SÉQUENÇAGE DE GÉNOMES
ASSEMBLAGE DE FRAGMENTS D'ADN

Olivier.Delgrange@umh.ac.be

Service d'Informatique G´en´erale

Comment sont séquencés les longs génomes ?

Séquencer = Déterminer la séquence de bases d'une molécule d'ADN

Cas concret : Le **génome humain**

- 2.91 Gpb
- 23 paires de chromosomes

Contraintes pratiques :

- Plusieurs individus pour mettre en évidence les SNPs (*Single Nucleotide Polymorphism* : positions pour lesquelles 2 alternatives ou plus sont possibles à des fréquences appréciables [$> 1\%$] parmi la population)
- Erreurs de séquençage
- Longueur maximale pouvant être segmentée en une fois : $\approx$ 700pb
- Impossibilité d'extraire des fragments de 700pb en des positions et des orientations précises. Processus de découpage **aléatoire**.

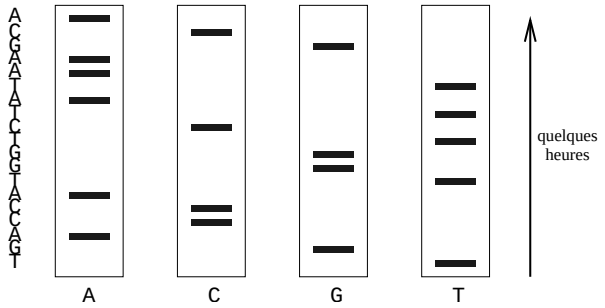
$\Longrightarrow$ Problème extrêmement complexe

Séquenç age d'un petit fragment (500→700pb)

Orientation $5' \rightarrow 3'$ mais brin souvent inconnu

Séquençage par *gel d'électrophorèse* :

Le fragment est cloné dans un vecteur et les copies sont réparties dans 4 gels spécifiques soumis à un champ électrique
Sous l'effet du champ électrique et selon le gel, les copies sont coupées en des nucléotides spécifiques (A, C, G ou T) et les morceaux se déplacent à une vitesse inversément proportionnelle à leurs longeurs.



Des détecteurs optiques déterminent la suite de nucléotides du fragment en recombinant les positions détectées dans les 4 gels.

La séquence est approximative (5% d'erreurs) car des endroits flous perturbent la détection automatique.

Comment organiser le séquenç age de très longues molécules ?

On dispose d'un processus bio-chimique aléatoire capable d'extraire des petits fragments des molécules d'ADN

- Longueur des fragments : ≈ 500pb
- Les positions sont inconnues
- L'orientation est inconnue

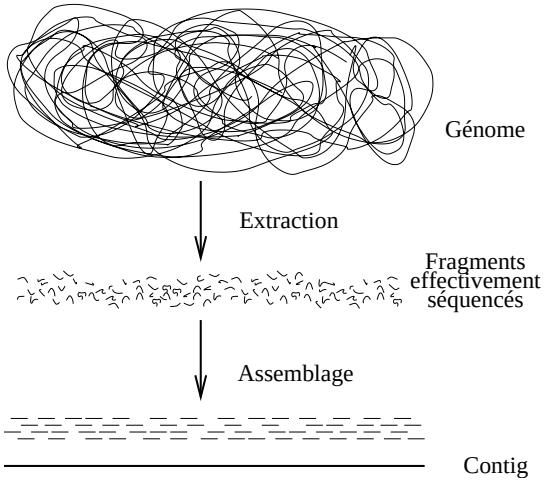
Chaque fragment peut-être séquencé par la méthode du gel d'électrophorèse.

La suite du travail repose sur des méthodes informatiques : utilisation des chevauchements approximatifs pour construire un *contig* ("super-séquence")

1. Shotgun complet du génome

L'ensemble de fragments est obtenu à partir de la **totalité du génome**.

La séquence complète du génome doit être reconstruite par assemblage des fragments.



Méthode adaptée aux génomes **peu répetitifs**.

Cependant utilisée par CELERA pour produire un "brouillon" du génome humain (février 2001) :
ADN prélevé sur 5 individus, 27 300 000 fragments de 543pb ($\Rightarrow$ 14 900 000 000pb).
Chevauchements d'au moins 40pb avec au plus 6% d'erreurs

2. Shotgun hiérarchique

Une librairie de longs morceaux est d'abord construite : le génome est découpé en des **positions précises** grâce aux enzymes de restriction.

