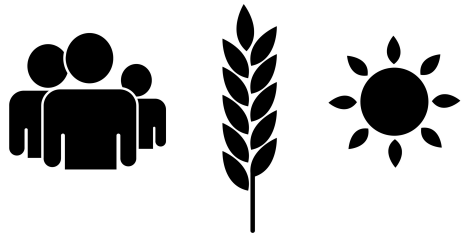


ALIGNEMENT DE GÉNOMES COMPLETS

Hélène Chiapello, INRA – Unité MaIAGE

23 octobre 2017



Hélène Chiapello

Helene.chiapello@inra.fr



Alignement de génomes

1. Introduction

- Contexte
- Principe des outils
- Les outils disponibles

2. Principaux outils

- Mummer
- BlastZ et YASS
- La famille d'outils « MAUVE »
- Mugsy
- ParSNP

3. Les formats d'alignement

4. L'évaluation des outils d'alignement

Partie 1 : Introduction - Contexte

Sequencing Projects*

Complete Projects 11 796

Permanent Drafts 85 019

Incomplete Projects 59 791

Targeted Projects 1 237

Projects with Genbank Data

Seq. Projects 83 053

Archaeal Projects 766

Bacterial Projects 70 071

Eukaryal Projects 4 267

Viral Projects 7 949

Le développement d'outils performants de comparaison de génomes est un enjeu important de la génomique et de la bioinformatique

*source GOLD : <http://www.genomesonline.org>, oct 2017

Pourquoi comparer des génomes ?

Quelques enjeux clés en biologie :

. Pour l'annotation des génomes :

- . Aide à l'**assemblage** et à l'**annotation** de génomes « drafts » ;
- . Analyse et prédiction de certains **éléments fonctionnels** des génomes (motifs, îlots,...) ;

. Pour les analyses de phylogénomique:

- . Analyse de la **microévolution** des génomes ;
- . Définition d'un **génomme minimal ancestral** ;
- . Construction d'**arbres phylogénétiques**.

La comparaison de génomes

Deux grands types de stratégies :

- **La comparaison des protéomes :**

- identification des familles de protéines (orthologues & paralogues)
- annotation fonctionnelle des protéines

- **L'alignement des génomes :**

- alignement global ou local des chromosomes complets ou partiels
- identification des séquences conservées (codantes et non codantes)

➤ Les deux approches sont complémentaires

➤ C'est la distance évolutive qui détermine le choix de la méthode

Frazer et al. 2003. Genome Research. 13:1-12.

Kellis et al. 2004. J Comput Biol. 11:319-55.

L'alignement de génomes chez les procaryotes

Contexte :

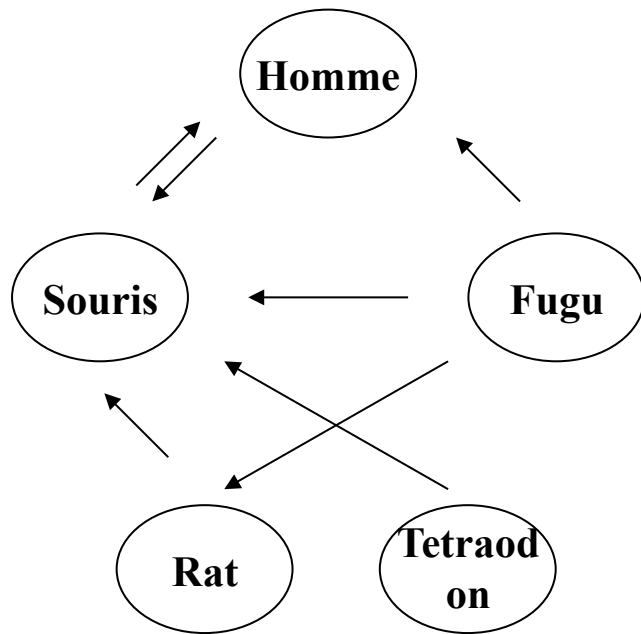
- **Séquences génomiques de grande taille** (en général entre 1 à 6 Mb pour 1 génome bactérien)
- **Grande diversité d'événements mutationnels** : mutations ponctuelles, insertions/délétions, et divers types de réarrangements (inversions, duplications, translocations).

Quelques exemples d'application:

- Analyse de la **structure et de l'évolution** des génomes, en particulier à l'échelle micro-évolutive ;
- Définition d'un **génomme minimal ancestral** ;
- Identification de **l'origine d'un phénotype particulier** (pathogénicité, adaptation,...)
- Prédiction et analyse de la structure de certains **éléments fonctionnels** (motifs, îlots) ;
- Aide à **l'assemblage et à l'annotation** de génomes "drafts".

Et chez les eucaryotes ?

Un objectif central : améliorer l'annotation



Les enjeux clés :

- La détermination de la structure des gènes
- La définition de régions synténiques
- La prédiction des éléments cis-régulateurs

Les spécificités :

- les séquences répétées (50 % du génome humain)
- les grandes duplications de segments chromosomiques (5 % du génome humain)
- les alignements pré-calculés avec des outils spécifiques et un génome de référence

Ureta-Vidal A *et al.* 2003. Nature Reviews. 4:251-262.

Alignement de génomes : les besoins.

Des logiciels « récents », des besoins nombreux :

1. Amélioration des logiciels capables d'aligner deux génomes : **bases statistiques rigoureuses**
2. Mise au point de logiciels capables de réaliser de **l'alignement multi-génomes**
3. Mise au point d'outils de **visualisation des alignements** à l'échelle des génomes
4. Définition de **jeux de données de tests et de validations** des algorithmes

Miller. 2001. *Bioinformatics*. 17:391-397

Du gène/de la protéine au génome :

- **Nombreux outils existants pour aligner deux gènes ou deux protéines :**

Alignement par programmation dynamique
(*Needleman & Wunsch, 1970*), **FASTA** (*Pearson & Lipman, 1988*), **BLAST** (*Altschul & al. 1990*), etc...

- **Ces outils ne fonctionnent pas ou mal à l'échelle d'un génome**
 - Les temps de calcul sont prohibitifs
 - Les alignements de génomes complets nécessitent des outils dédiés qui tiennent compte de certaines spécificités : les répétitions, les grandes insertions/délétions, les réarrangements.

Les principes de base

- **Concerne des génomes “proches” (sinon approches protéiques)**
- **Changement d'échelle**
 - Techniques classiques d'alignement de protéines pas applicables
 - Utilisation des techniques issues de l'algorithmique du texte
 - Principe de l'ancrage : détection de matchs exacts ou inexacts.
- **Différents types d'outils**
 - Alignement global ou local.
 - Alignement 2 à 2 ou multiple.
 - Génomes réarrangés ou non.
 - Génome de référence ou non

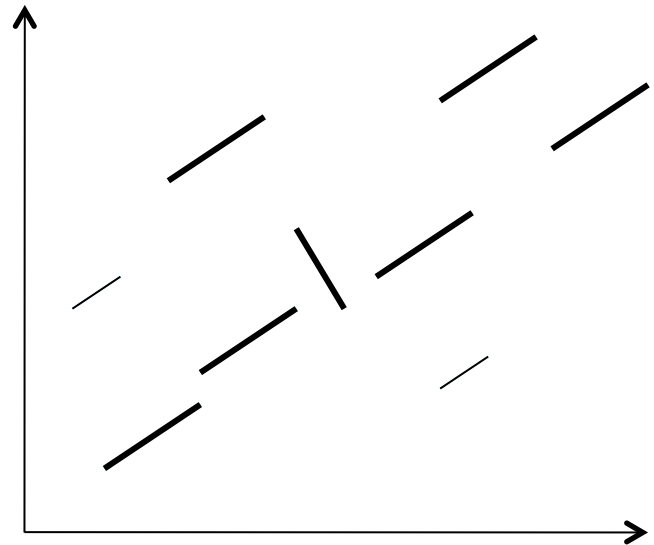
Alignement local de génomes

But : trouver les régions similaires dans deux séquences
quels que soient l'ordre et la position de ces similarités

- Permet d'identifier des réarrangements locaux, des duplications
- Permet d'aligner des génomes en cours de séquençage

En général 3 étapes :

1. Détection de 'matches' (graines)
2. "Clustering" des graines voisines
3. Extension sans et avec gap.



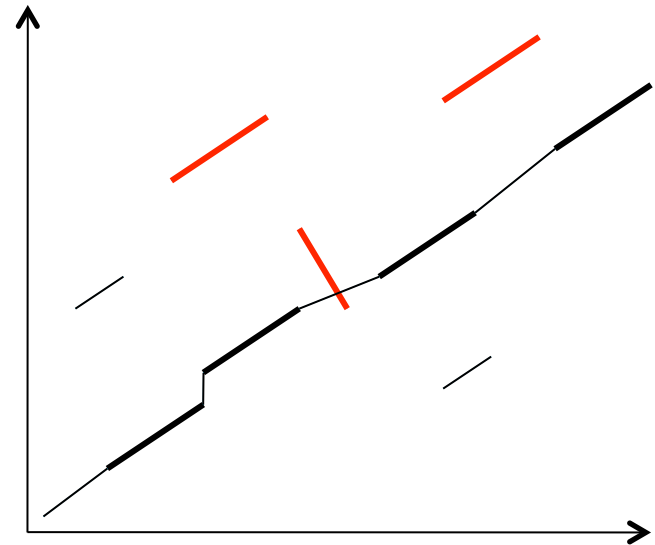
Alignement global de génomes

But : aligner au mieux les génomes sur la totalité de leur longueur

- Certains algorithmes détectent les réarrangements
- En général, pas adapté aux génomes en cours de séquençage

