Calcul du Nombre de Breakpoints entre Génomes: Tour d'Horizon des Résultats Algorithmiques connus

Guillaume Fertin

LINA, UMR 6241, Université de Nantes

Outline

Introduction Générale

Que cherche-t-on à faire ?

- Comparaison de **paires** d'espèces
  - inférence de gènes fonctionnellement liés
  - construction d'arbres phylogéniques
- Pour cela :
  - on compare leurs génomes
  - une valeur est calculée, qui représente une forme de (dis)similarité entre les deux espèces

Le problème peut donc être formulé comme suit

Etant donné deux génomes G_1 et G_2, et une mesure de (dis)similarité M, calculer la valeur $M(G_1, G_2)$

Introduction Générale

Que cherche-t-on à faire ?

- Comparaison de **paires** d'espèces
  - inférence de gènes fonctionnellement liés
  - construction d'arbres phylogéniques
- Pour cela :
  - on compare leurs génomes
  - une valeur est calculée, qui représente une forme de (dis)similarité entre les deux espèces

Le problème peut donc être formulé comme suit

Etant donné deux génomes G_1 et G_2, et une mesure de (dis)similarité M, calculer la valeur $M(G_1, G_2)$

Modélisation du problème

Génomes

- Génome: séquence ordonnée de gènes
- Gènes: portions d'ADN
- Les gènes seront représentés par des entiers signés
- Signe +/- $\Rightarrow$ brin d'ADN sur lequel se situe le gène

Example

$G_1 = +1 + 3 - 4 + 5 + 2 - 6$
$G_2 = +1 + 2 - 5 - 3 + 6 - 4$

lina LABORATOIRE D'INFORMATIQUE DE NANTES ATLANTIQUE

Hypothèses Simplificatrices

Contenu en Gènes

- Les deux génomes comparés ont le même contenu en gènes
- Si un gène existe dans un seul des deux génomes, il sera supprimé de la séquence
- $\Rightarrow$ insertions/suppressions ne sont pas prises en compte dans ce modèle

Outline

Comment comparer deux génomes ?

Mesures de (Dis)Similarité

- Basées sur la *structure* des génomes
- On calcule une *mesure* qui représente la proximité (ou l'éloignement) de deux génomes
- Mesure étudiée dans cet exposé: nombre de breakpoints

Breakpoints

Definition

- Soit G_1 et G_2 deux génomes
- On suppose, wlog, que $G_1 = +1 + 2 + 3 \ldots + n$
- Alors il existe un breakpoint entre deux gènes i and $i+1$ de G_1, si dans G_2 on n'a:
  - ni: $+i\ +(i+1)$
  - ni: $-(i+1)\ -i$
- $\Rightarrow$ breakpoint = adjacence entre deux gènes dans G_1, qui n'a **pas été conservée** dans G_2

Breakpoints

Example

- Supposons $G_1 = +1 + 2 + 3 + 4 + 5 + 6$
- Si $G_2 = +1 + 2 - 5 - 4 + 3 + 6$
- Alors il existe 3 breakpoints entre G_1 et G_2
- $G_2 = +1 + 2 \bullet -5 - 4 \bullet +3 \bullet +6$

Remarques

- Cette mesure est *symétrique*: $B(G_1, G_2) = B(G_2, G_1)$ pour tous génomes G_1 et G_2
- Un breakpoint est l'extrémité d'un événement de réarrangement (inversion, transposition, etc.)
- Historiquement, une des premières mesures proposées/étudiées
- Si aucun gène n'est dupliqué, le calcul de $B(G_1, G_2)$ est polynomial

Breakpoints

Example

- Supposons $G_1 = +1 + 2 + 3 + 4 + 5 + 6$
- Si $G_2 = +1 + 2 - 5 - 4 + 3 + 6$
- Alors il existe 3 breakpoints entre G_1 et G_2
- $G_2 = +1 + 2 \bullet -5 - 4 \bullet +3 \bullet +6$