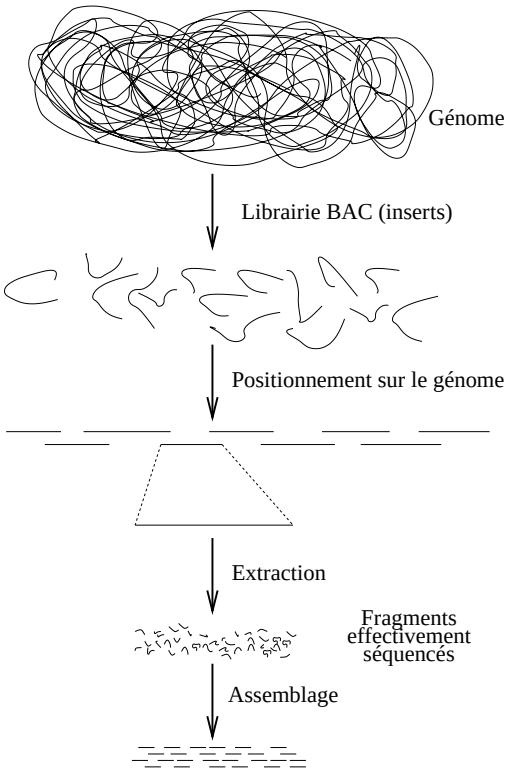
Les morceaux sont insérés dans des chromosomes bactériens artificiels (BACs) pour être dupliqués (*inserts*).

La séquence de chaque insert est déterminée selon le processus du shotgun.

Avantages :
- Moins de conséquence des répétitions sur l'assemblage lorsque les occurrences sont isolées dans des inserts différents (génome humain : 50% de répétitions, bactéries : 1.5% de répétitions et drosophile : 3% de répétitions).
- Les erreurs de séquençage et de clonage sont plus faciles à corriger car les informations sont locales.
- Le séquençage peut-être facilement distribué sur plusieurs laboratoires.

Le shotgun hiérarchique a été utilisé par CELERA parallèlement au shotgun complet pour valider leur résultat de brouillon du génome humain.
De nouveaux inserts ont été utilisés en combinaison avec des séquences humaines déjà publiées dans les banques de séquences par la recherche publique.

Le *Consortium international de séquençage du génome humain* a utilisé le shotgun hiérarchique (inserts de 100 000pb à 200 000pb) pour proposer son premier brouillon (février 2001)

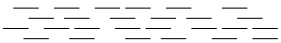


Remarques :

- un "brouillon" est une séquence ne présentant pas tous les critères de qualité des séquences finies.
- le brouillon n'est pas sensé couvrir tout l'ensemble du génome ; des "trous" séparent les régions connues.
- le brouillon est construit à partir d'un nombre restreint d'individus
- le brouillon propose une "couverture" restreinte dans la détermination des fragments effectivement séquencés.

Le problème d'assemblage de fragments

Étant donné une collection de fragments de quelques centaines de paires de bases, il faut construire la séquence cible

 fragments

 séquence cible

```
ACCGT              ⟹    ACCGT
CGTGC                      CGTGC
TTAC                   TTAC
TACCGT                  TACCGT
                        ----------
                        TTACCGTGC
```

Consensus

Remarque : la longueur de la séquence cible est connue à 10% près

Complications du problème

1. Erreurs

– Erreurs de séquençage
– Erreurs dans le mécanisme de fragmentation et de clonage

⇒ Substitutions et indels (gaps) : 5%

```
ACCAGT             ⟹    ACCAGT
CGGT                       C-G-GT
TTAC                   TTAC
TGCCGT                  TGCC-GT
                        ----------
                        TTACC-GTGT
```

Consensus “vote de majorité”

2. Fragments chimériques

2 fragments distincts fusionnent pour n'en former qu'un seul

```
ACCGT              ⟹    ACCGT
CGTGC                      CGTGC
TTAC                   TTAC
TACCGT                  TACCGT
TTATGC     *            ---------
                        TTACCGTGC

                        TTA---TGC
```

Les fragments chimériques doivent être détectés et éliminés avant l'assemblage par des algorithmes spécifiques.

