Assemblage *de novo*

Assemblage de la séquence du génome

Entrée : lectures d'ADN (appariées en général)

Sortie : (longues) séquences de régions contigues, déterminées selon chevauchements entre les fragments

```
s1: AATGCC.......................GGATTC
s2:    GCCTTACAC.......................AGGATTC
s3:         ACACTG.....................TCAAG
s4:           ACTGAAGG.....................ATTC
s5:              GAAGGTTTA.....................CGGACC
---------------------
B : AATGCCTTACACTGAAGGTTTA........GGATTCAAGGATTC..CGGACC
```

TIGR assembler (1995)

une approche glouton utilisée pour assembler le génome de *H. influenzae*

1. analyse de k-mers dans les fragments : chevauchements potentiels entre fragments avec k-mers partagés (score determiné par nombre de k-mers en commun)
2. identification de fragments avec régions répétées
3. initialisation de la séquence assemblée (contig) par un fragment
4. répéter : ajout du meilleur fragment à la séquence assemblée

En 2 : fragment avec trop de chevauchements potentiel=repeat

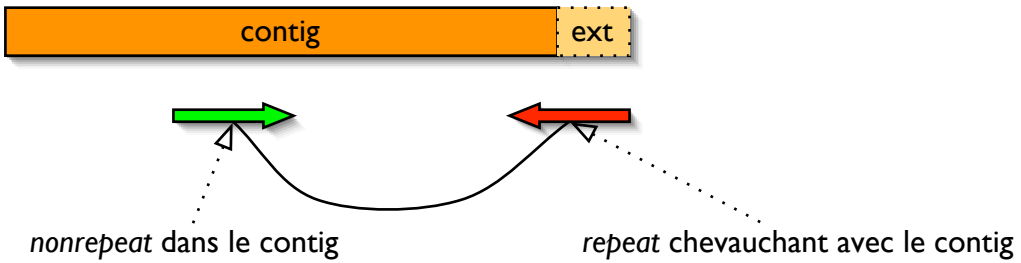
Sutton et al, *Genome Sci. Techn.* 1 : 9 (1995)

TIGR 2

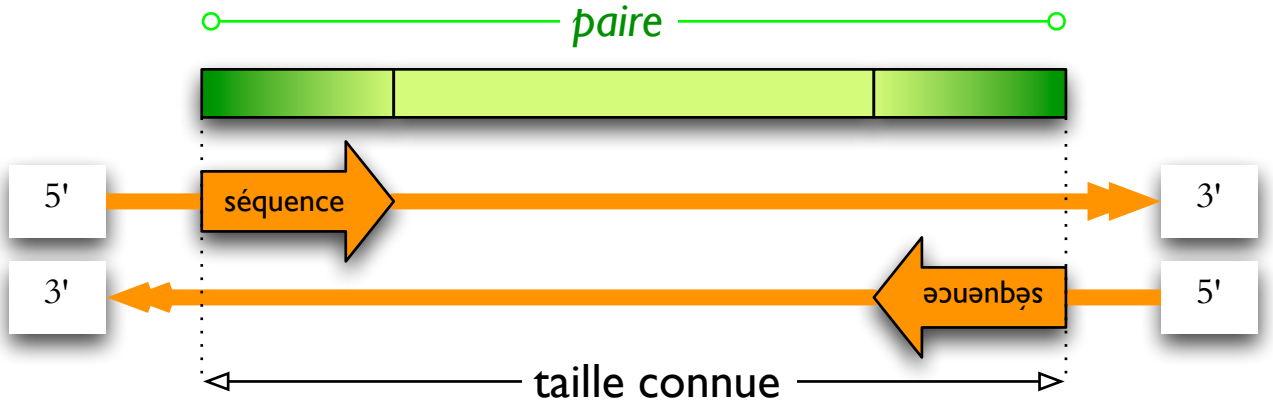
trouver meilleur fragment à ajouter : alignement local (Smith-Waterman), pour tous les fragments avec chevauchements potentiels

contig finit s'il n'y a plus de fragments à ajouter : à la frontière d'une région répétée ou à un vrai trou

extension dans la région répétée : utiliser des séquences shotgun appariées (*mate pairs*)



Mate pairs

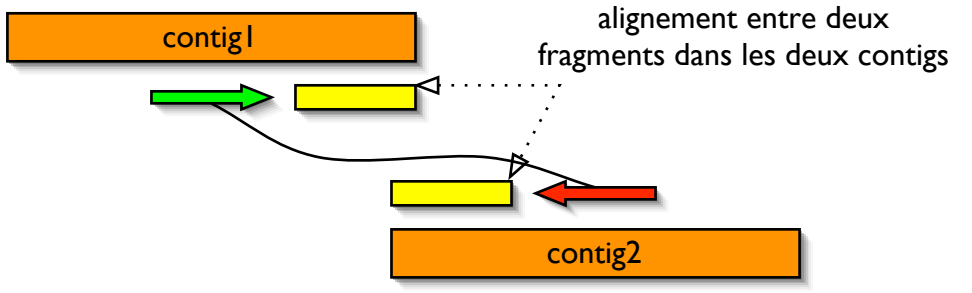


distances : 2k (M13), 10k (plasmid), 100k (BAC)

aident à orienter des contigs, à construire des ossatures (*scaffolds*), et à traverser des régions répétées

TIGR 3

joindre des contigs si évidence par chevauchements et mate pairs



Assemblage : overlap-layout-consensus

Overlap : déterminer les chevauchements parmi les séquences shotgun

Layout : déterminer l'ordre des séquences shotgun

Consensus : déterminer la séquence des contigs

Calcul de tous les chevauchements

$\mathcal{F} = \{f_1, \ldots, f_n\}$ ensemble de séquences shotgun

trouver le meilleur chevauchement entre chaque paire : $O(n^2\ell^2)$ où ℓ est une borne supérieure sur la longueur des fragments.

Améliorer :

1. moins de paires comparés (hachage par k-mers)
2. trouver le meilleur chevauchement plus rapidement (alignement rapide)

$\Rightarrow$ graphe de chevauchements (*overlap graph*)

Problème théorique : *shortest superstring* — NP-difficile
($\rightarrow$ mauvaise abstraction — on a des beaucoup de régions répétées dans le génome !)

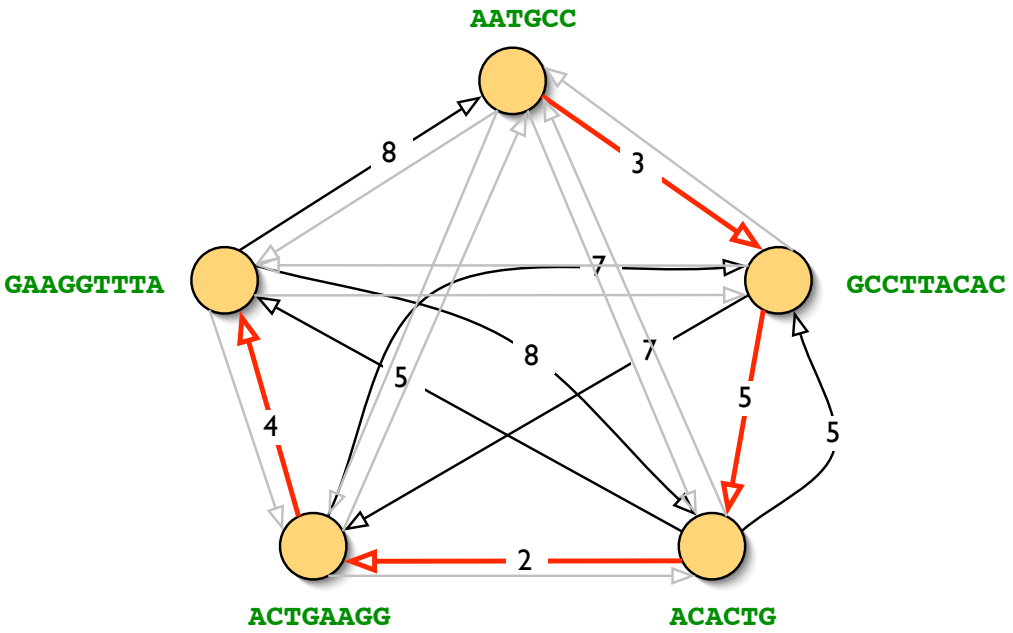
Un exemple (CAP3)

Séquence combinée de tous les fragments:

1. Tableau de hachage des k-mers de la séquence combinée.
2. Chaque fragment f ainsi que son complément sont comparés à la séquence combinée, pour trouver des HSPs (v. BLAST).
3. Les chevauchements potentiels de 2) sont évalués par alignement dans une bande.