En général trois étapes

1. Détection de matches
- 2. Ancrage et chainage des matches**
3. Traitement des régions entre les matches



Outils d'alignement de deux génomes

Quelques exemple d'outils

Logiciel	Référence	Principe	URL + Distribution
Mummer 1	<i>Delcher et al., 1999</i>	Alignement global par recherche de MUMs (Maximal Unique Matches).	http://www.tigr.org/software/mummer/ Programme + serveur Web
Blastz/ PipMaker	<i>Schwartz et al., 2000</i>	Alignement local par une adaptation de Gapped BLAST.	http://bio.cse.psu.edu/pipmaker Seveur Web
Avid	<i>Bray et al. 2003</i>	Alignement global par programmation dynamique.	http://glass.lcs.mit.edu/ Programme
YASS	<i>Noé. et al, 2005</i>	Alignement local basé sur des graines inexactes	http://loria.fr/projects/YASS Programme + serveur Web
Mauve Contig Mover	<i>Rissman et al., 2009</i>	Alignement global d'un génome draft sur un génome de référence	http://gel.ahabs.wisc.edu/mauve Programme java
YOC	<i>Uricaru et al., 2015</i>	Alignement global à partir du chainage d'alignements locaux YASS	https://github.com/ruricaru/YOC Programme

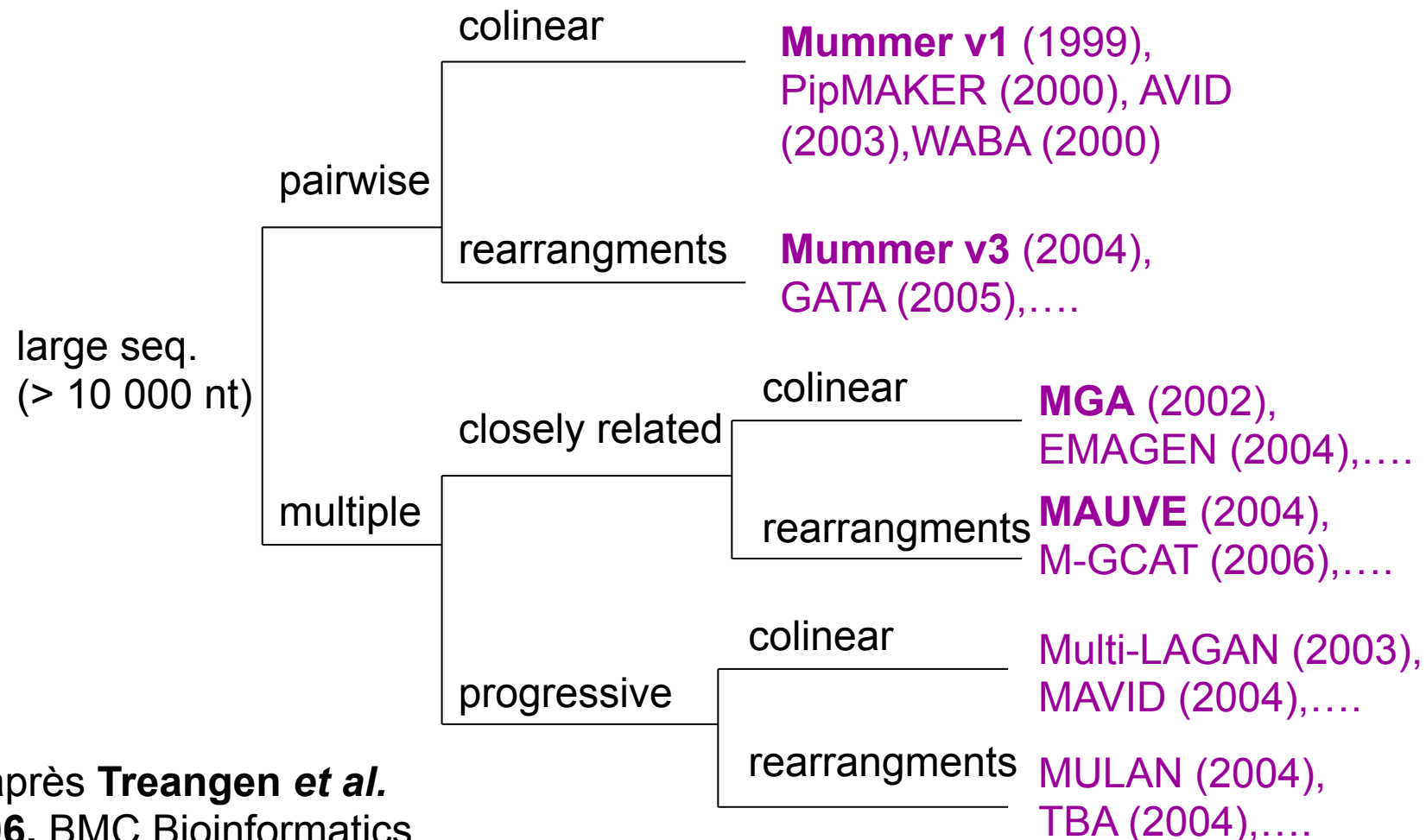
Outils d'alignement multi-génomés

Quelques exemple d'outils

Logiciel	Référence	Principe	URL + distribution
Mummer 2 et 3	<i>Delcher et al., 2002, Kurtz et al., 2004</i>	Alignement global par recherche de MUMs (Maximal Unique Matches)	http://www.tigr.org/software/mummer/ Programme + serveur Web
MGA	<i>Höhl et al., 2002</i>	Alignement global par recherche de MEMs (Maximal Exact Matches)	http://bibiserv.techfak.unibielefeld.de/mga/ Programme
LAGAN/multiLAGAN/ ShuffleLAGAN	<i>Brudno et al., 2003</i>	Alignement global par recherche récursive de matchs dégénérés	http://lagan.stanford.edu/lagan_web/index.shtml Programme + serveur Web
MAUVE	<i>Darling et al. 2004</i>	Alignement global par recherche de MUMs et LCBs	http://gel.ahabs.wisc.edu/mauve Programme java
ProgressiveMAUVE	<i>Darling et al., 2010</i>	Alignement global par recherche de matchs	http://gel.ahabs.wisc.edu/mauve Programme java
MUGSY	<i>Angiuoli et al., 2011</i>	Alignement global basé sur alignements locaux Mummer	http://mugsy.sf.net Programme
ParSNP	<i>Treangen et al., 2014</i>	Alignement global basé sur des multiMUMs et LCBs	http://harvest.readthedocs.io/en/latest/content/parsnp.html Programme

Classification des outils d'alignement de génomes*

Comment distinguer les outils ?



*d'après **Treangen *et al.* 2006**. BMC Bioinformatics.

Mummer

- **Alignement d'un génome de référence avec n génomes proches**
- **Principe de l'ancrage** : détection de MUMs (Maximal Unique Matches) ou MEMs (Maximal Exact Matches)
- Utilisation de technique issues de **l'algorithmique du texte** (arbre des suffixes)
- **Outil modulaire & Open Source** (3 versions) avec outils de visualisation graphique

Delcher & al. 1999, 2002, Kurtz & al. 2004
<http://www.tigr.org/software/mummer>

Les programmes de Mummer

<http://www.tigr.org/software/mummer/manual>

Mummer est une suite logicielle avec différents outils de :

- **Maximal Exact Matching :**
`mummer`, `repeat-match`, `exact-tandems`
- **Clustering :**
`gaps` (sans réarrangement), `mgaps` (avec réarrangement)
- **Alignment generators (pairwise):**
`NUCmer` (mummer+mgaps), `PROmer` (en protéine),
`run-mummer1` (sans réarrangement), `run-mummer3`
(avec réarrangement)
- **Utilities :**
`MapView`, `mummerplot`, `show-aligns`, `show-coords` ,
`show-tiling`.

Quatre grandes étapes :

- 1. Recherche des MUMs** : Maximal Uniques Matches
- 2. Tri des MUMs** : recherche du plus long ensemble de MUMs dans le même ordre sur les 2 génomes
- 3. Fermeture des gaps** en effectuant une recherche locale des grandes insertions, répétitions, régions divergentes, répétitions en tandem et SNPs.
- 4. Sortie de l'alignement** incluant les MUMs et les alignements des régions dans les gaps.

Delcher *et al.* 1999. NAR.

Etape 1 : la recherche des MUMs

Mummer

Un MUM (Maximal Unique Match) :

- Est présent exactement une fois sur les génomes A et B
- N'est pas inclus dans un MUM plus long

```
Genome A: tcgatcGACGATCGCGCCGTAGATCGAATAACGAGAGAGCATAAcgactta  
Genome B: gcattaGACGATCGCGCCGTAGATCGAATAACGAGAGAGCATAAtccagag
```

mismatches

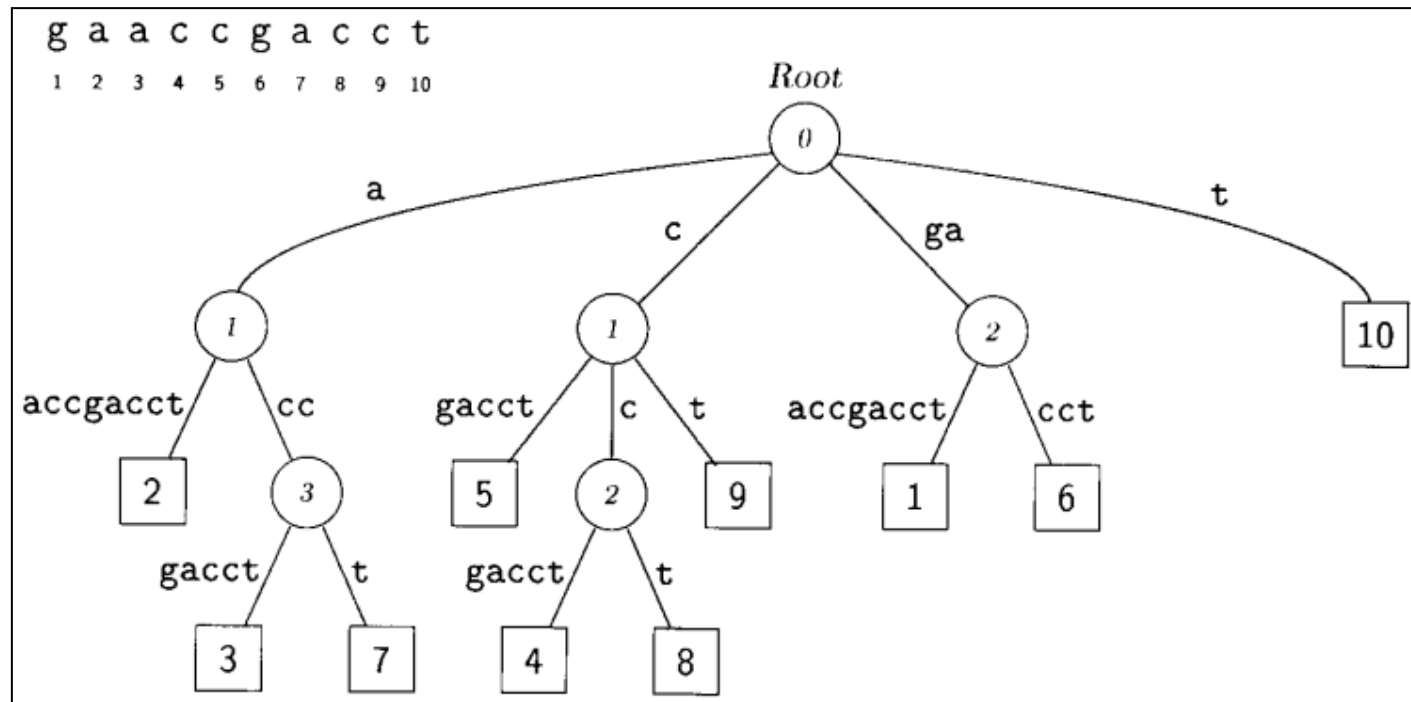
Un MUM « suffisamment long » est forcément inclus dans l'alignement global des génomes.