3. Contamination

Le clonage de fragments est réalisé par intégration du fragment au sein de l'ADN d'un vecteur. Après clonage, les fragments doivent être "purifiés" pour les nettoyer de l'ADN du vecteur.

Lorsque la purification n'est pas bien effectuée, on parle de *contamination* des fragments.

Les fragments contaminés doivent être éliminés **avant** l'assemblage.

Les séquences complètes de vecteurs sont bien connues $\Rightarrow$ recherche dans les banques.

4. Orientation inconnue

Les fragments sont orientés $5' \rightarrow 3'$ mais sur n'importe quel brin.

```
CACGT              ⟹  CACGT
ACGT                    ACGT
ACTACG                    CGTAGT
GTACT                        AGTAC
ACTGA                           ACTGA
CTGA                             CTGA
                       -------------
                       CACGTAGTACTGA
```

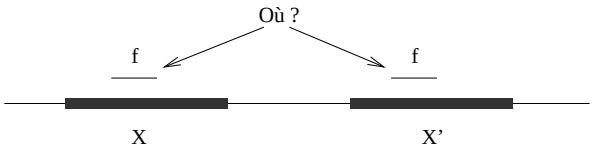
Considérer les fragments complémentaires et inversés également

$\Rightarrow$ un fragment sera présent soit tel quel, soit complémenté et inversé **(pas les 2)**

5. Régions répétées

Les petites répétitions entièrement recouvertes par des fragments ne posent pas de problème. Les longues répétitions posent des problèmes.

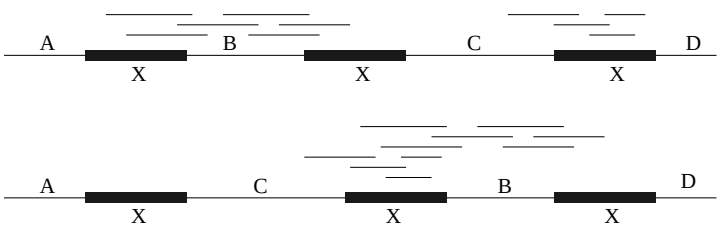
A. Fragment totalement recouvert par une répétition



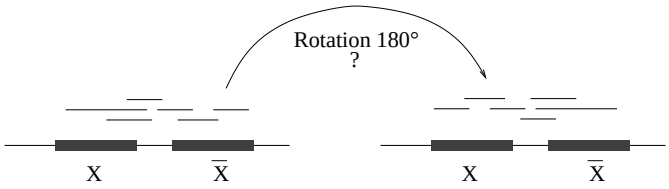
avec X proche de X'

L'endroit où on place f influe sur la séquence consensus

B. Alignement ambigu



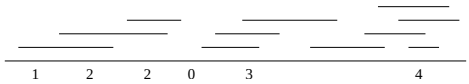
C. Répétitions inversées



6. Manque de couverture

Le mécanisme de fragmentation étant aléatoire, certaines positions risquent de ne pas être couvertes.

Couverture en position i = nombre de fragments couvrant i



La couverture est impossible à calculer car les positions des fragments sont inconnues.

Couverture moyenne = $\frac{\sum |f|}{|S|}$

S'il y a un manque de couverture, on obtient plusieurs *contigs*

La méthode de fragmentation doit être telle quel la couverture en chaque point soit > 1

En pratique, la couverture moyenne devrait être > 8

Remarque : Le draft de CELERA a été construit avec une couverture moyenne de 5 tandis que la séquence finie de la drosophile (début 2000) a été calculée avec une couverture moyenne de 10.

Modèles formels

Aucun modèle n'est pleinement satisfaisant !