Remarques

- Cette mesure est *symétrique*: $B(G_1, G_2) = B(G_2, G_1)$ pour tous génomes G_1 et G_2
- Un breakpoint est l'extrémité d'un événement de réarrangement (inversion, transposition, etc.)
- Historiquement, une des premières mesures proposées/étudiées
- Si aucun gène n'est dupliqué, le calcul de $B(G_1, G_2)$ est polynomial

Outline

Duplications

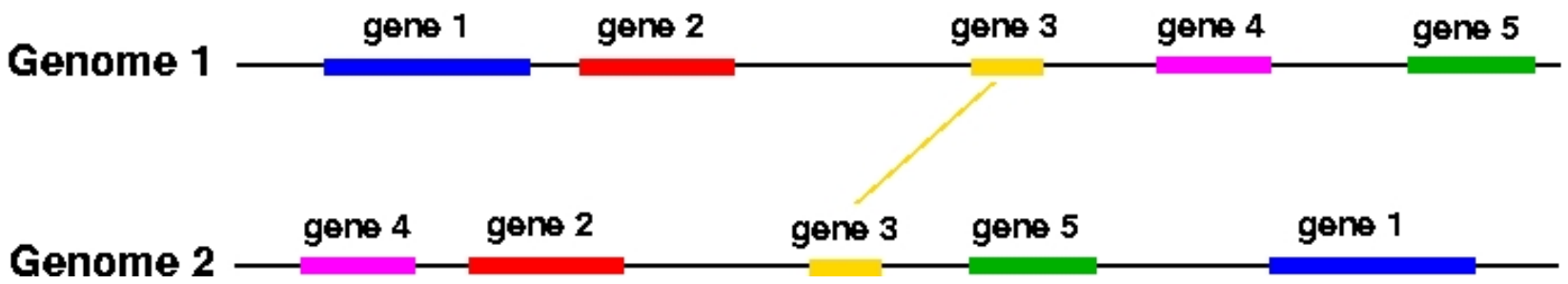
Les duplications doivent être prises en compte

- Calculer $B(G_1, G_2)$ est polynomial, quand ni G_1 ni G_2 ne contiennent de duplications
- On sait désormais que cette hypothèse est trop restrictive:
  - $\sim$ 15% chez l'humain
  - $\sim$ 16% chez la levure
  - $\sim$ 25% chez Arabidopsis

- Mais comment gérer les duplications ?

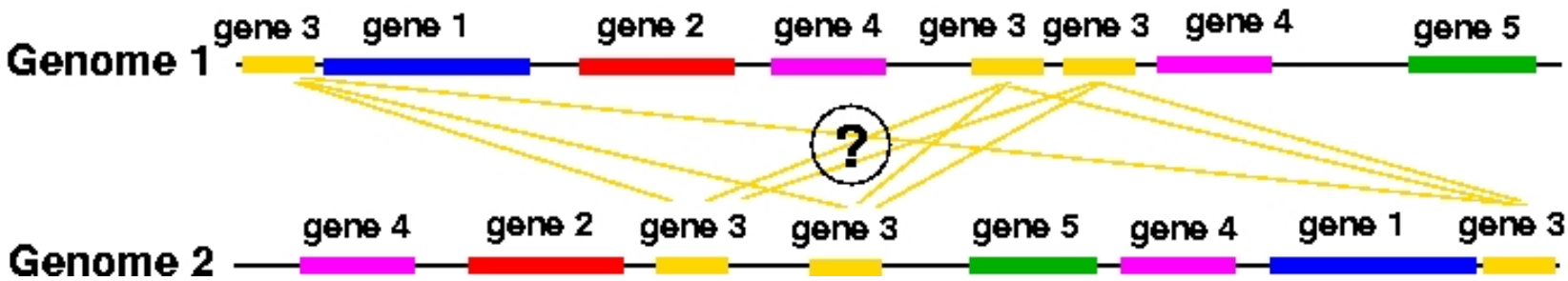
Premier problème

Sans duplication, pas d'ambiguïté



Premier problème

Avec duplications, comment faire ?