Figure extraite de Delcher *et al.* 1999. NAR.

L'arbre des suffixes

Mummer

Principe : les MUMs sont identifiés en construisant **un arbre des suffixes** des génomes A et B.



- **Chaque nœud interne (circulaire)** représente une séquence répétée (label=long. seq.)
- **Chaque feuille (carré)** représente un suffixe complet (label=position)

Figure extraite de Delcher *et al.* 1999. NAR.

L'arbre des suffixes et les MUMs

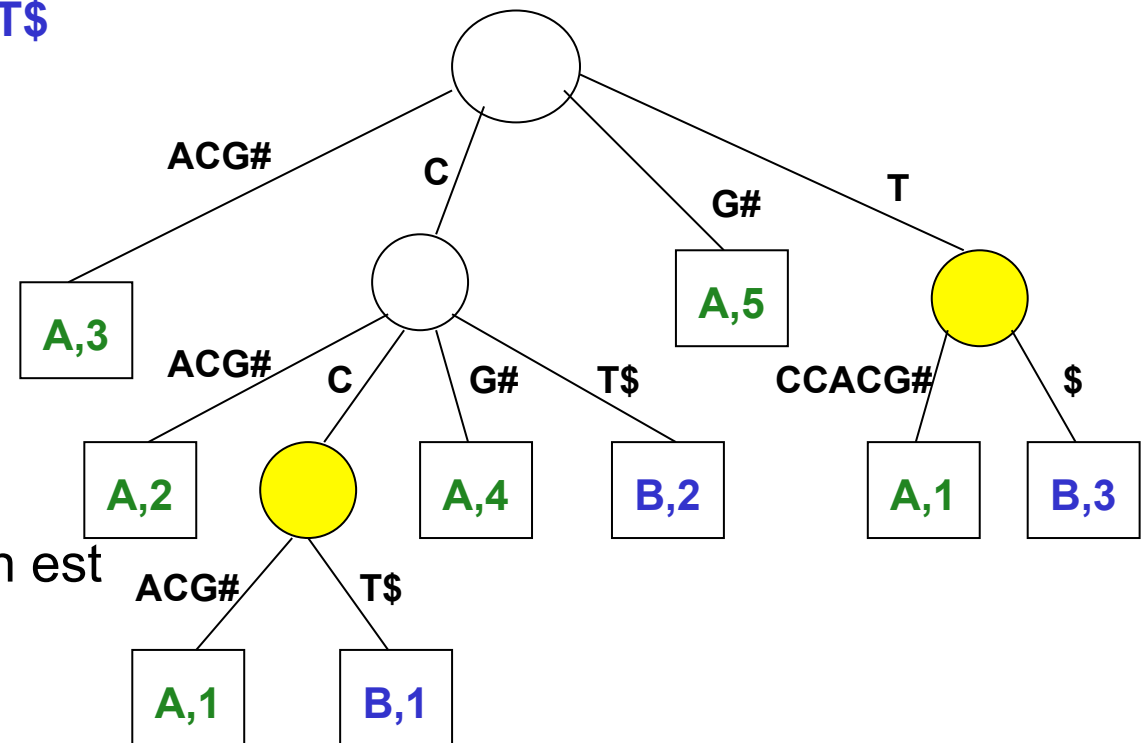
Mummer

Sequence A = TCCACG#

Sequence B = CCT\$

- **Un Match Unique** est sur un nœud interne avec deux feuilles de chaque génome A et B

- **Un Match Unique est Maximal (MUM)** si le caractère avant le match est différent sur chaque génome.



Mummer

Des propriétés intéressantes :

- Les arbres des suffixes peuvent être construits en un temps linéaire de la longueur des génomes
- Ils permettent d'identifier des MUMs en un temps linéaire de la taille des MUMs (une lecture de l'arbre)
- Le principal paramètre à choisir est la longueur minimale des MUMs à détecter (20 à 50 pb).

Etape 2: le tri des MUMs

Mummer

- **Tri des MUMs** en fonction de leur position sur le génome A
- Recherche du plus long ensemble de MUMs ordonné sur les génomes A et B (**algorithme LIS** : Longest Increasing Subsequence, modifié pour tenir compte de la longueur des MUMs et du chevauchement).

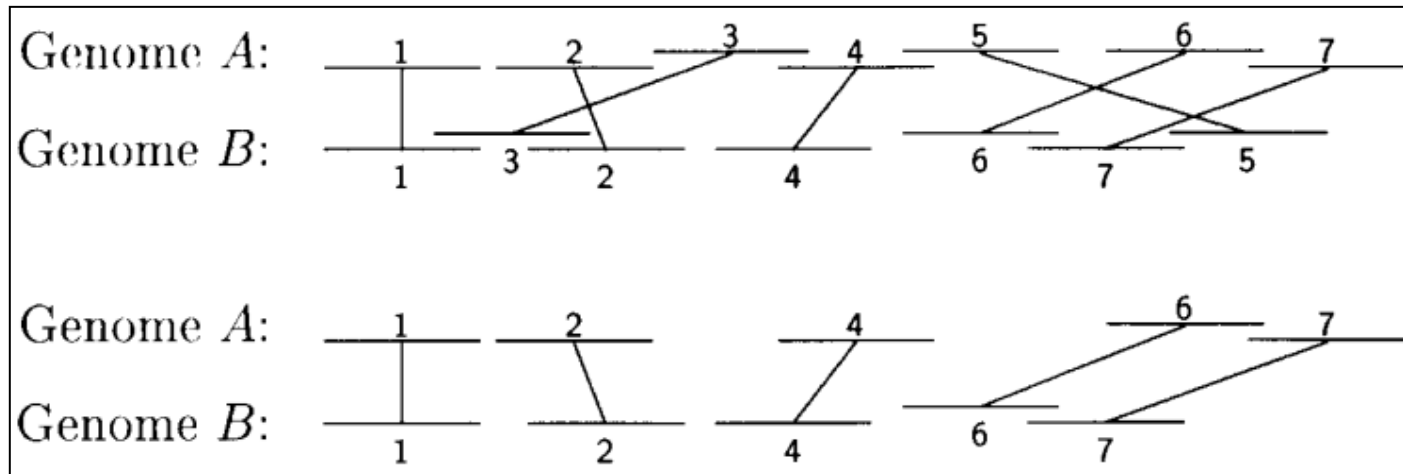


Figure extraite de Delcher *et al.* 1999. NAR.

Etape 3 : fermeture des gaps

Mummer

Les quatre types de gaps après la recherche des MUMs

1. SNP: exactly one base (indicated by ^) differs between the two sequences. It is surrounded by exact-match sequence.

```
Genome A: cgtcatgggcgttcgtcgttg
Genome B: cgtcatgggcattcgtcgttg
                ^
```

2. Insertion: a sequence that occurs in one genome but not the other.

```
Genome A: cggggtaaccgc.....cctggtcggg
Genome B: cggggtaaccgcgttgctcggggtaaccgccctggtcggg
                ^^^^^^^^^^^^^^^^^
```

3. Highly polymorphic region: many mutations in a short region.

```
Genome A: ccgcctcgccctgg.gctggcgcccgcttc
Genome B: ccgcctcgcccagttgaccgcgcccgcttc
                ^ ^ ^ ^
```

4. Repeat sequence: the repeat is shown in uppercase. Note that the first copy of the repeat in Genome B is imperfect, containing one mismatch to the other three identical copies.

```
Genome A: cTGGGTGGGACAACGTaaaaaaaTGGGTGGGACAACGTc
Genome B: aTGGGTGGGGCgACGTgggggggggTGGGTGGGACAACGTa
                ^             ^             ^
```

Figure extraite de Delcher *et al.* 1999. NAR.