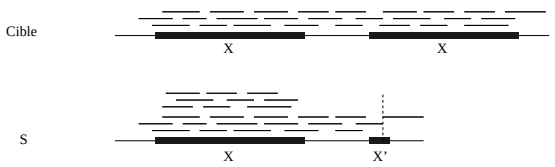
1. Plus courte "super-chaîne" commune

INPUT : Collection $\mathcal{F}$ de fragments
OUTPUT : **Plus courte** chaîne S telle que $\forall f \in \mathcal{F} : f$ facteur de S

Exemple : $\mathcal{F} = \{\texttt{ACT}, \texttt{CTA}, \texttt{AGT}\} \Rightarrow S = \texttt{ACTAGT}$

INCONVENIENTS :

– Ne permet pas les erreurs
– Les orientations des fragments doivent être connues
– La plus courte chaîne n'est pas toujours celle recherchée par les biologistes dans le cas de répétitions



– Couverture irrégulière
– Liaison manquante : aucune suite de fragments chevauchants ne lie le début de X' à la fin de X'

Problème NP-Complet $\Rightarrow$ approximations
Bonne base de travail pour le reste

2. Reconstruction

$d(a,b)$ =distance d'édition pour passer de a à b

Def : $d_s(a,b) = min_{s \in S(b)} d(a,s)$ où $S(b)$ est l'ensemble de tous les facteurs de b

Exemple : $a =$ `GCGATAG` et $b =$ `CAGTCGCTGATCGTACG` $\Rightarrow d_s(a,b) = 2$

```
   GC-GATAG
CAGTCGCTGATCGTACG
```

Def : Soit $\epsilon \in [0,1]$
f est un **facteur approximatif** de S au niveau ϵ si $d_s(f,S) \leq \epsilon |f|$

INPUT : Collection $\mathcal{F}$ de fragments et $\epsilon \in [0,1]$
OUTPUT : **plus courte** chaîne S telle que $\forall f \in \mathcal{F}$:

$$min(d_s(f,S), d_s(\bar{f},S)) \leq \epsilon |f|$$

INCONVENIENTS :
- Cas des répétitions
- Cas de faible couverture
- Cas de manque de liaison

Problème NP-Complet (plus compliqué que le modèle 1)

3. Multi-Contig

A. Sans erreur

Étant donné $\mathcal{F}$, on considère un alignement de ses fragments ou des complémentaires (1 seul des 2)

Exemple : $\mathcal{F} = \{$`GTAC`, `TAATG`, `TGTAA`$\}$

```
TGTAA
  TAATG
     GTAC
```

Def : Le **lien le plus faible** d'un alignement est la longueur du plus court chevauchement.
Seuls sont comptés les chevauchements qui ne sont pas complètement couverts par un autre fragment (les "non-liens").

Exemple :

```
GAAT          ⟹ 2
  TCGAGG
 ATCG
```

Def : un alignement est un t-**contig** si son lien le plus faible est $\geq t$

INPUT : Collection $\mathcal{F}$ de fragments et l'entier t
OUTPUT : Partition de $\mathcal{F}$ en un **nombre minimal** de classes $\mathcal{C}_i$ tel que chaque $\mathcal{C}_i$ admet un t-contig

Exemple : $\mathcal{F} = \{\texttt{GTAC}, \texttt{TAATG}, \texttt{TGTAA}\}$

$t = 3 \Rightarrow$ TAATG GTAC
TGTAA

$t = 2 \Rightarrow$ TAATG GTAC
TGTAA
ou TAATG GTAC
TGTAA

$t = 1 \Rightarrow$ TGTAA
TAATG
GTAC

B. Avec erreurs

L'alignement donne une séquence consensus

INPUT : Collection $\mathcal{F}$ de fragments, l'entier t et $\epsilon \in [0, 1]$
OUTPUT : Partition de $\mathcal{F}$ en un **nombre minimal** de classes $\mathcal{C}_i$ tel que chaque $\mathcal{C}_i$ admet un t-contig pour un ϵ-consensus
(La distance entre chaque fragment f et sa projection dans le consensus est au plus de $\epsilon|f|$)

Exemple : A-GTC
TCTCA
CTC-G

ATCTCAG
^^^^^

Problème NP-Complet
Résout l'assemblage dans certains cas des répétitions

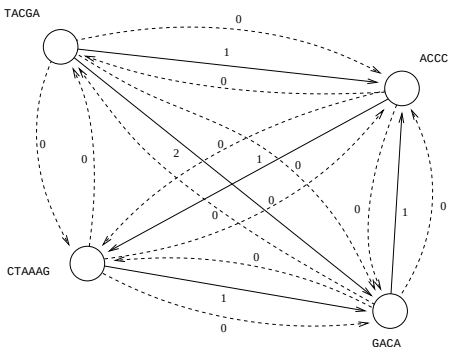
Algorithmes

Sans erreur, orientation connue

Représentation des chevauchements

Def : **Overlap Multigraph** $\mathcal{OM}(\mathcal{F})$: graphe orienté pondéré
– sommets : les fragments $f \in \mathcal{F}$
– arc de $f \in \mathcal{F}$ à $g \in \mathcal{F}$ de poids $t \geq 0$ si $suff(f, t) = pref(g, t)$ et $f \neq g$

