009

# Overlap Layout Consensus

- L'information complète sur chaque read en entrée n'est nécessaire que pour les phases overlap et consensus.

# Overlap Layout Consensus

- L'information complète sur chaque read en entrée n'est nécessaire que pour les phases overlap et consensus.
- La phase de raffinement du graphe permet de ne stocker qu'une quantité limitée d'information sur chaque chevauchement (coordonnées, longueur) $\Rightarrow$ implémentation efficace en mémoire.

# Overlap Layout Consensus

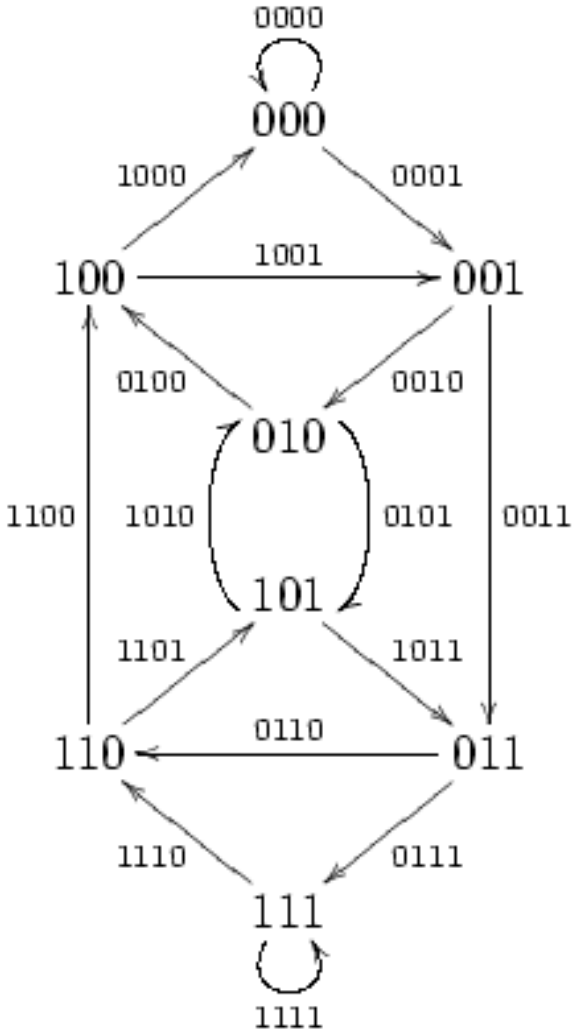
- L'information complète sur chaque read en entrée n'est nécessaire que pour les phases overlap et consensus.
- La phase de raffinement du graphe permet de ne stocker qu'une quantité limitée d'information sur chaque chevauchement (coordonnées, longueur) $\Rightarrow$ implémentation efficace en mémoire.
- Mais trouver un chemin Hamiltonien est tout de même un problème NP-complet...

# Chemin eulérien

On utilise dans cette approche un autre type de graphe lié aux reads : le graphe des $k$-mers, qui est un cas particulier de graphe de De Bruijn.

Definition
Soit un alphabet $\mathfrak{A} = \{\alpha_1, \ldots, \alpha_n\}$ de taille $n$. On note $G_{n,L} = (S, A)$ le graphe de de Bruijn d'ordre $L$ construit sur l'alphabet $\mathfrak{A}$. Les sommets $S$ de $G_{n,L}$ représentent les $L^n$ mots de longueur $L$ construits à partir des lettres de l'alphabet $\mathfrak{A}$. Il existe un arc $A_{i,j}$ allant du sommet $s_i$ au sommet $s_j$ du graphe $G_{n,L}$ si le suffixe de longueur $L-1$ du mot $u$ et le préfixe de longueur $L-1$ du mot $v$ correspondent.

# Chemin eulérien



# Chemin eulérien

Théorème
*Les graphes de De Bruijn sont eulériens.*

**Rappel :** un graphe est eulérien si on peut parcourir le graphe en utilisant exactement une fois chaque arc et en revenant au point de départ.

Théorème (Théorème d'Euler (version orientée))
*Un graphe orienté fortement connexe est eulérien si et seulement pour chaque sommet le degré entrant est égal au degré sortant.*

# Chemin eulérien

- Décomposition des reads en $k$-mers :

```
ATTGCTAACTGT                    Séquence d'origine (read)
ATTGCT
 TTGCTA
  TGCTAA
   GCTAAC                                  6-mers
    CTAACT
     TAACTG
      AACTGT
```

# Chemin eulérien

1. On choisit un $k$ (impair) suffisamment petit pour capturer les chevauchements et suffisament grand pour éviter les répétitions...
2. Nettoyage des reads (méthode statistique de détection des erreurs de séquençage)
3. Construction du graphe (et stockage des $k$-mers dans une structure de données compressée)
4. Nettoyage du graphe (détection des erreurs par analyse de la structure)
5. Recherche de chemins

# Chemin eulérien

- Formalisation simple de la structure de graphe de De Bruijn

# Chemin eulérien

- Formalisation simple de la structure de graphe de De Bruijn


- Le problème du chemin eulérien est polynomial.