Etape 3 : fermeture des gaps

Mummer

SNPs

- soit entre deux MUMs (simple)
- soit traités comme des répétitions

Insertions

- soit des transpositions
 - soit des vraies insertions
- => elles n'apparaissent pas dans l'alignement

Régions

polymorphiques

- si courtes : alignées par programmation dynamique
- si longues : recherche récursive de MUMs de taille inférieure

Répétitions

- détectées par MUMs chevauchants
- => elles n'apparaissent pas dans l'alignement (choix de l'occurrence conservée en position dans les deux génomes)

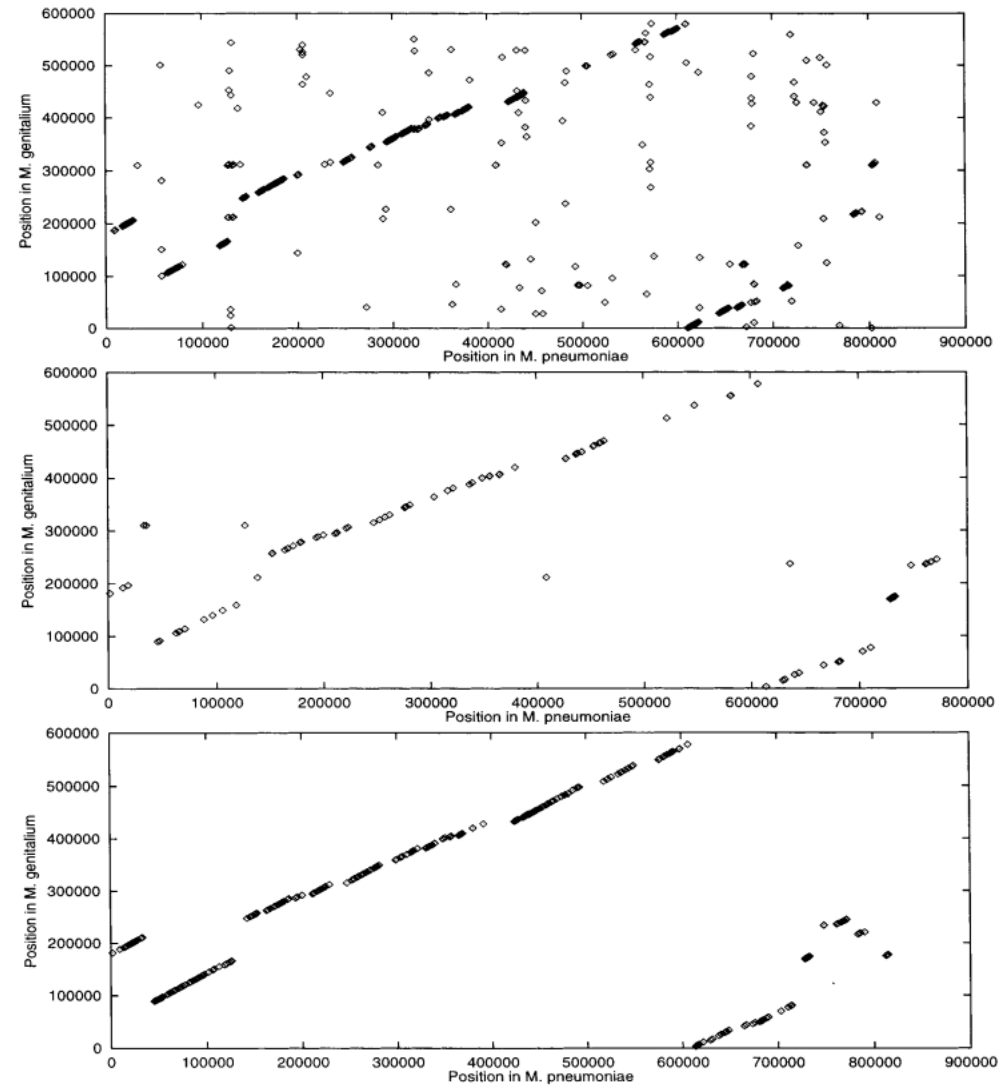
Exemple de résultat

Mummer

- **FASTA :**
Segments 1000pb > 50 % id.
sur 80 % de longueur.

- **25-mers** uniques
(MUMs de taille fixe)

- **MUMs de taille min 15 pb**
alignés avec Mummer



Mycoplasma genitalium X
Mycoplasma pneumoniae

Figure extraite de Delcher *et al.* 1999. NAR.

BlastZ

Principe

- Calcule des **alignements locaux** entre 2 génomes
- Utilisé et validé pour l'alignement des génomes homme/souris
- Variante du programme GappedBLAST (*Altschul et al., 1997*) qui comporte 3 étapes :
 - Recherche des matchs exacts (ou quasi exacts)
 - Extension des matchs sans permettre de gaps
 - Alignement final étendu par programmation dynamique avec gaps permis

Schwartz et al., 2000 & 2003

<http://bio.cse.psu.edu/>

BlastZ

Les 5 étapes principales

1. **On élimine les séquences répétées** spécifiques à chaque espèce
2. **Pour toutes les paires de 12-mers identiques** (au max à 1 transition près)
 - i. **Extension de l'alignement** dans les 2 directions sans gap jusqu'à score chute en dessous d'un seuil S (par exemple 3000).
 - ii. **Si score alignement > 3000**
 - Répéter l'étape d'extensions (i) mais cette fois-ci avec gaps
 - Retenir l'alignement si score >5000
3. **Répéter l'étape 2 avec un ancrage plus sensible** (par exemple 7-mers identiques) et un score plus bas pour les 2 types d'alignements (par ex 2000)
4. **Ajustement des positions des alignements** en fonction de l'étape 1
5. **Filtrage des alignements** en fonction de l'objectif

Quelques particularités

BlastZ

• Algorithme

- Il existe une option pour **détecter les régions identiques dans le même ordre et la même orientation** sur les deux génomes
- Utilisation d'un **score d'alignement adapté aux génomes complets** (tient compte de la complexité de la séquence)

• Performances

- **Masquage automatique de régions répétées** sur le premier génome
- **Passage des 8-mer aux 19-mers** dont 12 positions sont identiques à une transition près.

BlastZ

Alignement génomes Souris/Homme

- **Une grande spécificité. Par exemple, si score seuil=3000 :**
 - 39 % génome H s'aligne avec génome S
 - 0,164 % génome H s'aligne avec génome S rev/comp
- **Validation biologique : analyse détaillée du K20 H (homologue au K2 S)**
 - 40,5 % du K20 de H est aligné avec K2 S dont
 - 98,5 % des CDS
 - 87,1 % des 3'UTR
 - 89,0 % des 5'UTR

Schwartz et al., 2003

MultiZ

- Généralisation de BlastZ
- L'algorithme **TBA** (pour "threaded blockset aligner") construit un alignement multiple en se basant sur des "blocs" présents dans le même ordre et la même orientation dans les séquences comparées
- Ne tient pas compte des inversions ou duplications mais peut détecter des matchs communs à un sous-ensemble de séquences.
- Permet de construire un alignement multiple global.

Blanchette et al., Gen. Res. 2004

Alignement de deux génomes

YASS

Le principe :

- Calcule des **alignements locaux** entre 2 génomes
- **Principe** : comme tous les outils d'alignement nucléique local basé sur une **heuristique** (BLAST, FASTA, PATTERNHUNTER, BLASTZ ...), YASS procède en 2 étapes :
 1. recherche utilise des graines pour détecter des régions potentiellement similaires
 2. extension des régions contenant des graines pour générer des régions alignées avec insertions/délétions
- **Une interface Web** permet de lancer YASS et de générer différents types de sortie: fichiers textes dans différents formats, fichiers graphiques (**dotplot**)

Noé et Krucherov, 2005

<http://bioinfo.lifl.fr/yass/>

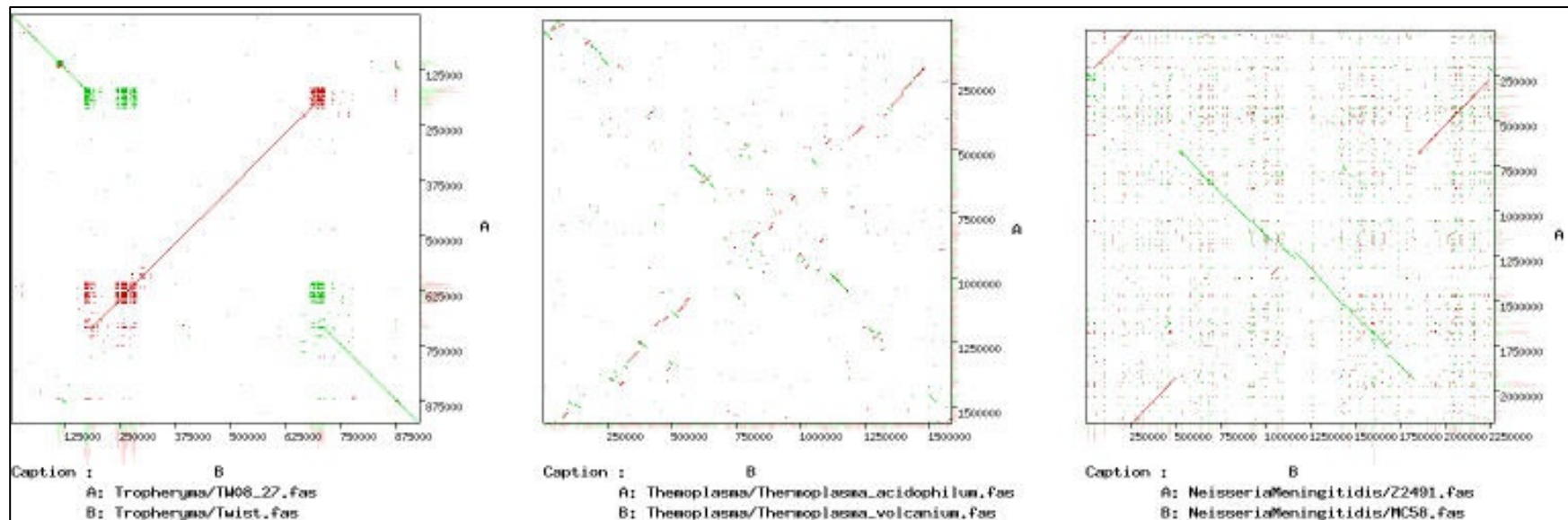
YASS

- Utilise une **famille de graines**, possiblement chevauchantes et un critère de nouveau hit assurant un bon ratio sensibilité/sélectivité.
- **Les graines espacées** utilisées tiennent compte des transitions et améliorent la sensibilité (les transitions sont des mutations purine/ purine [A<->G] or pyrimidine/pyrimidine [C<->T])
- Utilise différents **scores et une « e-value »** pour évaluer la pertinence des alignements
- Un **filtre paramétrable** existe pour les répétitions de faible complexité.
- Plusieurs **paramètres statistiques** d'alignement produits (biais de mutations des triplets, rapport transition/transversion,..)
- Étape de **post-traitement** pour grouper les alignements avec des gaps.