Exemple : $a =$TACGA
$b =$ACCC
$c =$CTAAAG
$d =$GACA



Tout chemin décrit un alignement entre des fragments
Un chemin Hamiltonien décrit une super-chaîne

Soient :
- P un chemin de $\mathcal{OM}(\mathcal{F})$
- $A \subseteq \mathcal{F}$ les fragments correspondant à P
- $S(P)$ la super-chaîne dérivée de P

On a $||A|| = |S(P)| + w(P)$ avec $||A|| = \sum_{a \in A} |a|$ et $w(P)$ le poids de P.

Si on ne considère que les chemins Hamiltoniens, minimiser $|S(P)|$ revient à maximiser $w(P)$

Chemins et plus courte super-chaîne

Super-chaî ne $\xrightarrow{?}$ Chemin Hamiltonien

NON ! Mais la plus courte super-chaî ne : OUI

Def : $\mathcal{F}$ est **substring-free** si $\forall f, g \in \mathcal{F}, f \neq g$, on a f non facteur de g et g non facteur de f

Théorème : Soit $\mathcal{F}$ une collection substring-free de fragments. Si S est une plus courte super-chaîne composée des fragments de $\mathcal{F}$ alors il existe un chemin Hamiltonien P tel que $S(P) = S$

Def : Deux collections $\mathcal{F}$ et $\mathcal{G}$ sont **équivalentes** si $\forall f \in \mathcal{F}, \exists g \in \mathcal{G}$: f est facteur de g et $\forall g \in \mathcal{G}, \exists f \in \mathcal{F}$: g est facteur de f

Proposition : Soit $\mathcal{F}$ une collection. Il existe une et une seule collection substring-free équivalente à $\mathcal{F}$

Il convient donc d'" épurer" la collection $\mathcal{F}$ avant de rechercher la plus courte super-cha î ne

Algorithme d'approximation : *Greedy*

Problème NP-Complet $\Longrightarrow$ approximation

On cherche à maximiser les chevauchements $\Longrightarrow$ pour chaque paire de nœuds de $\mathcal{OM}(\mathcal{F})$, on ne conserve que l'arc de poids maximal.

$\Longrightarrow \mathcal{OG}(\mathcal{F})$: **Overlap Graph**

On ajoute progressivement les arcs de poids maximaux jusqu'à ce que le chemin contienne tous les nœuds

! ! ! Il faut empêcher la formation de cycles :
- on peut "entrer" dans un nœud au plus une fois
- on peut "sortir" d'un nœud au plus une fois

GREEDY$(\mathcal{OG}(\mathcal{F}), n)$

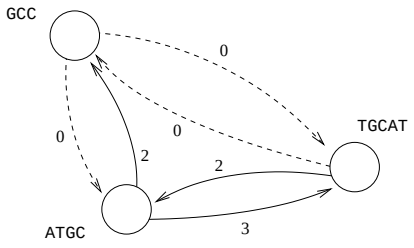
Pour $i \leftarrow 1$ **Jusque** n
Faire $in[i] \leftarrow 0; out[i] \leftarrow 0$
    MAKESET(i)
  Tri des arcs par poids décroissants
Pour Tout $arc(f, g)$ *par poid décroissant*
Faire Si $in[g] = 0$ **et** $out[f] = 0$ **et** FINDSET$(f) \neq$ FINDSET(g)
        Alors SELECT(f, g)
            $in[g] \leftarrow 1; out[f] \leftarrow 1$
            UNION(FINDSET(f), FINDSET(g))
    Si *il ne reste qu'un composant*
        Alors Break
Retourner *(arcs choisis)*

Avec :

– MAKESET(i) : initialise l'ensemble $\{i\}$
– FINDSET(f) : retourne l'ensemble contenant f
– UNION(E_1, E_2) : fusionne les 2 ensembles E_1 et E_2
– SELECT(f, g) : choisit $arc(f, g)$

GREEDY ne fournit pas toujours une solution optimale !