# Chemin eulérien

- Formalisation simple de la structure de graphe de De Bruijn

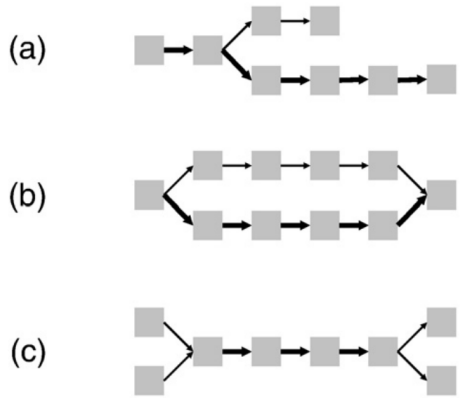

- Le problème du chemin eulérien est polynomial.

- Mais la taille de l'entrée peut tout de même être énorme, dépendamment de $k$ et des reads $\Rightarrow$ efforts à faire sur le côté "stockage" (algo du texte plus avancée...)

# Nettoyer l'assemblage

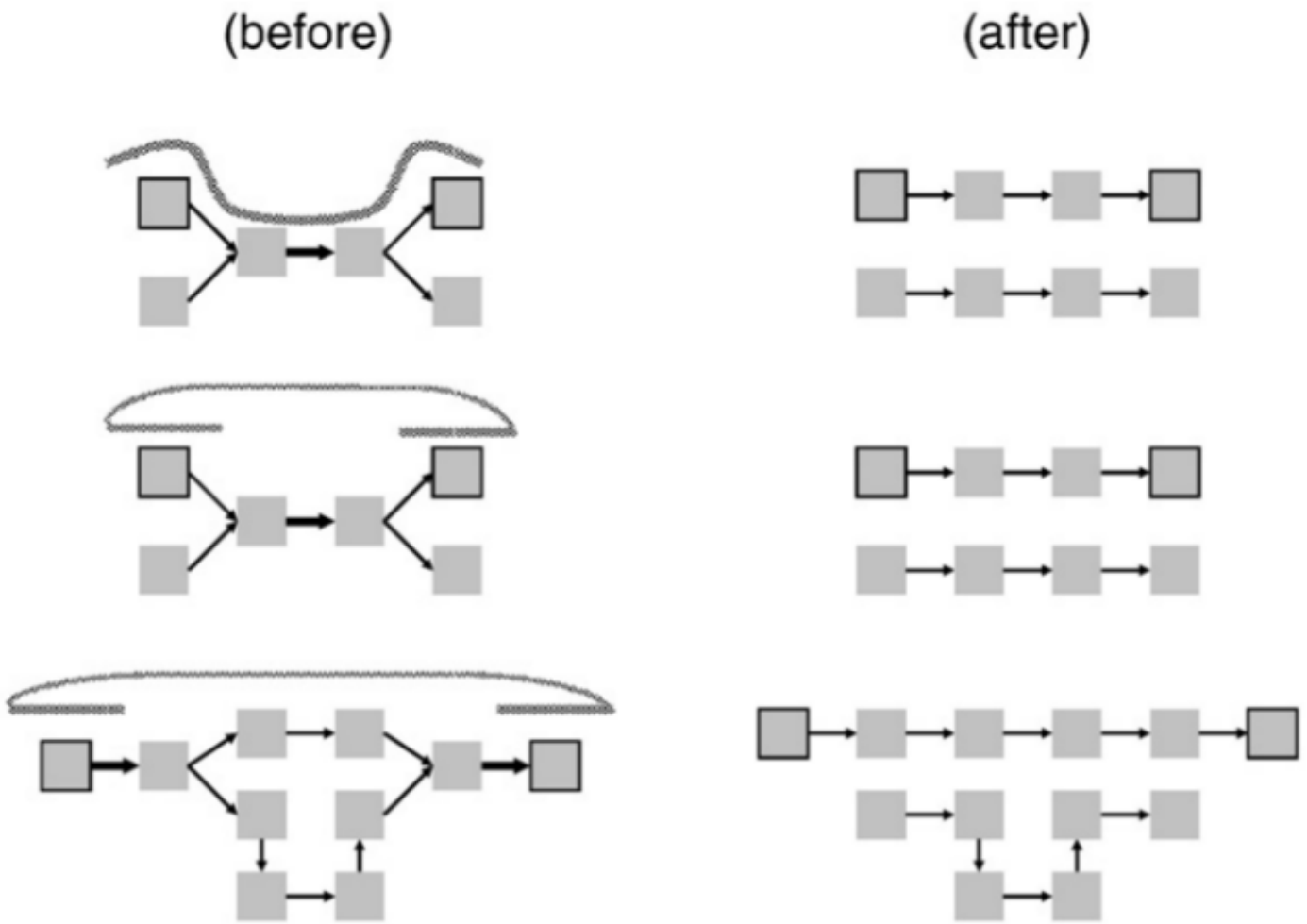
La plupart des assembleurs ont des post-traitements pour réaliser les tâches suivantes :

- Corriger les erreurs de séquençage (a)
- Détecter les polymorphismes (b)
- Gérer les répétitions (c)
- Détecter les régions mal assemblées




J.R.Miller et al., *Assembly algorithms for next-generation sequencing data* , Genomics, 2010

# Nettoyer l'assemblage



# Échafaudage (Scaffolding)

Après l'assemblage, on dispose de contigs, généralement de grande taille. Il reste à les placer sur le génome.