<http://bioinfo.lifl.fr/yass/>

YASS

- **En entrée** : 2 fichiers fasta ou multifasta.
- **En sortie** : Fichiers textes ou dotplot / conversion au format alignement blast.
- **Paramètres principaux** : -r (indique si séquence en réverse ou non), -d (format de sortie), -o (fichier de sortie), -S (si 1^{ère} sequence en multifasta).



Exemples de dotplots générés avec YASS

MAUVE 1 et 2

Principe :

- **Alignement multiple de génomes : un des premiers algorithmes qui traite les réarrangements (avec ShuffleLAGAN)**
- **Principe de l'ancrage** : initialement avec des MUMs (Maximal Unique Matches) puis avec des graines inexactes
- **Identification de blocs colinéaires = LCBs (Locally Colinear Blocks)**
- **Interface graphique + ligne de commande**

Darling A. *et al*, *Genome Research* 2004
<http://gel.ahabs.wisc.edu/mauve>

MAUVE 1 et 2

Quatre étapes principales

1. **Recherche des alignements locaux (multiMUMs)**
2. **Calcul d'un arbre phylogénétique** basé sur les multiMUMs
3. **Ancrage :**
 - **identification des LCBs (Locally Colinear Blocks)** en utilisant les multiMUMs présents sur tous les génomes
 - **Ancrage récursif** pour identifier les ancres supplémentaires entre les LCBs et à l'intérieur des LCBs
4. **Alignement multiple progressif (clustalw) des régions entre les ancres** en utilisant l'arbre de l'étape 2.

1. Recherche des multiMUMs

MAUVE 1 et 2

- **Méthode de hachage (« seed and extend algorithm »)**
 - construction d'une liste triée de k-mers pour chaque génome,
 - identification des « seed » = matchs présents dans au moins 2 génomes en 1 occurrence,
 - extension des matchs jusqu'à ce qu'un mismatch apparaisse dans un des génomes.
- **Détecte les multiMUMs de taille min. k :**
 - directs et reverse/complémentaires
 - communs à tous les génomes
 - communs à un sous-ensemble de génomes
- **Complexité en temps** : $O(G^2n + Gn \log Gn)$, G =nombre de génomes, n =longueur du plus grand génome.

2. Calcul d'un arbre phylogénétique

MAUVE 1 et 2

- **Technique du Neighbor-joining** (Saitou and Nei, 1987) appliquée aux multiMUMs communs à un sous-ensemble de génomes.

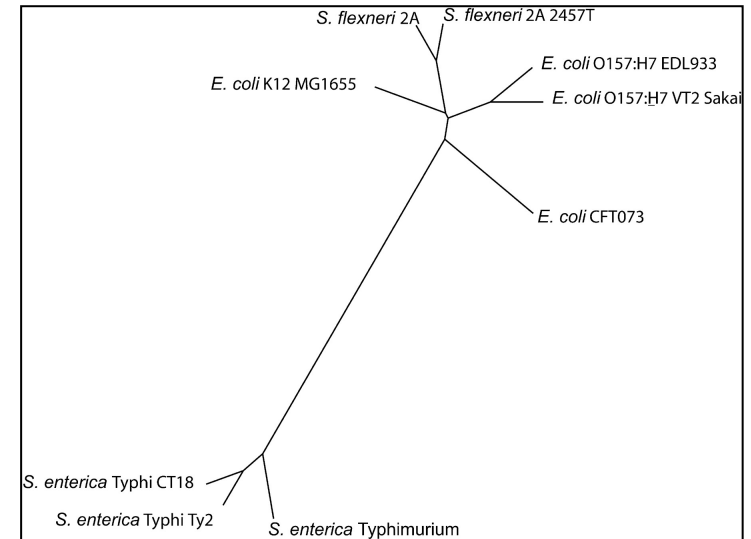
- **Distance calculée entre tous les couples de génomes G_1, G_2 :**

$$d(G_1, G_2) = 1 - \frac{\sum \text{long}(\text{multiMUMsNonChevauchants}G_1G_2)}{\text{LongMoyGénome}G_1G_2}$$

- **Le problème des matches chevauchants est résolu en gardant ceux de multiplicité maximale.**

- **Résultat : arbre guide non enraciné.**

Figure extraite de Darling *et al.* 2004. Gen. Res.



3. Ancrage : identification des LCBs.

MAUVE 1 et 2

- **Recherche des LCB = Locally Colinear Block** : ensemble de multiMUMs colinéaires sur tous les génomes comparés.
- **Algorithme utilisé pour passer des multiMUMs aux LCBs (partitionnement minimum) : « breakpoint elimination »**
 - **basé sur un poids affecté à chaque LCB** (somme des tailles des multiMUMs inclus dans chaque LCB)
 - **principe** : en supprimant des LCBs de faible poids, on supprime des points de rupture pour obtenir une **partition minimale**
 - **par défaut, le poids minimum d'un LCB est de $3k$** , où k est la taille initiale des graines.

Les étapes en images

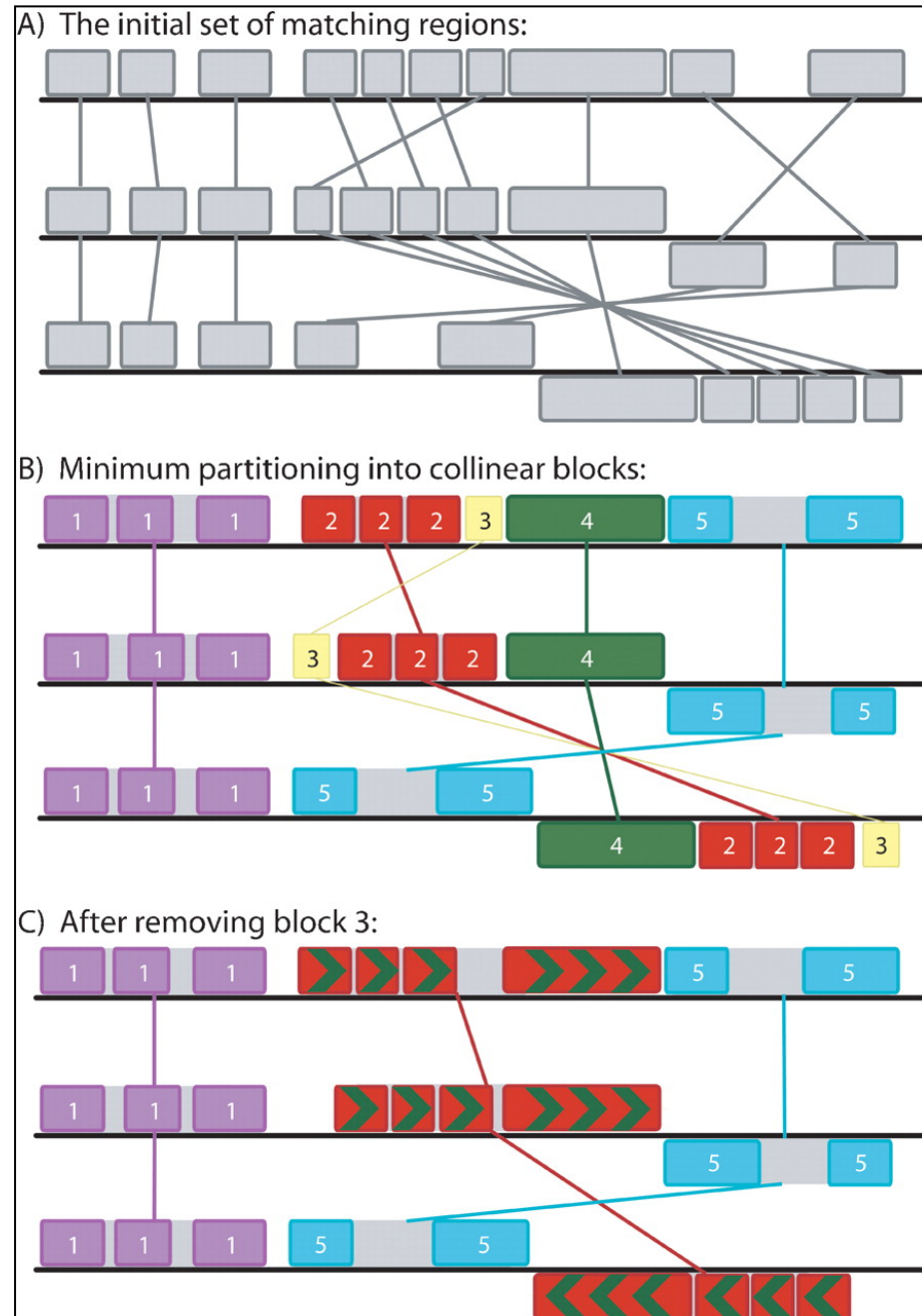
MAUVE 1 et 2

1. Détection des multiMUMS

2. Partitionnement en LCBs

3. Elimination des points de rupture (LCBs de faible poids)

Figure extraite de Darling *et al.* 2004.
Gen. Res.