Exemple :



GREEDY :
```
ATGC
 TGCAT
      GCC
---------
ATGCATGCC
```
$\rightarrow 9$

Optimal :
```
TGCAT
   ATGC
     GCC
--------
TGCATGCC
```
$\rightarrow 8$

Graphes acycliques

Soit S une chaîne
Soit $A = \{[i, j] | 1 \leq i < j \leq |S|\}$ un échantillon d'intervalles de S

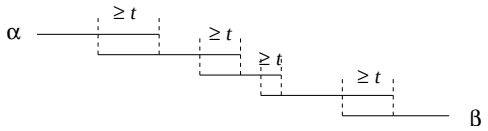
Dans quelle mesure la collection $\mathcal{F}$ de fragments engendrée par A permet de reconstruire S ?

Def : L'échantillon A **couvre** S si $\forall i : 1 \leq i \leq |S|$: i appartient à au moins un intervalle de A

Def : L'échantillon A est **sub-interval free** s'il ne contient pas deux sous-intervalles $[i, j]$ et $[k, l]$ avec $[i, j] \subseteq [k, l]$ (avec $i \neq k$ ou $j \neq l$)

Def : Deux intervalles $\alpha, \beta \in A$ sont **liés au niveau** t si $|\alpha \cap \beta| \geq t$

Def : L'échantillon A est connecté au niveau t si pour tous les intervalles $\alpha, \beta \in A$ (α "avant" β), il existe une suite d'intervalles $\alpha_i \in A : \alpha = \alpha_0, \beta = \alpha_k$ et α_i lié au niveau t à α_{i+1} avec $i < k$



La qualité d'un échantillon A dépend de si :
- A couvre S
- A est sub-interval free
- A est connecté au niveau t (≈ 10 en pratique)

Def : $\mathcal{OM}(\mathcal{F}, t)$: $\mathcal{OM}(\mathcal{F})$ avec uniquement les arcs de poids $\geq t$

Def : $\mathcal{OG}(\mathcal{F}, t)$: $\mathcal{OG}(\mathcal{F})$ avec uniquement les arcs de poids $\geq t$

Théorème : Soit S une séquence et A un échantillon de S subinterval-free et connecté au niveau t pour $t \geq 0$. Le multigraphe $\mathcal{OM}(\mathcal{F},t)$, où $\mathcal{F}$ est la collection de fragments engendrée par A, possède un chemin Hamiltonien P. De plus, si A couvre S, alors P peut être choisi tel que $S = S(P)$.

Les cycles dans un graphe de chevauchements sont nécessairement dus aux répétitions dans S :

Théorème : Soit $\mathcal{F}$ une collection engendrée par un échantillon A de S. Si $\mathcal{OG}(\mathcal{F},t)$ contient un cycle, alors S possède une répétition de longueur $\geq t$.

Lorsque S ne contient pas de répétition, l'assemblage de fragments peut être facilement résolu :

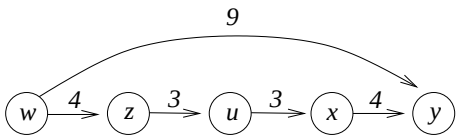
Théorème : Soit S une séquence, A un échantillon de S, connecté au niveau t, qui couvre S et subinterval-free, et soit $\mathcal{F} = S[A]$.
Si S n'a aucune répétition de longueur $\geq t$, alors le graphe $\mathcal{OG}(\mathcal{F}, t)$ a un unique chemin Hamiltonien P et $S = S(P)$.