Huang et Madan, *Genome Research* 9 :868 (1999)

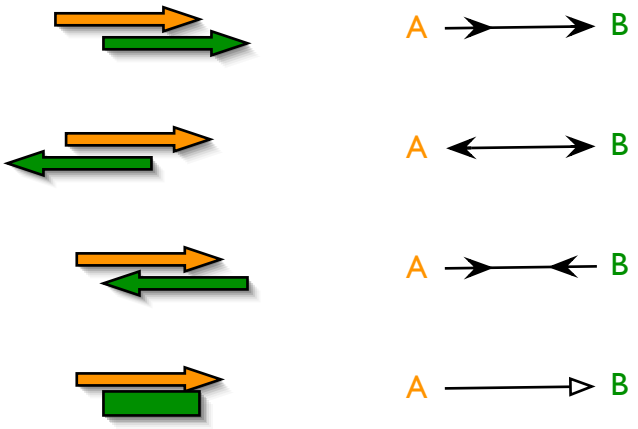
Graphe de chevauchements



[problème d'orientation ignoré]

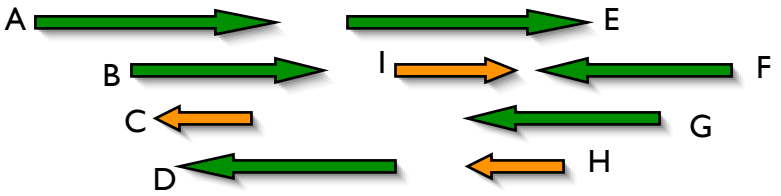
Layout

Catégories de chevauchements (orientation inconnue des fragments)



Myers, *J Comput Biol* 2 : 275 (1995)

Exemple



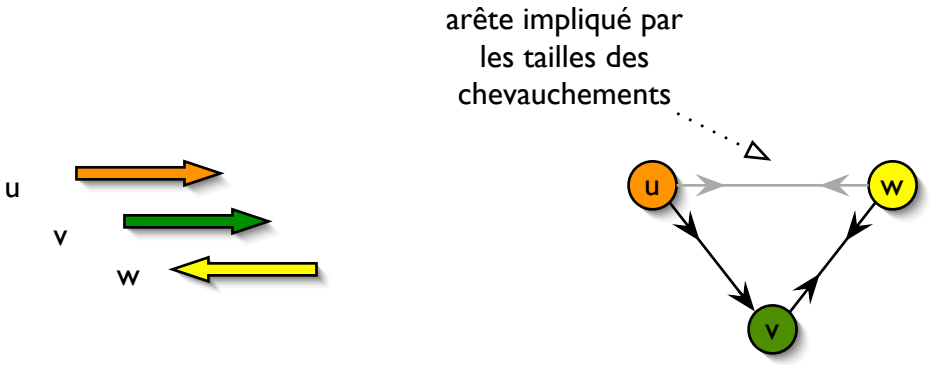
compter les flèches arrivant à un vertex : layout=chemin avec arêtes de chevauchements propres+ forêts avec arêtes de contention

Myers, *J Comput Biol* 2 : 275 (1995)

Simplification du graphe

1. enlever les arêtes de couverture complet

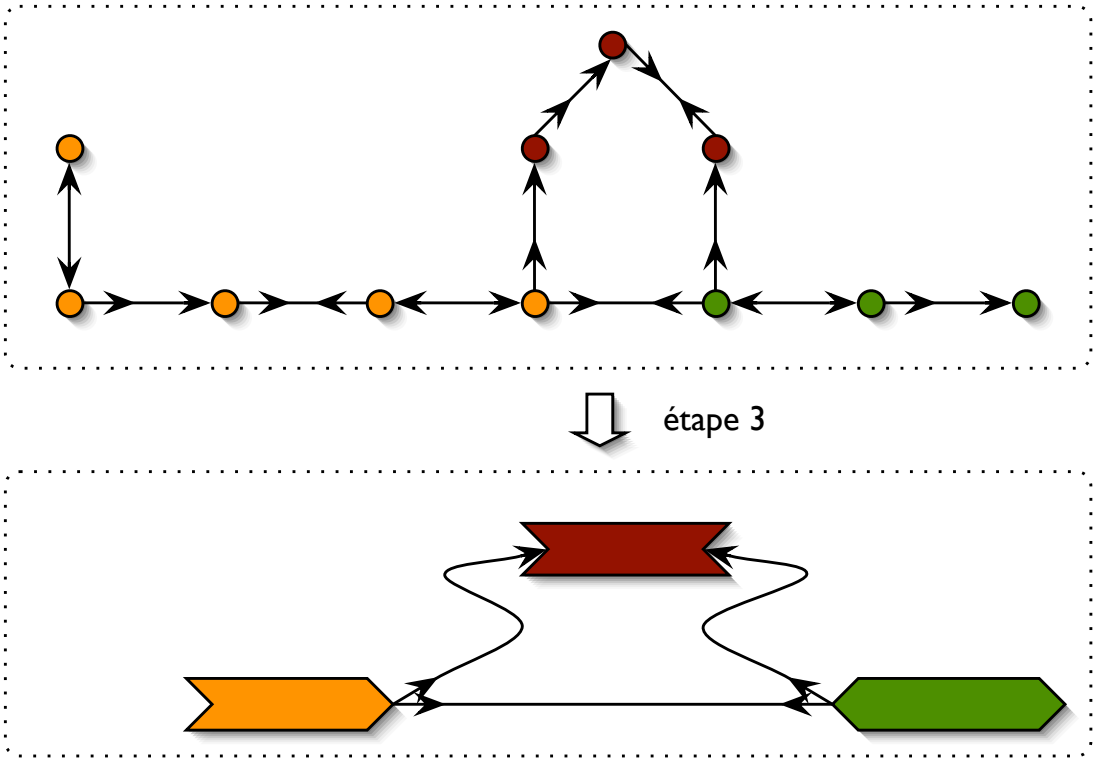
2. enlever des arêtes «transitifs» : si $u \rightleftharpoons v$, $v \rightleftharpoons w$ et $u \rightleftharpoons w$ sont des arêtes compatibles (orientation+taille de chevauchements), alors enlever $u \rightleftharpoons w$



Myers, *J Comput Biol* 2 : 275 (1995)

Simplification du graphe 2

3. collapser des chemins : «super-vertices» ou contigs



Myers, *J Comput Biol* 2 : 275 (1995)

Séquence de consensus

Le layout donne la position approximative de chaque fragment. Trouver la séquence de consensus : alignement multiple