3. Ancrage récursif

MAUVE 1 et 2

- **Augmente la sensibilité** (cas des régions polymorphes, des répétitions).
- **Deux types recherches récursives traitent :**
 - **les régions entre les LCBs** : même algorithme avec une valeur de k plus faible et en levant le critère d'unicité sur l'ensemble du génome (unicité uniquement dans l'intervalle de recherche)

$$k = seed_size(S) - 2 \quad seed_size(S) = \log_2\left(\sum_{j=1}^G \frac{length(S_j)}{G}\right)$$

- **les régions entre les multiMUMs d'un LCB** : calcul du paramètre k en fonction de la longueur de la séquence S du gap à traiter, arrêt de la recherche si plus d'ancre nouvelle ou séquence restante < 200pb.

$$k = seed_size(S)$$

4. Alignement multiple progressif

MAUVE 1 et 2

Algorithme : clustalw (Thompson et al., 1994)

- Programmation dynamique progressive
- Utilise l'arbre phylogénétique calculé à l'étape 1. Chaque étape aligne deux alignements ou séquences en suivant l'ordre de branchement dans l'arbre.
- Permet d'aligner les régions entre les ancres (si < 10 kb)

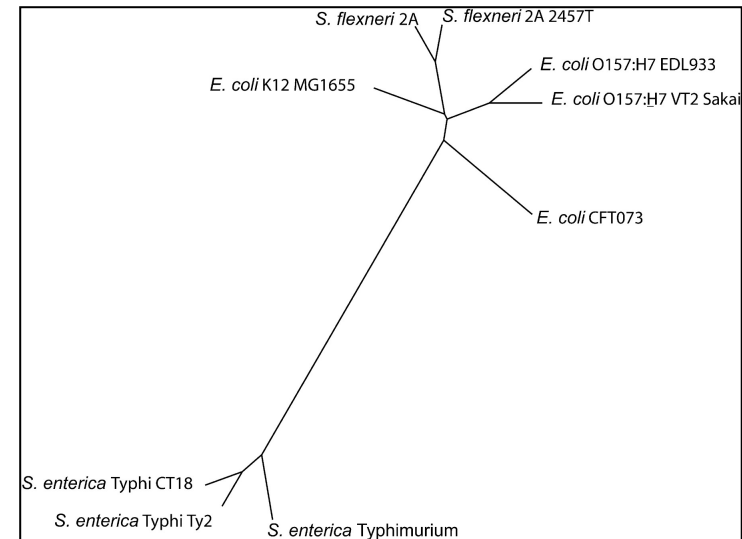


Figure extraite de Darling *et al.* 2004. Gen. Res.

Un exemple de résultat

MAUVE 1 et 2

- 9 génomes d'entérobactéries

- environ 3 heures de calcul (PC, 2.4 GHz, 1 Go de mémoire)

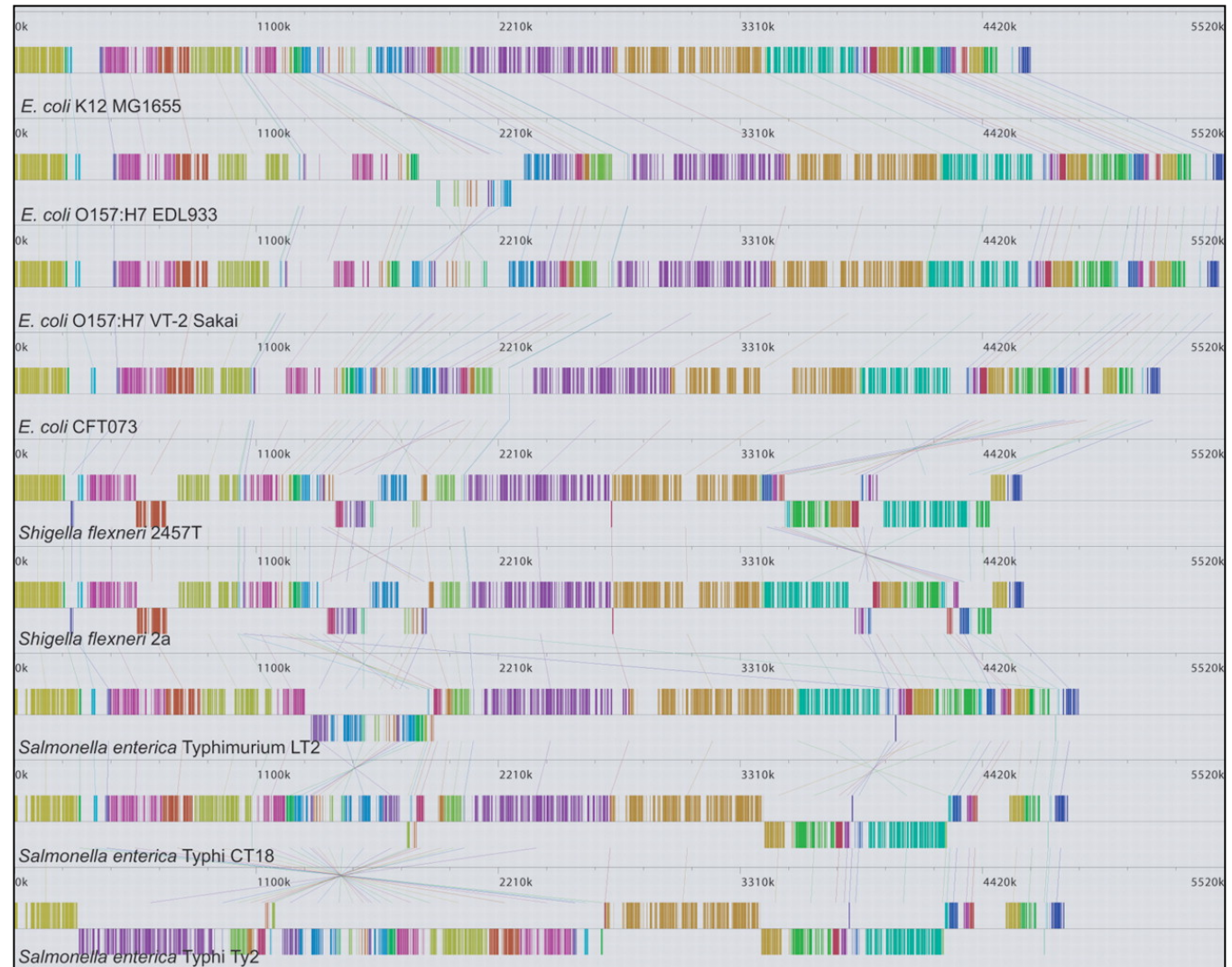


Figure extraite de Darling *et al.* 2004. Gen. Res.

MAUVE 1 et 2

Alignement des 9 génomes d'entérobactéries :

- **45 LCBs de poids minimum 69**
- **définition d'un squelette :**
 - régions de l'alignement sans gap de plus de 50 nt sur aucun génome
 - taille = 2.86 Mb (1252 segments)
soit 58% de couverture moyenne

Mauve : les limitations des premières versions

MAUVE 1 et 2

- **Mauve cible les comparaisons de génomes proches** : les souches d'une même espèce ou organismes très proches (comme *E. coli* et *Salmonella* ou *Y. pestis*).
- **Le poids minimum des LCB** est un paramètre déterminant qui doit être souvent réglé manuellement pour une estimation correcte des réarrangements génomiques.
- Mauve n'est pas capable d'aligner des génomes contenant **de grandes quantités de duplications.**
- **Les alignements produits par Mauve contiennent des "spurious matches" et des alignements erronés**
- **Les grandes régions chromosomiques partagées uniquement par un sous-groupe d'organismes ne sont pas alignées**

Alignement multiple : les outils Mauve

ProgressiveMAUVE

Le principe

- **Nouvel algorithme d'alignement permettant d'aligner les régions conservées uniquement dans un sous groupe de séquences**
- **Trois innovations principales dans l'algorithme :**
 - (1) Un nouveau score d'alignement a été créé => ceci permet la détection précise des positions de réarrangements quand les génomes ont un contenu en gènes très différents
 - (2) Une heuristique « glouton » permet d'optimiser l'ensemble d'ancres choisi en se basant sur le score calculé en (1)
 - (3) Un modèle de Markov caché (HMM) permet d'éliminer les alignements erronés du résultat final (méthode de Treangen *et al.*, 2008).
- **Méthode plus robuste pour traiter les génomes plus divergents ou ayant subi de multiples réarrangements, insertions/délétions et répétitions**

Darling A. *et al*, *PLoSone* 2010
<http://gel.ahabs.wisc.edu/mauve>

Alignement multiple : les outils Mauve

ProgressiveMAUVE

Avantages

- Peut être appliqué à un plus grand nombre de génomes que Mauve
- Peut aligner des génomes plus divergents (en principe jusqu'à 50 % d'identité)
- En général ne nécessite pas d'ajustement de paramètres.
- Permet de générer le *pan-génome*, e.g. les régions conservées entre un sous-ensemble de génomes
- Est plus précis que l'algorithme Mauve

Limitations

- Plus lent que l'algorithme initial de Mauve
- Consomme plus de mémoire que l'algorithme initial de Mauve

Darling A. *et al*, *PLoSone* 2010
<http://gel.ahabs.wisc.edu/mauve>

Alignement multiple : les outils Mauve

Mauve Contig Mover (MCM)

Le principe

- **MCM est une adaptation de ProgressiveMauve pour les génomes drafts constitués de contigs (454, Solexa,...)**
- **Principe** : utiliser un génome de référence (organisation la plus proche possible) pour ordonner les contigs en utilisant un algorithme itératif basé sur des alignements Mauve de contigs
 - (1) **Processus itératif** : les contigs sont alignés itérativement sur le génome de référence en minimisant le nombre total de blocs colinéaires (LCBs)
 - (2) MCM s'arrête quand l'ordre des contigs entre 2 itérations ne varie plus.
- **Résultat** : un fichier fasta des contigs réordonnés et réorientés par rapport au génome de référence et **une liste des contigs** (i) ordonnés sur la référence et (ii) contenant des LCBs (« Conflicting Order Information » i.e. réarrangement, mauvais assemblage,...)