Algorithme : Tri topologique

- on choisit le seul sommet n'apparaissant à la queue d'aucun arc
- on le supprime (le fragment est intégré à la super-chaîne)
- on recommence jusqu'à ce qu'il ne reste plus aucun sommet

Exemple :

w=AGTATTGGCAATC
z =AATCGATG
u =ATGCAAACCT
x =CCTTTTGG
y =TTGGCAATCACT



Solution Greedy :

```
AGTATTGGCAATC   AATCGATG
    TTGGCAATCACT     ATGCAAACCT
                              CCTTTTGG
-----------------------------------
AGTATTGGCAATCACTAATCGATGCAAACCTTTTGG
```

Longueur : 36, lien le plus faible : 0

Solution tri topologique :

```
AGTATTGGCAATC             TTGGCAATCACT
         AATCGATG    CCTTTTGG
              ATGCAAACCT
-----------------------------------
AGTATTGGCAATCGATGCAAACCTTTTGGCAATCACT
```

Longueur : 37, lien le plus faible : 3

$\Longrightarrow$ **Meilleure liaison !**

Cas réel : erreurs et fragments non orientés

L'assemblage de fragments peut être vu comme un alignement multiple où les gaps internes ne sont pas comptabilisés de la même façon que les gaps externes.

```
ACCTGAA
   TGC-CC
    GCGACT
   TCCA
----------
ACCTGCANCT
```

Les fragments sont généralement courts par rapport à la taille de l'alignement.

$\Longrightarrow$ Comment évaluer la qualité d'un assemblage ?

Score : En chaque colonne, on calcule l'entropie :

$$E = -\sum_{c|P_c \neq 0} P_c \log P_c$$

avec $P_{\mathsf{A}}, P_{\mathsf{C}}, P_{\mathsf{G}}, P_{\mathsf{T}}$ et $P_{\text{-}}$ les probabilités d'apparition des symboles ou de gaps.

$E = 0 \Longrightarrow$ uniformité +++
$E = \log 5 \Longrightarrow$ variabilité ---

Couverture : Notion étendue : la position d'un gap interne d'un fragment est couverte par le fragment.

$\Longrightarrow$ calcul de la **couverture minimale, maximale et moyenne**

Liaison :

```
        ACTTTT
TCCGAG       ACGGAC
        ACTTTT
TCCGAG       ACGGAC
        ACTTTT
TCCGAG       ACGGAC
------------------
TCCGAGACTTTTACGGAC
```

Bonne couverture mais mauvaise liaison

$\Longrightarrow$ Indétermination : interchangeabilité des morceaux

Assemblage en pratique ?

3 étapes :

1. rechercher les chevauchements approximatifs
2. construire l'alignement
3. calculer la séquence consensus

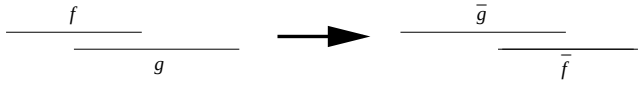
Les méthodes font souvent appel à des heuristiques !

1. Recherche des chevauchements

Considérer toutes les paires de fragments et leurs complémentaires.

$\Longrightarrow$ Alignement semi-global par programmation dynamique

Remarque : On peut éviter la programmation dynamique entre deux fragments s'ils ne présentent pas assez de similitudes (vérification en temps linéaire grâce aux arbres des suffixes)



$\Rightarrow$ uniquement (f, g) ainsi que $(f, \overline{g})$

		A	T	C	G	G	C	A	T	T	C	A	G	T
	0	0	0	0	0	0	0	0	0	0	0	0	0	0
A	0	1	-1	-1	-1	-1	-1	1	-1	-1	-1	1	-1	-1
T	0	-1	2	0	-2	-2	-2	-1	2	0	-2	-1	0	0
T	0	-1	0	1	-1	-3	-3	-3	0	3	1	-1	-2	1
A	0	1	-1	-1	0	-2	-4	-2	-2	1	2	2	0	-1
G	0	-1	0	-2	0	1	-1	-3	-3	-1	0	1	3	1
A	0	1	-1	-1	-2	-1	0	0	-2	-3	-2	1	1	2
C	0	-1	0	0	-2	-3	0	-1	-1	-3	-2	-1	0	0
C	0	-1	-2	1	-1	-3	-2	-1	-2	-2	-2	-3	-2	-1
A	0	1	-1	-1	0	-2	-4	-1	-2	-3	-3	-1	-3	-3
T	0	-1	2	0	-2	-1	-3	-3	0	-1	-3	-3	-2	-2
G	0	-1	0	1	1	-1	-2	-4	-2	-1	-2	-4	-2	-3
C	0	-1	-2	1	0	0	0	-2	-4	-3	0	-2	-4	-3
G	0	-1	-2	-1	2	1	-1	-1	-3	-5	-2	-1	-1	-3
G	0	-1	-2	-3	0	3	1	-1	-2	-4	-4	-3	0	-2
C	0	-1	-2	-1	-2	1	4	2	0	-2	-3	-5	-2	-1

```
ATTAGACCATGCGGC
   AT CGGCATTCAGT
```