Profile : enregistrer la fréquence de symbols dans l'alignement multiple

Joindre les séquences consécutives au contig dans l'ordre spécifié dans la phase *layout*, maintenir un profile dans le contig

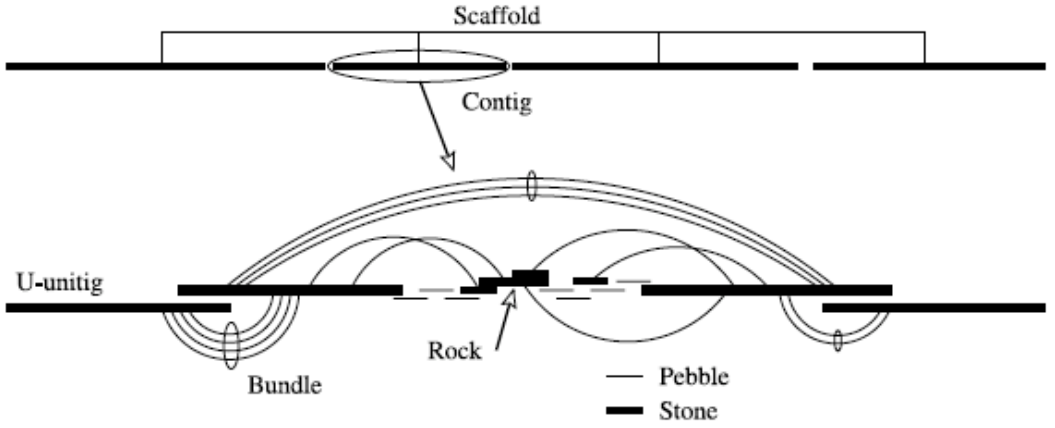
Problème : alignement d'une séquence à un profile (dans une bande autour de la position approximative)

$\Rightarrow$ contigs

Ossatures

Joindre des contigs si liens entre eux par deux *mate pairs* ou plus.

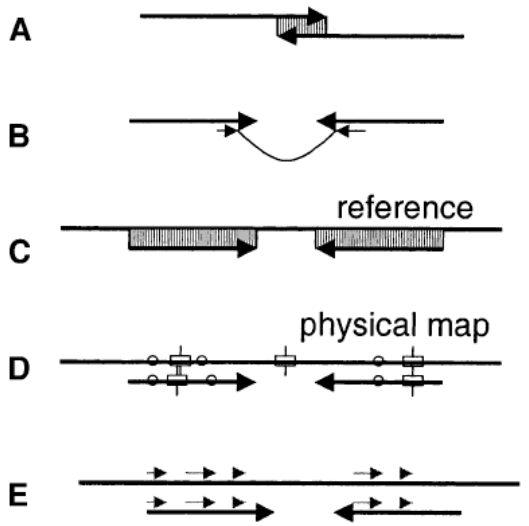
Ajout d'autres contigs dans les trous des ossatures



Myers & al, *Science* 287 :2196 (2000)

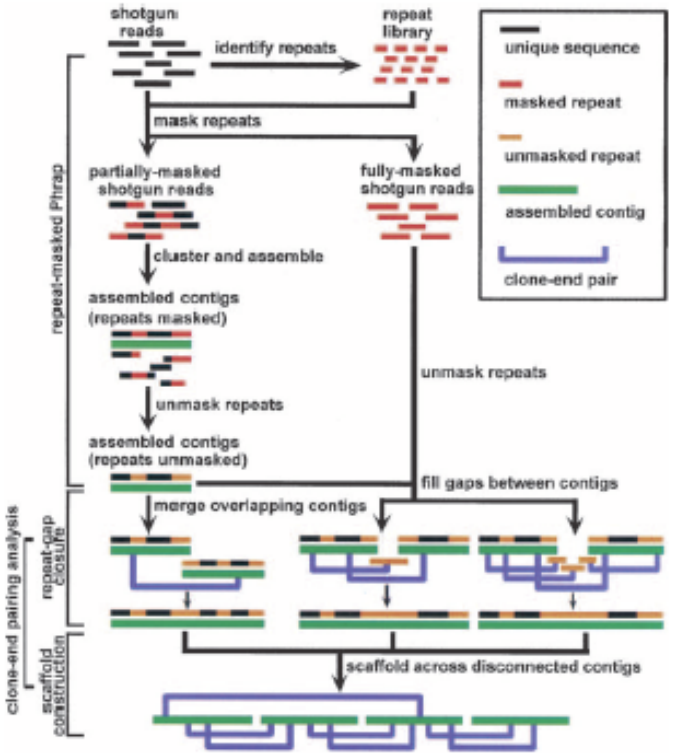
Ossatures — liens

Sources de liens : chevauchements entre contigs, mate pairs, alignement à un autre génome de référence, alignement à une carte physique, conservation de synténie



Pop & al, *Genome Res* 14 : 149 (2004)

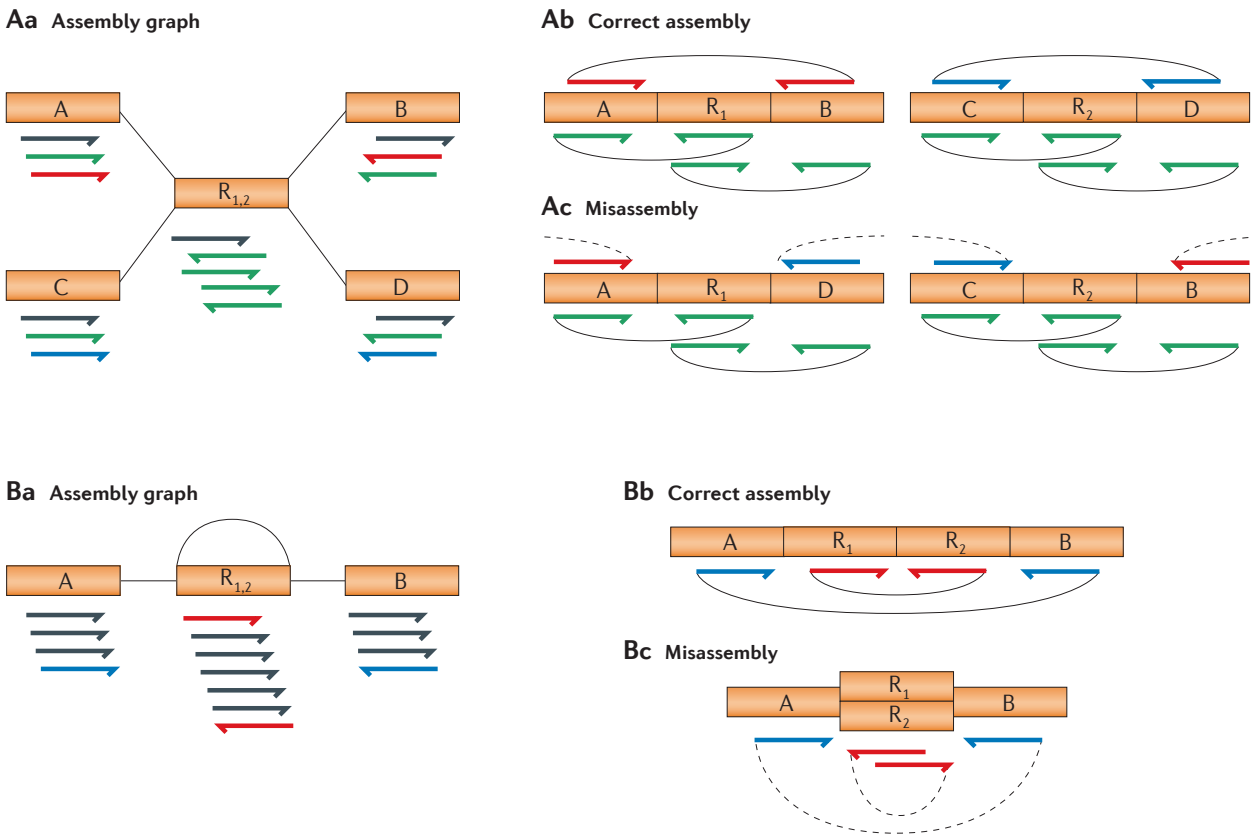
De séquences à d'ossatures



Wang & al, *Genome Res* 12 : 824 (2002)