Rissman. *et al*, *Bioinformatics* 2009

<http://gel.ahabs.wisc.edu/mauve>

Mugsy

Les caractéristiques

- Alignement multiple sans génome de référence
- Les génomes peuvent être drafts ou finis
- Peut détecter les réarrangements et les duplications

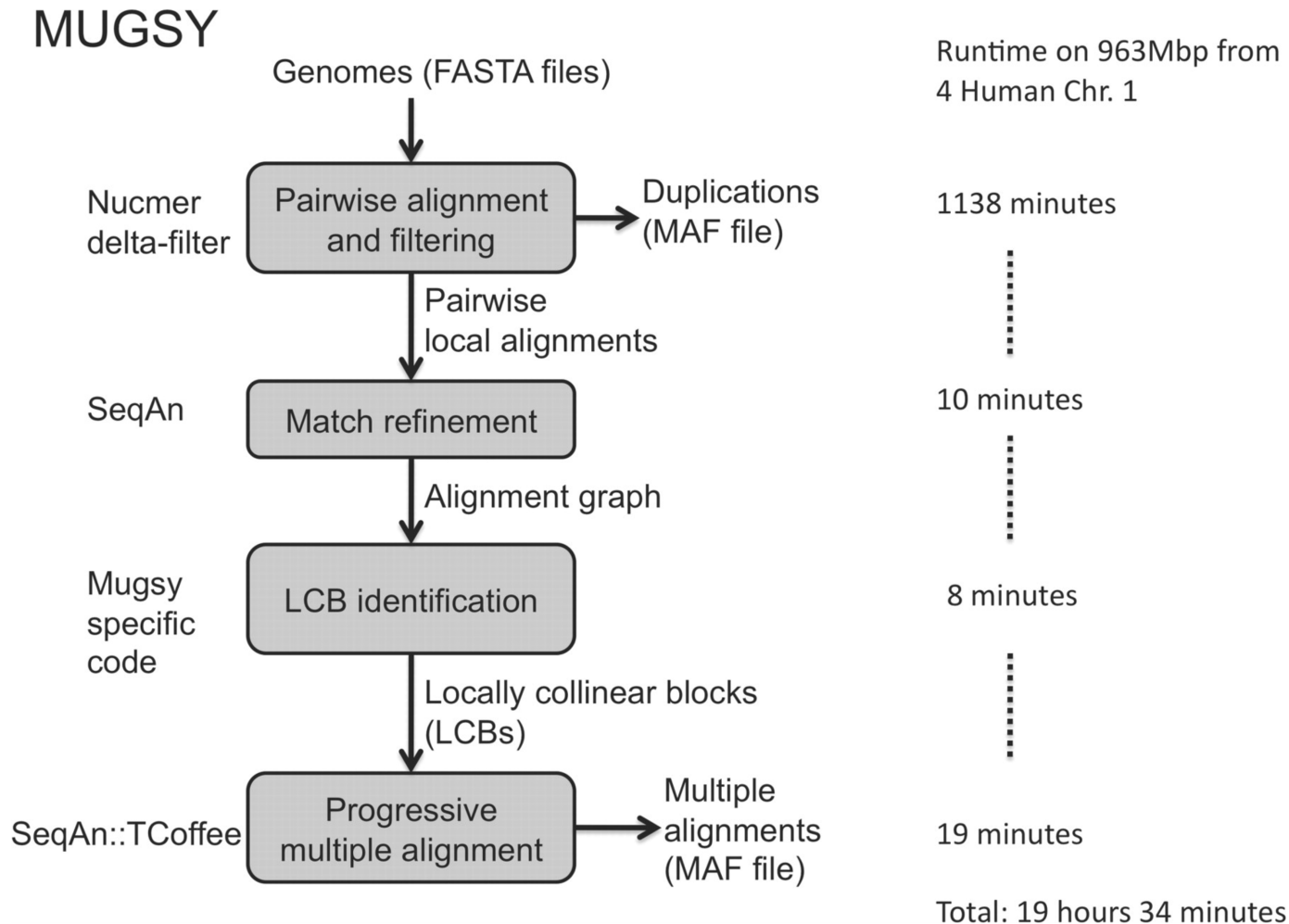
Le principe

- Basé sur le calcul d'alignements doubles filtrés avec Nucmer avec identification des duplications
- Construction d'un graphe d'alignement à partir des alignements doubles
- Identification des LCBs sur le graphe
- Alignement multiple des LCBs avec SeqAn:TCoffee

Angiuoli *et* Salzberg, *Bioinformatics* 2011

<http://mugsy.sourceforge.net>

Mugsy : principe



From: Mugsy: fast multiple alignment of closely related whole genomes

Bioinformatics. 2010;27(3):334-342. doi:10.1093/bioinformatics/btq665

Bioinformatics | © The Author(s) 2010. Published by Oxford University Press. This is an Open Access article distributed under the terms of the Creative Commons Attribution Non-Commercial License (<http://creativecommons.org/licenses/by-nc/2.5>), which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited.

Mugsy : résultats

Processing time to calculate whole-genome multiple alignments using three methods

	5 N.meningitidis	31 S.pneumoniae	57 E.coli	4 Human Chr 1
Pairwise search (min)	3	44	435	1138
+Mugsy (min)	3	56	720	37
+TBA (min)	<1	36	381	71
Mauve v2.3.1 (min)	5	377	DNF(1)	DNF(2)

From: Mugsy: fast multiple alignment of closely related whole genomes

Bioinformatics. 2010;27(3):334-342. doi:10.1093/bioinformatics/btq665

Bioinformatics | © The Author(s) 2010. Published by Oxford University Press. This is an Open Access article distributed under the terms of the Creative Commons Attribution Non-Commercial License (<http://creativecommons.org/licenses/by-nc/2.5>), which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited.

Alignement multiple : parSNP

parSNP

Les caractéristiques

- dédié à l'alignement multiple de gros jeux de données : centaines à milliers de génomes
- les génomes peuvent être drafts ou finis

Le principe

- **Basé sur le calcul des multiMUMs**
- **Processus itératif** : un algorithme « divide-and-conquer » adapté de M-GCAT recherche récursivement des matchs plus petits et calcule itérativement les suites de MultiMUMs
- **Les LCBs** sont déduits des multiMUMs colinéaires
- Les gaps entre les multiMUMs colinéaires sont alignés avec Muscle

Treangen et al. *Genome Biology* 2014

<http://harvest.readthedocs.io/en/latest/content/parsnp.html>

Alignement multiple : parSNP

parSNP

Les sorties

- Un fichier d'alignement multiple (format Gignr)
- Les SNPs
- Un arbre phylogénétique
- Un fichier des régions non alignées (optionnel)

Dans la suite Harvest il y a deux autres outils très utiles :

- **Gignr** : visualisation interactive des arbres et des variants
- **HarvestTools** : archivage et post-processing des alignements

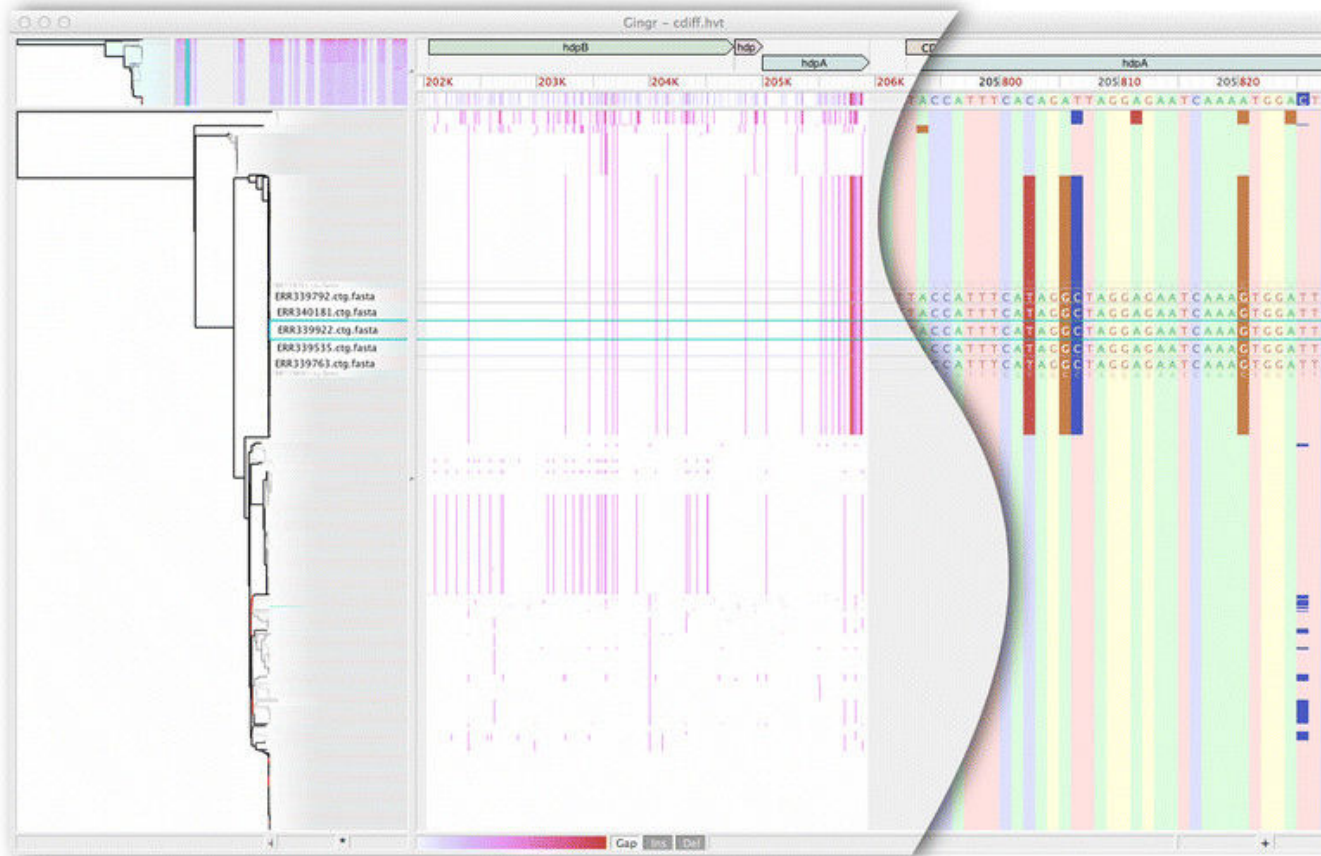
Treangen et al. *Genome Biology* 2014

<http://harvest.readthedocs.io/en/latest/index.html>

Alignement multiple : parSNP

parSNP

Exemple de résultat (visualisation avec Gingr)



Treangen et al. Genome Biology 2014

<http://harvest.readthedocs.io/en/latest/index.html>

Partie 3 : les formats d'alignement de génomes

- Pas de standart
- Beaucoup de formats propriétaires (mummer, mauve, mugsy gingr, ...) en lien avec outils de visualisation et volumétrie
- Deux formats beaucoup utilisés : XMFA et MFA.