2. Ordonner les fragments (greedy ou tri topologique)

Les sommets du graphe : $\mathcal{DF} = \mathcal{F} \cup \overline{\mathcal{F}}$ avec $\overline{\mathcal{F}} = \{\overline{f} | f \in \mathcal{F}\}$

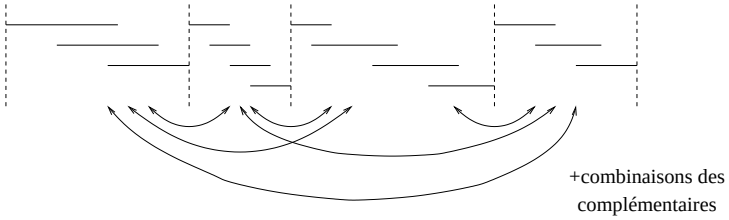
Si un fragment apparaî t dans le chemin, son complémentaire ne peut plus être choisi !

Remarques :

– le complémentaire d'un "bon chemin" est également un "bon chemin" :

$$\overline{f_1} \rightarrow \overline{f_2} \rightarrow \ldots \overline{f_k} \text{ et } \overline{f_k} \rightarrow \overline{f_{k-1}} \rightarrow \ldots \overline{f_1}$$

– Une multitude de solutions sont valides si plusieurs contigs.
Il s'agit là d'un problème de liaison.



3. Alignement et consensus

Les chevauchements approximatifs compliquent les choses

Exemple : $f \rightarrow g \rightarrow h$ avec f =CATAGTC
g =TAACTAT
h =AGACTATCC

2 alignements entre f et g (même score) :

```
CATAGTC          ou      CATAGTC
  TAA-CTAT                 TA-ACTAT
```

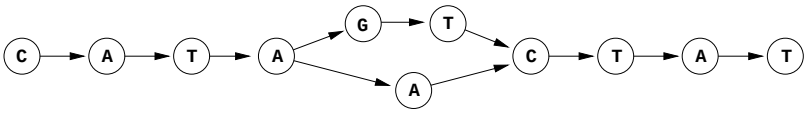
Lorsqu'on veut introduire h dans l'alignement, le second est meilleur :

```
CATAGTC
  TA-ACTAT
   AGACTATCC
-----------
CATAGACTATCC
```

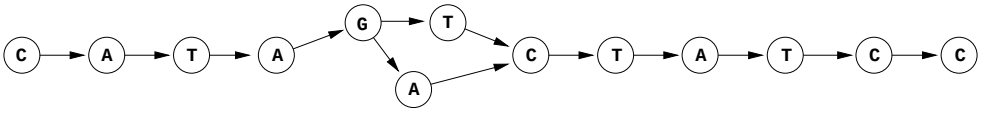
Pour résoudre cela, chaque alignement est représenté par un graphe acyclique dans lequel les match se partagent les mêmes nœuds et les autres bases sont séparées.

Exemple :

f et g :



avec h :



Autre problème : ambiguïté dans les alignements 2 à 2.

```
ACT-GG          ACT-GG
ACTTGG          ACTTGG
AC-TGG          ACT-GG
ACT-GG          ACT-GG
AC-TGG          ACT-GG
------          ------
ACTTGG          ACT-GG
```

Le deuxième est préférable.

On choisit une courte région nécessitant une amélioration et on effectue un alignement multiple sur les portions choisies.

Bibliographie

1. International Human Genome Sequencing Consortium, "Initial sequencing and analysis of the human genome",*Nature, vol. 409, pp. 860–921*, 15 feb 2001
   `http ://www.nature.com/cgi-taf/dynapage.taf?file=/nature/journal/v409/n6822/index.html`
2. J.C. Venter et al, "The Sequence of the Human Genome", *Science, vol. 291, pp. 1304–1351*, 16 feb 2001
   `http ://www.sciencemag.org/content/vol291/issue5507/`
3. J. Setubal et J. Meidanis, "Introduction to Computational Molecular Biology - Chapter 4",*PWS Publishing Company*, 1997