Repeats

la plus difficile est de séquencer les régions répétées : danger de réarragment et compression



Treangen & Salzberg *Nat Rev Genet* 13 :36 (2012) ’

Séquençage par hybridation (SBH)

Idée : $\mathcal{C}$ ensemble de sondes (p.e. tous les k-mers) arrangé sur une puce

On teste la présence de chaque $c \in \mathcal{C}$ dans une molécule d'ADN (séquence u) par hybridation

spectrum : ensemble de k-mers dans u

$$\mathcal{S}_k(u) = \{u[i, \cdots, i+k-1] : i = 1, \cdots, |u| - k + 1\}.$$

Problème : reconstruction de u à partir de $\mathcal{S}_k(u)$.

Reconstruction à partir du spectrum

Approche 1 : graphe de chevauchements où chaque k-mer de u est un vertex et des arêtes représentent des chevauchements de taille $(k-1)$. On cherche un chemin hamiltonien.

Approche 2 : g.d.c où chaque k-mer de u est une arête entre son préfixe et suffixe de taille $(k-1)$ [quand le spectrum inclut tous les k-mers, c'est un graphe de Bruijn] On cherche un chemin eulérien.

Chemin eulérien

Déf Un chemin/cycle eulérien d'un graphe visite chaque arête exactement une fois.

Thm. Il existe un cycle eulérien dans un graphe connecté orienté ssi $\text{degré}_{arr}(u) = \text{degré}_{sort}(u)$ dans chaque vertice u.

Il existe un algorithme qui trouve un cycle [ou chemin] eulérien (ou annonce qu'il n'y en a pas) en temps linéaire.

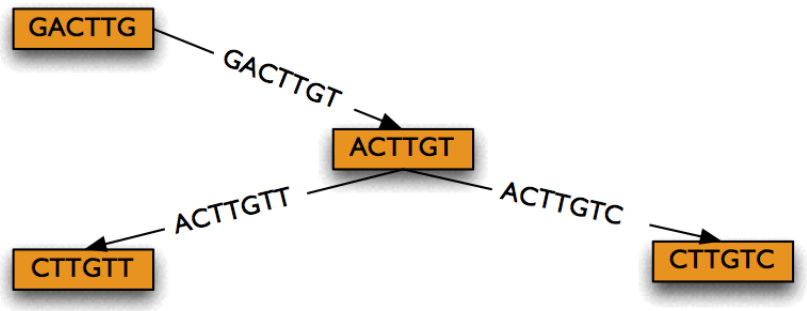
Séquençage shotgun par SBH

Il est désirable de calculer le spectrum avec un grand k (longueur d'une séquence qui peut être recounstruite est $\approx 2^k$) mais il n'est pas [encore] pratique d'utiliser des puces avec $k = 20$ p.e.

Idée : pourquoi ne pas générer le spectrum à partir de séquences shotgun ?

1. ensemble $\mathcal{F} = \{s_1, s_2, \ldots, s_n\}$ de séquences shotgun
2. ensemble de k-mers (p.e. $k = 24$) $\mathcal{F}_k = \bigcup_{i=1}^{n} \{s_i[j..j+k-1] : j = 1, \ldots, |s_i| - k + 1\}$
3. assembler la séquence à partir de $\mathcal{F}_k$ comme en SBH (chemins eulériens)

Idury-Waterman



1. Réduction du graphe par transformations pareilles à ce qu'on a vu pour layout (Myers 1995).

2. Chemins eulériens modifiés : la même arête peut être visitée plus qu'une fois (p.e. région répétée) ; quelques arêtes ne sont pas visitées de tout (erreurs de séquençage)

Idury et Waterman, *J Comput Biol* 2 : 291 (1995)

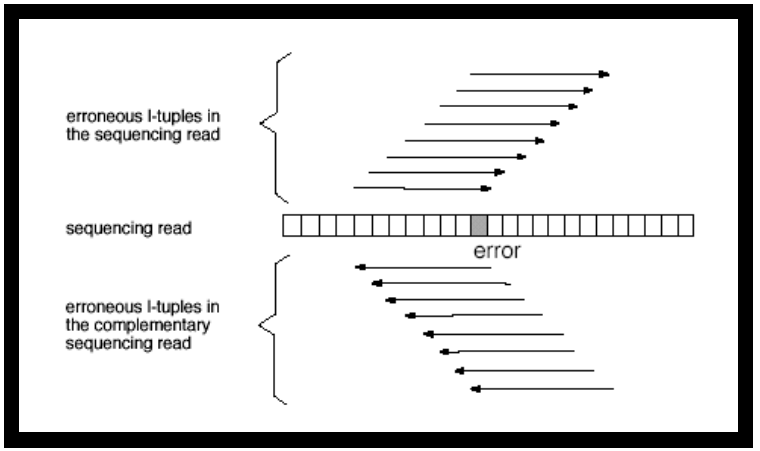
Euler

Une autre approche à l'assemblage shotgun inspirée par SBH : Euler

- correction d'erreurs : identification de k-mers rares
- augmentation de graphe de Bruijn pour transformer le problème du super-chemin eulérien en celui du chemin eulérien

Pevzner, Tang & Waterman, *RECOMB* 256 (2001)

Euler : erreurs



problème avec Idury-Waterman : trop de sommets dans le graphe créés par des erreurs de séquençage

solution de Euler : orphelin est un rare k-mer u dans le [multi-]spectrum $\mathcal{S}$ t.q. il existe exactement un $v \in \mathcal{S}$ avec $\|u - v\| = 1$ et v est fréquent — remplacer u par v.

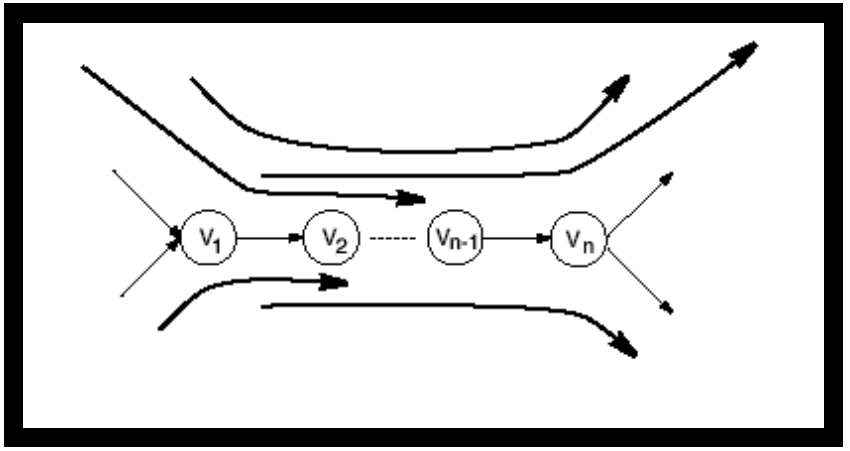
Pevzner, Tang & Waterman, *RECOMB* 256 (2001)

Euler : super-chemins

Départ : ensemble de k-mers avec info sur leurs occurrences (lecture shotgun + position dans la lecture)