On choisit les tailles de fragments et des reads de sorte à les apparier (*paired-ends* ou *mate pairs*), permettant ainsi de déterminer l'orientation et l'ordre des contigs. C'est l'opération d'échafaudage ou scaffolding.

C'est aussi un problème NP-complet...

# Finition

Il reste toujours des trous, des incohérences, des ambiguïtés à lever.

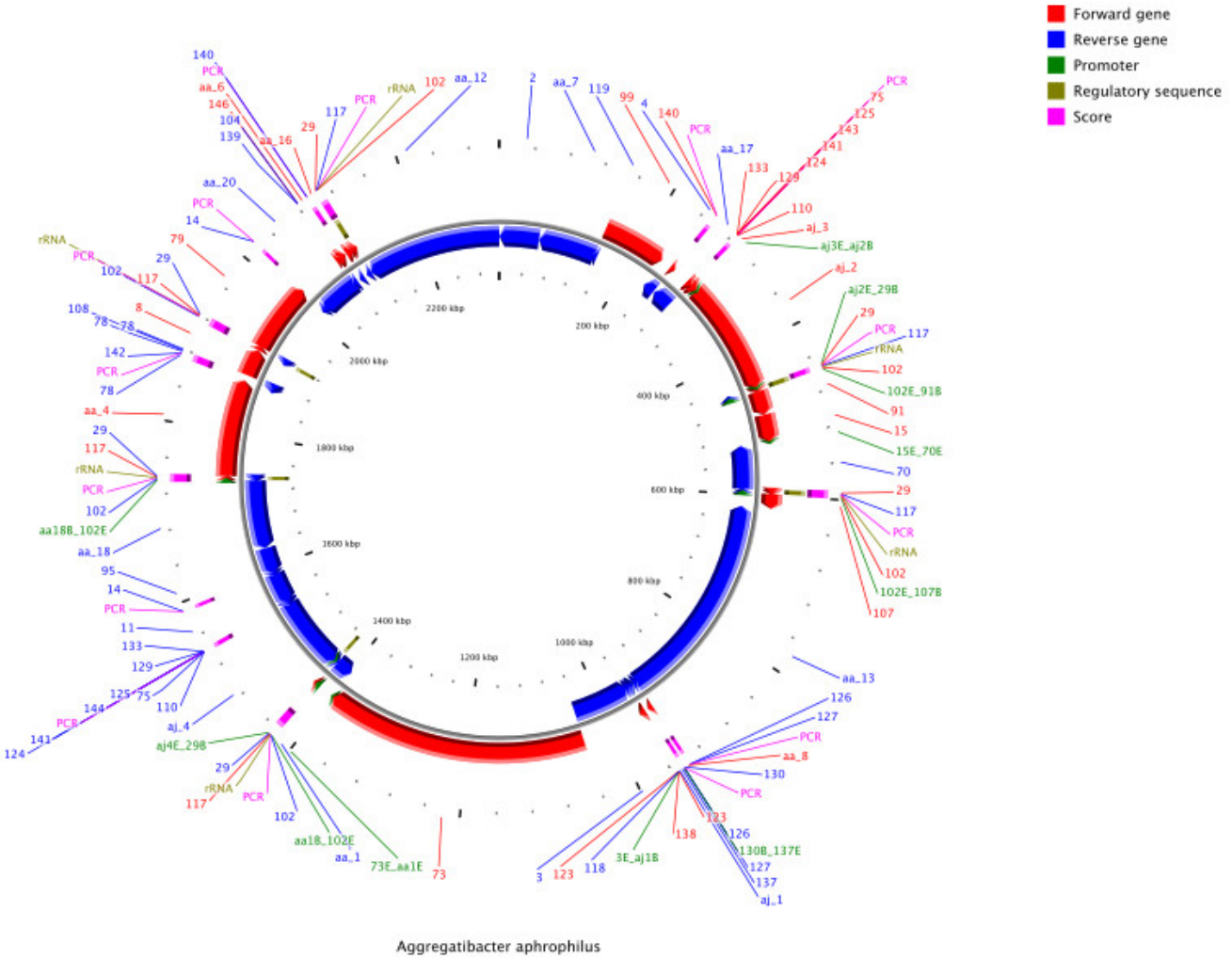
On réalise des expériences supplémentaires en laboratoire, on nettoie "à la main", afin de valider la correction de l'assemblage final.

C'est une opération très coûteuse, qui peut prendre des mois...

$\Rightarrow$ le temps et l'effort ne sont justifiés que pour les génomes "à haute priorité"

...les autres restent à l'état de brouillons (*draft genomes*)

# Finition




Nagarajan et al. BMC Genomics 2010