Le format XMFA

- A l'origine lié au logiciel Mauve
- Généralisation du format multi-fasta
- Format qui inclut l'alignement complet explicite des blocs colinéaires

Exemple :

```
>seq_num:start1-end1 ± comments (sequence name, etc.)
AC-TG-NAC--TG
AC-TG-NACTGTG
...

> seq_num:startN-endN ± comments (sequence name, etc.)
AC-TG-NAC--TG
AC-TG-NACTGTG
...
= comments, and optional field-value pairs, i.e. score=12345

> seq_num:start1-end1 ± comments (sequence name, etc.)
AC-TG-NAC--TG
AC-TG-NACTGTG
...
```

<http://darlinglab.org/mauve/user-guide/files.html>

Le format MFA

- Compatible avec plusieurs Genomes Viewers (IGV, GenomeView,...)
- Suite de blocs alignés avec lignes 'a' et 's'
- Ligne s inclut : id, start de l'align, taille de l'alignement sans gap, brin, taille de la séquence source et alignement

Exemple :

```
##maf version=1 scoring=tba.v8
# tba.v8 ((human chimp) baboon) (mouse rat))
# multiz.v7
# maf_project.v5 _tba_right.maf3 mouse _tba_C
# single_cov2.v4 single_cov2 /dev/stdin

a score=23262.0
s hg16.chr7      27578828 38 + 158545518 AAA-GGGAATGTTAACCAAATGA---ATTGTCTCTTACGGTG
s panTrol.chr6  28741140 38 + 161576975 AAA-GGGAATGTTAACCAAATGA---ATTGTCTCTTACGGTG
s baboon        116834 38 + 4622798 AAA-GGGAATGTTAACCAAATGA---GTTGTCTCTTATGGTG
s mm4.chr6      53215344 38 + 151104725 -AATGGGAATGTTAAGCAAACGA---ATTGTCTCTCAGTGTG
s rn3.chr4      81344243 40 + 187371129 -AA-GGGGATGCTAAGCCAATGAGTTGTTGTCTCTCAATGTG

a score=5062.0
s hg16.chr7      27699739 6 + 158545518 TAAAGA
s panTrol.chr6  28862317 6 + 161576975 TAAAGA
s baboon        241163 6 + 4622798 TAAAGA
s mm4.chr6      53303881 6 + 151104725 TAAAGA
s rn3.chr4      81444246 6 + 187371129 taagga
```

<https://cgwb.nci.nih.gov/FAQ/FAQformat.html#format5>

Partie 4 : le problème de l'évaluation

Comment évaluer un alignement de génomes ?

- **C'est un domaine de recherche actuel en bioinformatique.**

Plusieurs types d'approches :

- **Simulations** : données simulées à partir de régions codantes ou non ;
 - **Critères « biologiques »**: couverture, pourcentage d'identité, expertise biologique (données validées « manuellement » croisement avec annotations, gènes orthologues, phylogénie,...) ;
 - **Critères « statistiques »** : post-traitement des alignements via le % d'identité, ou le score (progressiveMauve), modèles statistiques (robustesse) ou probabilistes (GRAPE*).
- **Benchmark pour les alignements de génomes bactériens, sur le modèle de l'alignathon :**
<http://compbio.soe.ucsc.edu/alignathon>

*Lunter et al. 2008. Uncertainty in homology inferences: assessing and improving genomic sequence alignment. Gen Research.

Le problème de l'évaluation

Alignathon*

• Données

- 2 jeux de données simulées avec EVOLVER : 1 échelle primates, 1 échelles mammifères
- 1 jeu de données réelles de drosophiles

• Evaluation

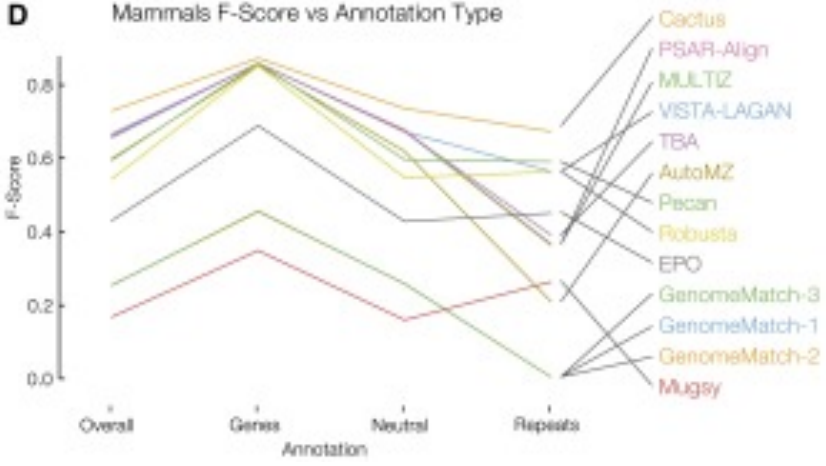
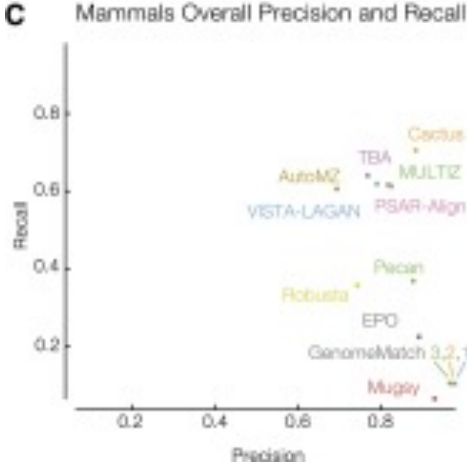
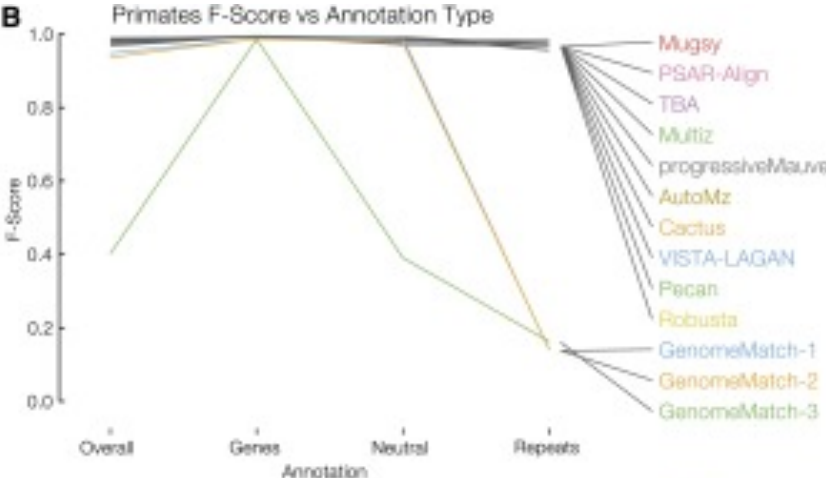
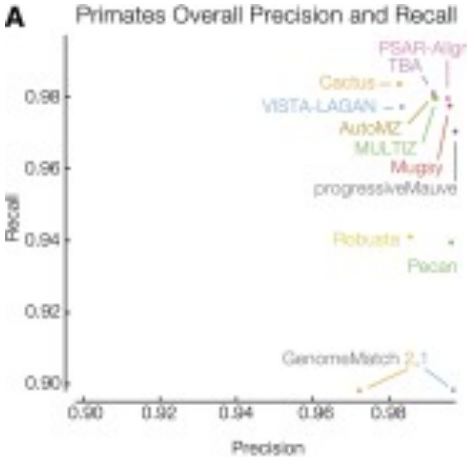
- 11 outils comparés : VISTA-LAGAN, TBA, Robusta, progressiveMauve, PSAR-Align, MultiZ, MUGSY, GenomeMatch, Pecan, EPO, Cactus
- Utilisation du F-Score :
$$Fscore = 2 \times (\text{Précision} \times \text{Rappel}) / (\text{Précision} + \text{rappel})$$
- Précision : nb de pb correctes dans l'alignement/nb total de pb de l'alignement produit
- Rappel : nb de pb correctes dans l'alignement/nb total de pb de l'alignement correct

*Earl et al. 2015. Alignathon: a competitive assessment of whole-genome alignment methods. Gen Research.

Le problème de l'évaluation

Alignathon*

- Exemple de résultat



*Earl et al. 2014. Alignathon: a competitive assessment of whole-genome alignment methods. Gen Research.

A retenir sur les outils d'alignement de génomes :

- Récents, manque de maturité ;
- Le choix des paramètres est décisif et souvent difficile en pratique ;
- Une solution unique en terme d'alignement est produite (alors que les heuristiques utilisées peuvent produire des solutions proches quasi équivalentes) ;
- Problème de choix des outils, souvent spécifiques à une problématique : identification d'exons : WABA, génomes procaryotes proches : Mummer...
- Certains outils produisent des alignements locaux non pertinents et peuvent nécessiter une étape de post-traitement.
- Il est très difficile de comparer et d'évaluer la qualité et la pertinence biologique des alignements de génomes.

Revue et benchmarks

- **Frazer KA et al. 2003.** Cross-Species Comparisons: A Review of Methods and Available Resources. *Genome Research* 13:1-12.
- **Kellis M et al. 2004.** Methods in comparative genomics: genome correspondence, gene identification and regulatory motif discovery. *J Comput Biol.* 11:319-55.
- **Couronne O et al. 2003.** Strategies and tools for whole-genome alignments. *Genome Research.* 13: 73-80.
- **Ureta-Vidal A et al. 2003.** Comparative genomics: genome-wide analysis in metazoan eukaryotes. *Nature Reviews.* 4:251-262.
- **Pollard DA et al. 2004.** Benchmarking tools for the alignment of functional noncoding DNA. *BMC Bioinformatics.* 5(1):6.
- **Miller W. 2001.** Comparison of genomic DNA sequences: solved and unsolved problems. *Bioinformatics* 17:391-397.
- **Swidam & Shamir. 2009.** Assessing the quality of whole genome alignments in bacteria. *Adv Bioinformatics*
- **Prakash & Tompa, 2007.** Measuring the accuracy of genome-size multiple alignments. *Genome Biology*
- **Chen & Tompa, 2010.** Comparative assessment of methods for aligning multiple genome sequences. *Nature Biotechnology*
- **Earl et al. 2015.** Alignathon: a competitive assessment of whole-genome alignment methods. *Genome Research.*

Fin !

Des questions ?