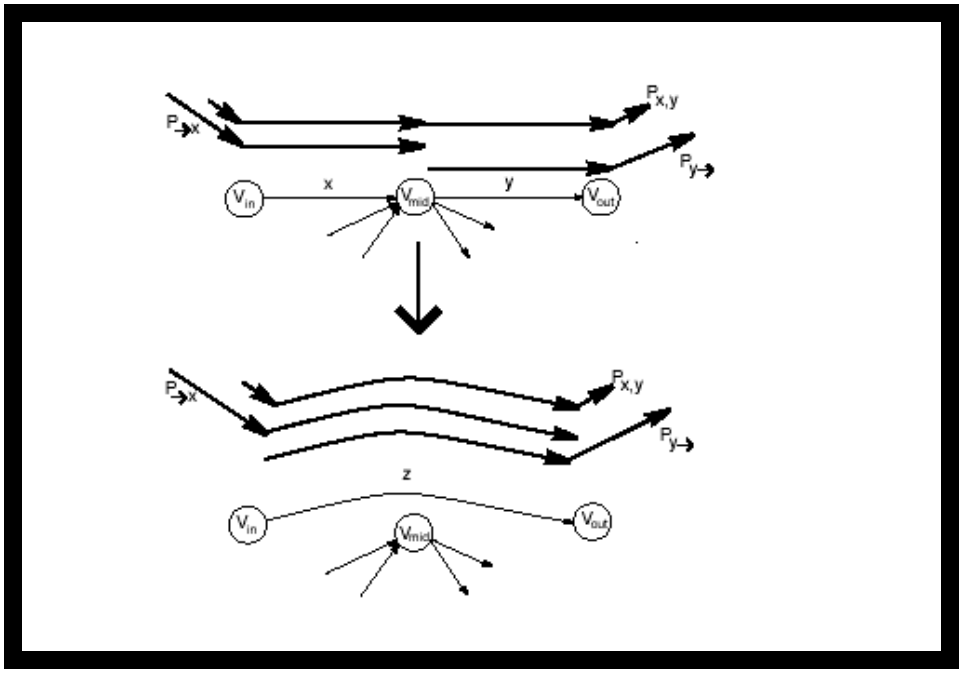
construire le graphe de Bruijn à arêtes multiples (une arête pour chaque occurrence d'un k-mer)

chemins initiels : définis par les séquences shotgun



Pevzner, Tang & Waterman, *RECOMB* 256 (2001)

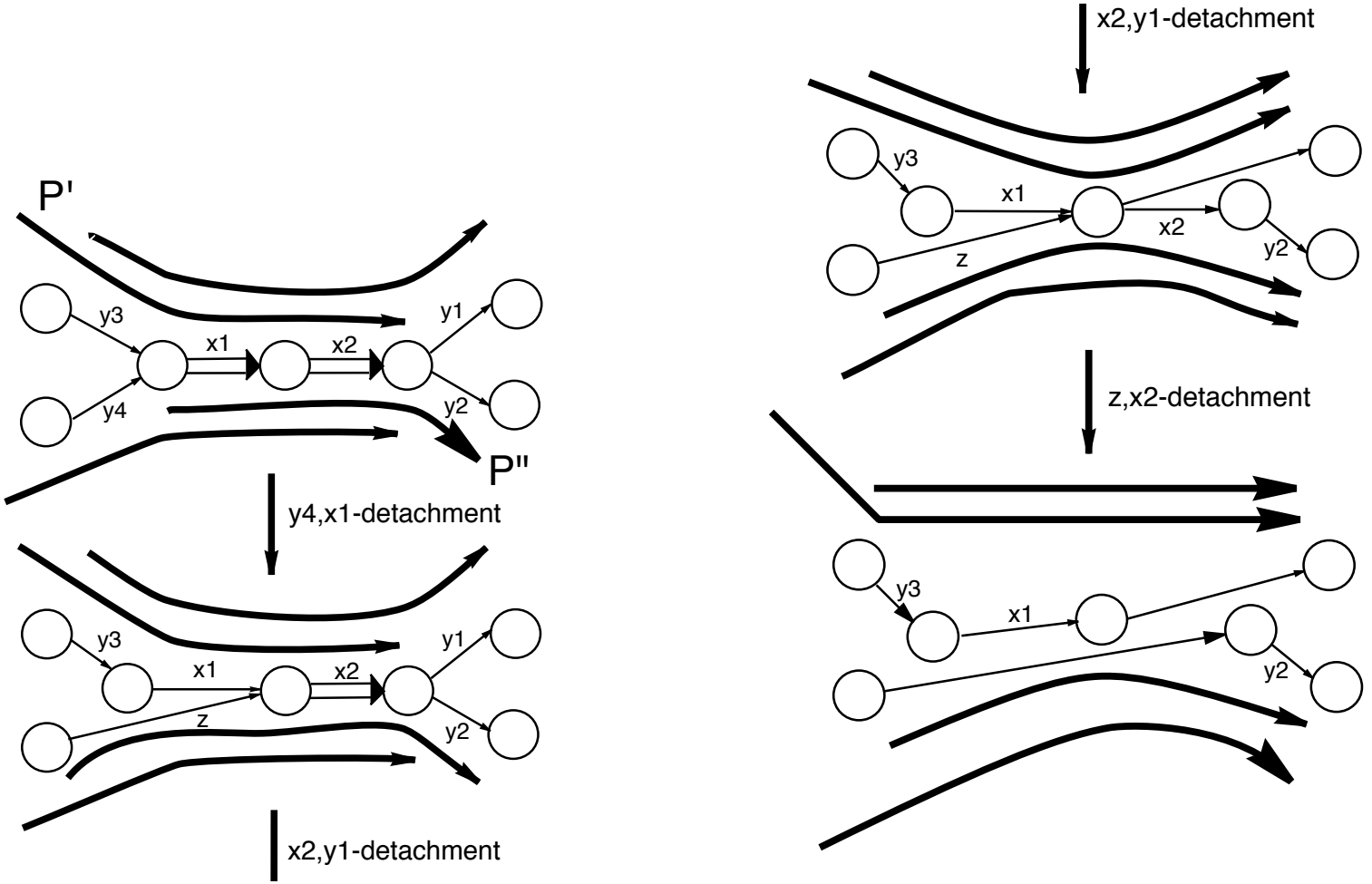
Euler : détachements



\+ vérifier quels chemins sont consistents

Pevzner, Tang & Waterman, *RECOMB* 256 (2001)

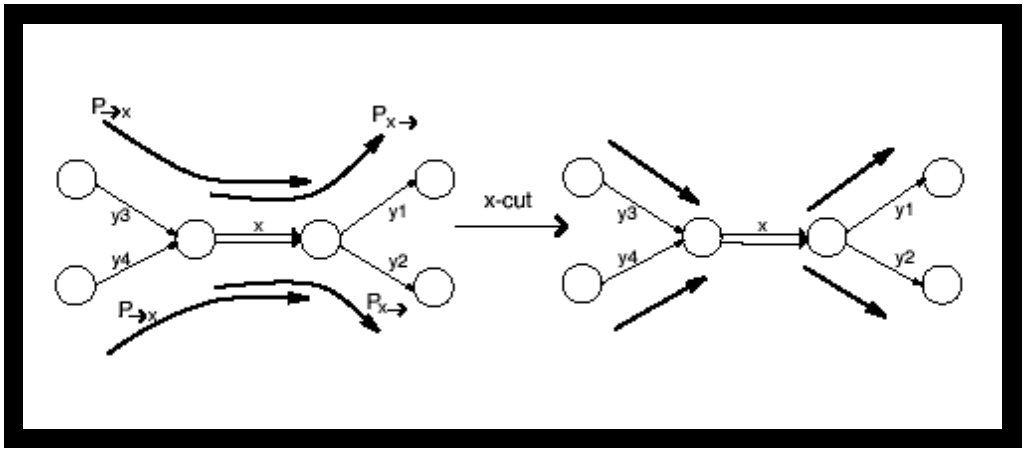
Euler : détachements 2



Pevzner, Tang & Waterman, *RECOMB* 256 (2001)

Euler : coupures

On ne peut pas décider. . . 2 contigs



Pevzner, Tang & Waterman, *RECOMB* 256 (2001)

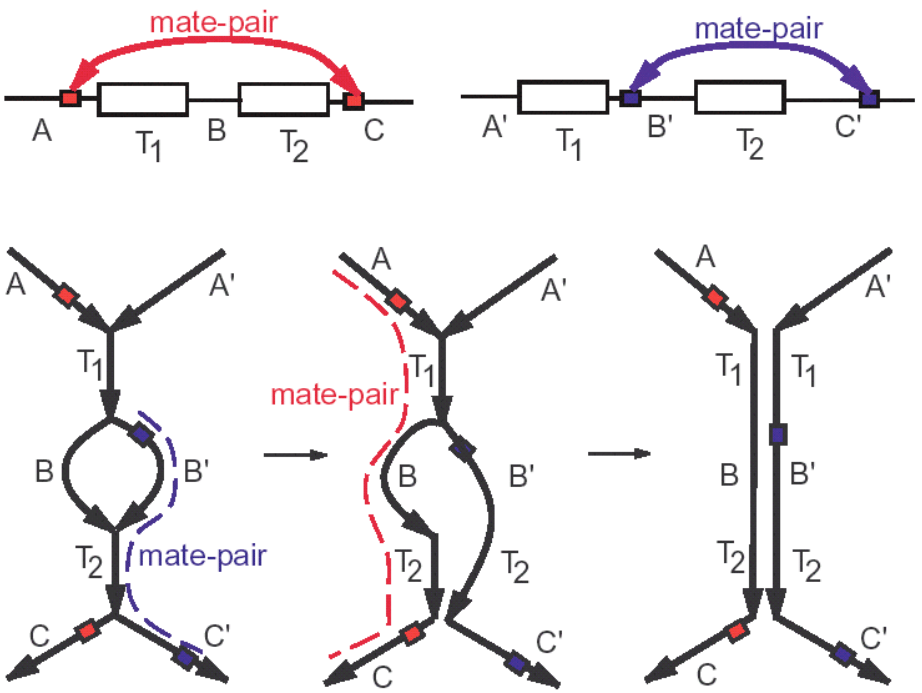
Philosophies

Euler : transformations de graphe pour obtenir un graphe eulérien

Idury-Waterman : réductions de graphe qui remplacent les chemins par des arêtes

Mate pairs et de Bruijn

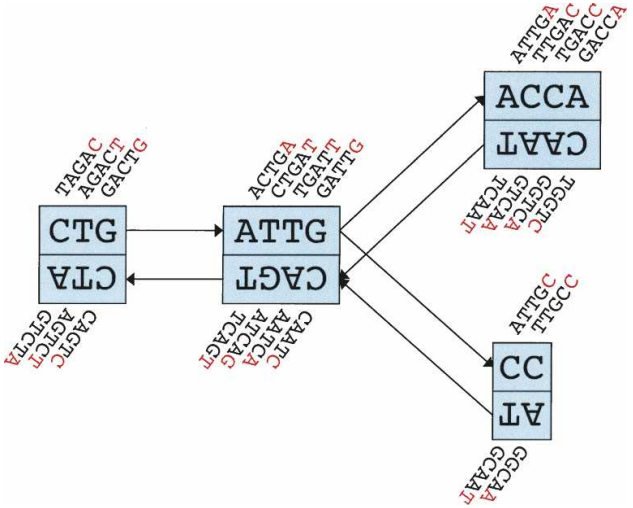
utiliser les *mate pairs* pour séparer les chemins



Pevzner & Tang, *Bioinformatics* 17 : S225 (2001)

Velvet

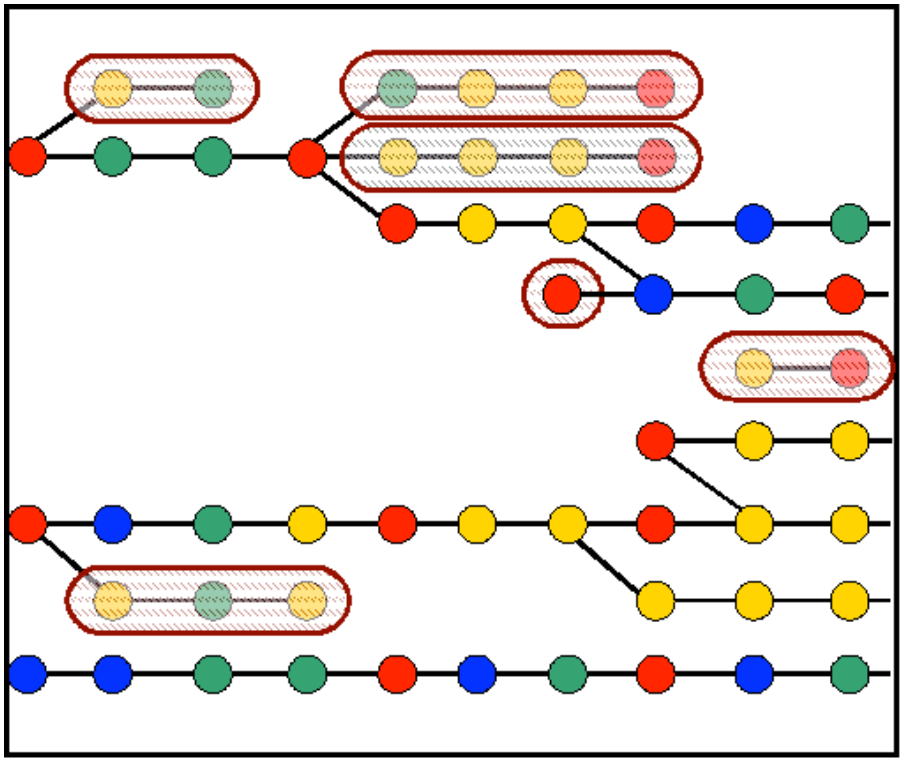
un nœud correspond à une séquence et son complément inverse : au début pour k-mers ($k = 21$), après fusionner si aucune ambiguïté



manipulation : correction d'erreurs (*tips* et *bubbles*)

Zerbino & Birney *Genome Res* 18 :821 (2008)

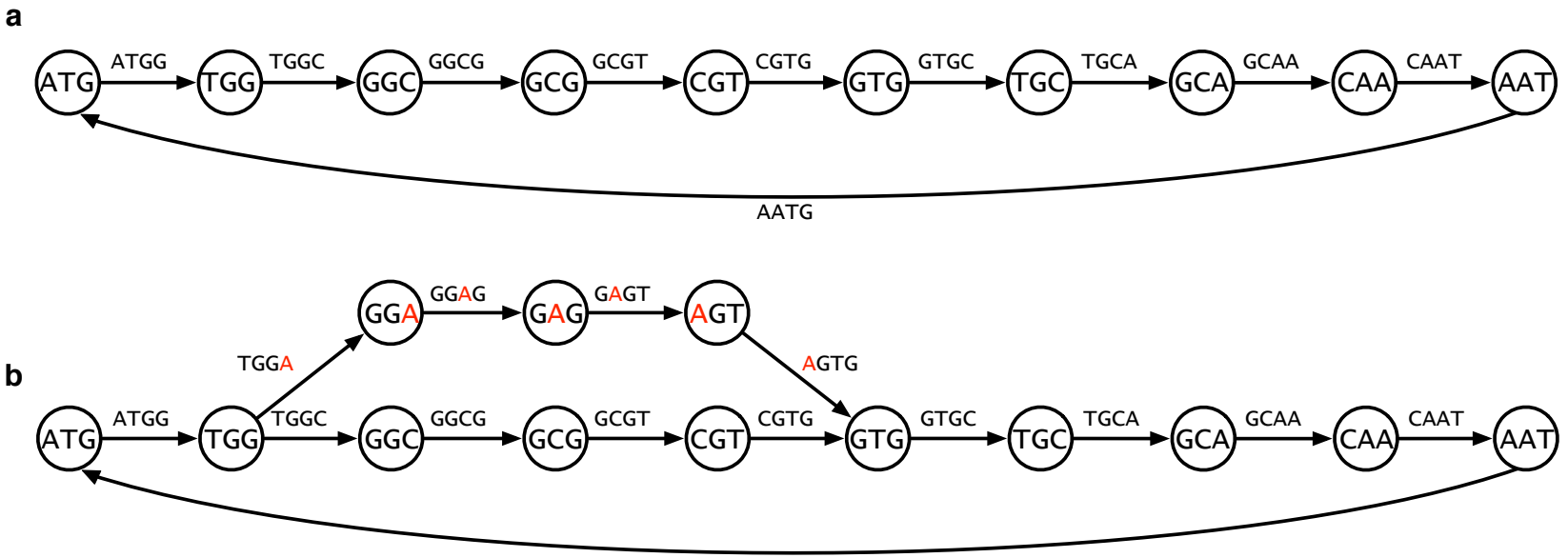
Tips



enlever cul-de-sac (longueur $\leq 2k$) — identifier avec parcours par profondeur

Simpson & al *Genome Res* 19 :1117 (2010)

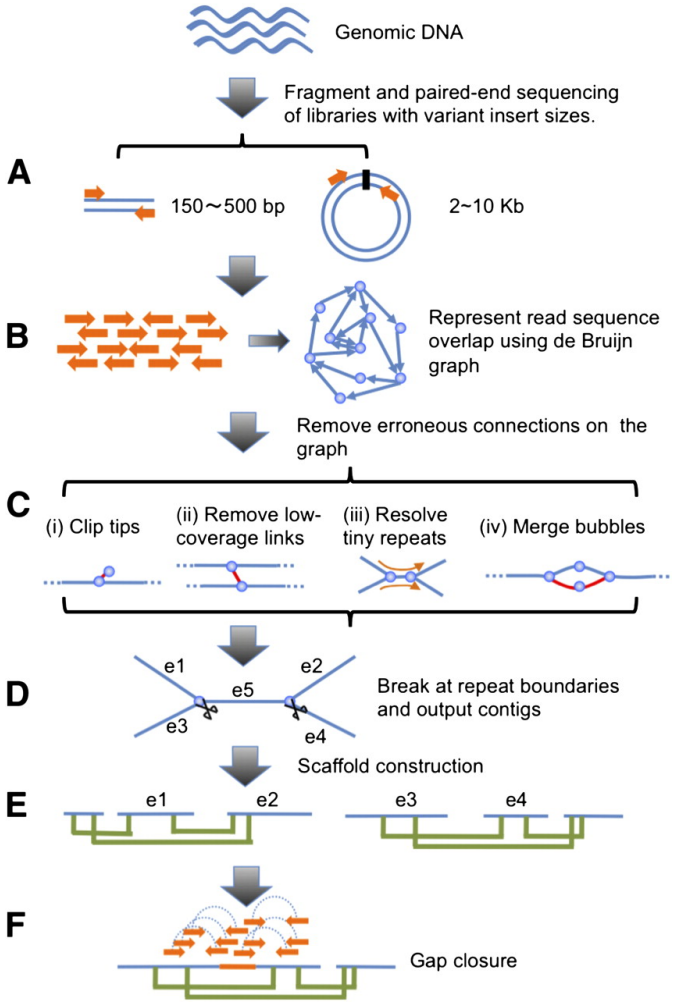
Bubbles



Tour Bus correction (Velvet) : parcours par largeur + nœud visité 2e fois + reculer et identifier ancêtre commun + alignement entre les deux possibiités

Compeau, Pevzner & Tessler *Nat Biotechnol* 29 :987 (2011)

Exemple : SOAPdenovo



Li & al *Genome Res* 20 :265 (2011)

ALLPATHS-LG

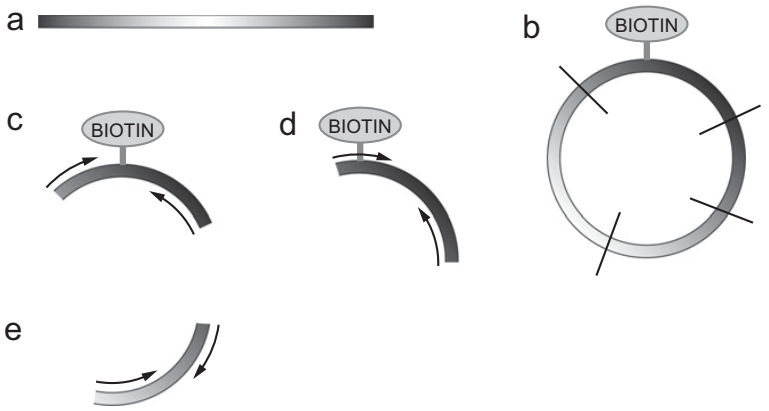
recette de données de séquençage + algorithme adapté

Table 1. Provisional sequencing model for de novo assembly

Libraries, insert types*	Fragment size, bp	Read length, bases	Sequence coverage, ×	Required
Fragment	180†	≥100	45	Yes
Short jump	3,000	≥100 preferable	45	Yes
Long jump	6,000	≥100 preferable	5	No‡
Fosmid jump	40,000	≥26	1	No‡

*Inserts are sequenced from both ends, to provide the specified coverage.
†More generally, the inserts for the fragment libraries should be equal to ~1.8 times the sequencing read length. In this way, the reads from the two ends overlap by ~20% and can be merged to create a single longer read. The current sequencing read length is ~100 bases.
‡Long and Fosmid jumps are a recommended option to create greater continuity.

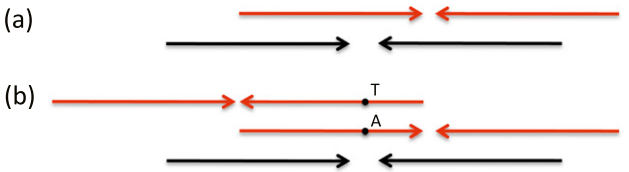
jump libraries et erreurs :



Gnerre & al *Proc Nat Acad Sci USA* 108 :1513 (2011)

ALLPATHS-LG

1. «doublage» avec jumps : de-Bruijn collapsé pour 96-mers



2. correction d'erreurs selon distribution de 24-mers

3. resolution avec appariements de lectures

⇒ 48-core, 512G RAM : 3 semaines de calcul pour génome dun mammifère

(SOAPdenovo : 3 jours)

Gnerre & al *Proc Nat Acad Sci USA* 108 :1513 (2011)

Évaluation

comparaison du génome diploïde avec la séquence assemblée : chemins maximaux (contigs ou échafaudage) avec adjacences consistentes avec un haplotype ou l'autre

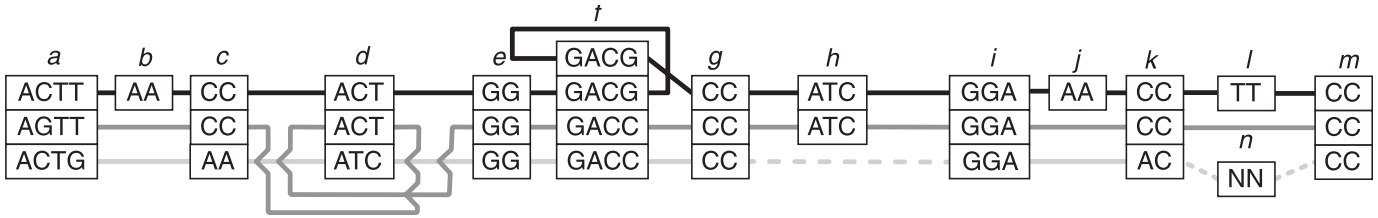


Figure 3. An adjacency graph example demonstrating threads, contig paths, and scaffold paths. Each stack of boxes represents a block edge. The nodes of the graph are represented by the *left* and *right* ends of the stacked boxes. The adjacency edges are groups of lines that connect the ends of the stacked boxes. Threads are represented (*inset*) within the graph as alternating connected boxes and colored lines. There are three threads shown: (*top to bottom*) black, gray, and light gray. The black and gray threads represent two haplotypes; there are many alternative haplotype threads that result from a mixture of these haplotype segments, which are equally plausible given no additional information to deconvolve them. The light-gray thread represents an assembly sequence. For the assembly thread, consistent adjacencies are shown in solid light gray. The dashed light gray line between the *right* end of block *g* and the *left* end of block *i* represents a structural error (deletion). The dashed light-gray line between the *right* end of block *k* and the *left* end of block *m* represents a scaffold gap, because the segment of the assembly in block *n* contains wild-card characters. The example, therefore, contains three contig paths: (from *left* to *right*) blocks *a*. . .*g* ACTGAAATCGGGACCCC; blocks *i, j, k* GGAAC; and block *m* CC. However, the example contains only two scaffold paths because the latter two contig paths are concatenated to form one scaffold path.

Earl & al *Genome Res* 21 :2224 (2011)

Évaluation : contiguité

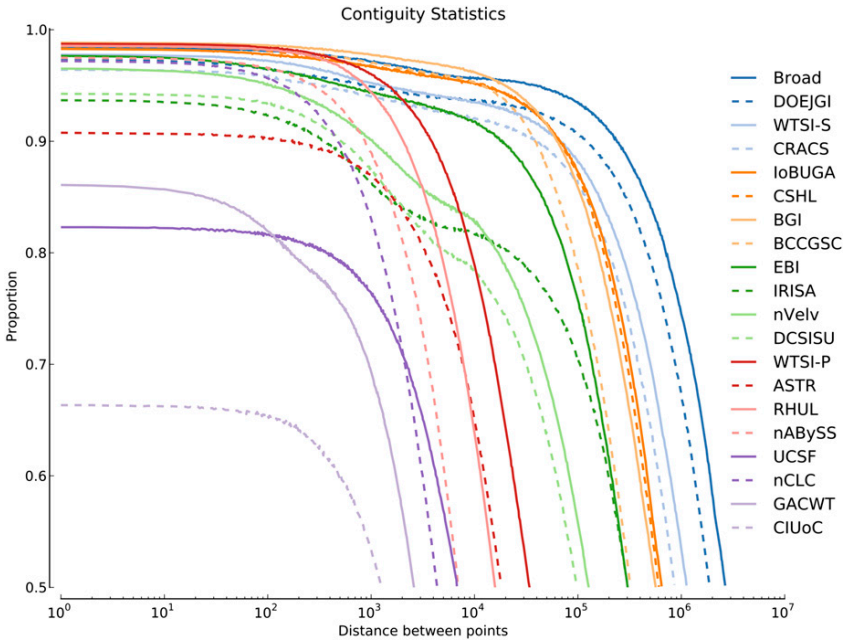


Figure 5. The proportion of correctly contiguous pairs as a function of their separation distance. Each line represents the top assembly from each team. Correctly contiguous 50 (CC50) values are the lowest point of each line. The legend is ordered *top* to *bottom* in descending order of CC50. Proportions were calculated by taking 100,000,000 random samples and binning them into 2000 bins, equally spaced along a $\log_{10}$ scale, so that an approximately equal number of samples fell in each bin.

est-ce que'une paire de positions $i < j$ (tiré au hasard) dans un haplotype se trouvent dans un scaffold aligné correctement ?

Earl & al *Genome Res* 21 :2224 (2011)

Assembleurs de novo

ID	Overall	CPNG50	SPNG50	Struct	CC50
Broad	31	2 (7.25×10^4)	3 (2.11×10^5)	3 (1244)	1 (2.66×10^6)
BGI	37	1 (8.23×10^4)	6 (1.17×10^5)	6 (1878)	7 (5.66×10^5)
WTSI-S	38	9 (2.48×10^4)	1 (4.95×10^5)	2 (475)	3 (1.14×10^6)
DOEJGI	44	14 (1.15×10^4)	2 (4.86×10^5)	1 (456)	2 (1.89×10^6)
CSHL	57	3 (4.23×10^4)	8 (7.17×10^4)	14 (5146)	6 (6.11×10^5)
CRACS	58	11 (1.55×10^4)	5 (1.44×10^5)	4 (1666)	4 (8.61×10^5)
BCCGSC	60	5 (3.63×10^4)	4 (1.46×10^5)	10 (2867)	8 (3.22×10^5)
EBI	64	16 (9.39×10^3)	7 (1.13×10^5)	7 (2055)	9 (3.04×10^5)
IoBUGA	65	7 (3.06×10^4)	12 (3.54×10^4)	15 (6310)	5 (6.47×10^5)
RHUL	71	6 (3.20×10^4)	13 (3.31×10^4)	8 (2551)	15 (1.59×10^4)
WTSI-P	74	4 (3.80×10^4)	11 (4.21×10^4)	13 (4895)	13 (3.41×10^4)
DCSISU	99	12 (1.35×10^4)	10 (5.61×10^4)	12 (4319)	12 (9.75×10^4)
nABySS	100	10 (1.99×10^4)	16 (2.00×10^4)	5 (1731)	16 (6.97×10^3)
IRISA	103	17 (8.20×10^3)	9 (5.82×10^4)	11 (3725)	9 (3.04×10^5)
ASTR	106	8 (2.52×10^4)	14 (3.13×10^4)	9 (2818)	14 (1.81×10^4)
nVelv	114	18 (5.65×10^3)	15 (2.75×10^4)	18 (8626)	11 (1.27×10^5)
nCLC	115	15 (9.47×10^3)	18 (9.54×10^3)	16 (7283)	18 (4.36×10^3)
UCSF	138	12 (1.35×10^4)	17 (1.35×10^4)	20 (24,987)	17 (6.84×10^3)
GACWT	149	20 (2.53×10^3)	19 (7.82×10^3)	17 (8622)	19 (2.60×10^3)
CIUoC	152	19 (5.60×10^3)	20 (5.60×10^3)	19 (11,282)	20 (1.27×10^3)

(CPNG50) contig path NG50 ; (SPNG50) scaffold path NG50 ; (Struct) sum of structural errors ; (CC50) length for which half of any two valid columns in the assembly are correct in order and orientation

(Broad) ALLPATHS-LG, (WTSI-S) SGA, (BGI) SOAPdenovo

Earl & al *Genome Res* 21 :2224 (2011)