Gérer les ambiguïtés

On cherche à retrouver une *permutation*

- $\Rightarrow$ Etablir un matching $\mathcal{M}$ entre gènes de G_1 et gènes de G_2, pour chaque famille de gènes
- Possiblement, en supprimant des gènes
- Le but étant de revenir à une permutation $\Rightarrow$ amibiguïtés inexistantes

Deux possibilités

- **Exemplarisation** entre G_1 et G_2
- **Matching Maximum** entre G_1 et G_2

Gérer les ambiguïtés

On cherche à retrouver une *permutation*

- $\Rightarrow$ Etablir un matching $\mathcal{M}$ entre gènes de G_1 et gènes de G_2, pour chaque famille de gènes
- Possiblement, en supprimant des gènes
- Le but étant de revenir à une permutation $\Rightarrow$ amibiguïtés inexistantes

Deux possibilités

- **Exemplarisation** entre G_1 et G_2
- **Matching Maximum** entre G_1 et G_2

Exemplarisation vs Matching Maximum

Exemplarisation

Dans le matching $\mathcal{M}$, on garde **un seul** gène de chaque famille
$\Rightarrow$ Gène ancestral

Matching Maximum

Dans le matching $\mathcal{M}$, on garde **le maximum** de gènes de chaque famille
$\Rightarrow$ On garde le maximum d'informations dans les génomes

Exemplarisation vs Matching Maximum

Exemplarisation

Dans le matching $\mathcal{M}$, on garde **un seul** gène de chaque famille
$\Rightarrow$ Gène ancestral

Matching Maximum

Dans le matching $\mathcal{M}$, on garde **le maximum** de gènes de chaque famille
$\Rightarrow$ On garde le maximum d'informations dans les génomes

Exemplarisation

Example

$$G_1 = +1\ +2\ +3\ -2\ +5\ +4\ +4\ -3$$

$$G_2 = +3\ -1\ +2\ +1\ +5\ -3\ +2\ +4\ +3$$

Une exemplarisation possible

$$G_1^E = +1\ +2\ +3\ +5\ +4$$

$$G_2^E = +1\ +5\ -3\ +2\ +4$$

$\Rightarrow$ L'exemplarisation est notée par le triplet $(G_1^E, G_1^E, \mathcal{M})$

Exemplarisation

Example

$$G_1 = +1\ +2\ +3\ -2\ +5\ +4\ +4\ -3$$

$$G_2 = +3\ -1\ +2\ +1\ +5\ -3\ +2\ +4\ -3$$

Une exemplarisation possible

$$G_1^E = +1\ +2\ +3\ +5\ +4$$

$$G_2^E = +1\ +5\ -3\ +2\ +4$$

$\Rightarrow$ L'exemplarisation est notée par le triplet $(G_1^E, G_1^E, \mathcal{M})$

Matching Maximum

Example

$$G_1 = +1\ +2\ +3\ -2\ +5\ +4\ +4\ -3$$

$$G_2 = +3\ -1\ +2\ +1\ +5\ -3\ +2\ +4\ +3$$

Un matching maximum possible

$$G_1^M = +1\ +2\ +3\ -2\ +5\ +4\ -3$$

$$G_2^M = +3\ -1\ +2\ +5\ -3\ +2\ +4$$

$\Rightarrow$ Le matching maximum est noté par le triplet $(G_1^M, G_1^M, \mathcal{M})$

Matching Maximum

Example

$$G_1 = +1\ +2\ +3\ -2\ +5\ +4\ +4\ -3$$

$$G_2 = +3\ -1\ +2\ +1\ +5\ -3\ +2\ +4\ -3$$

Un matching maximum possible

$$G_1^M = +1\ +2\ +3\ -2\ +5\ +4\ -3$$

$$G_2^M = +3\ -1\ +2\ +5\ -3\ +2\ +4$$

$\Rightarrow$ Le matching maximum est noté par le triplet $(G_1^M, G_1^M, \mathcal{M})$

Question Subsidiaire (pour les deux versions)

Quel matching choisir ?

- Pour une instance (G_1, G_2) et un modèle de matching donnés, quel est le **meilleur** matching ?
- **Critère retenu**: on recherche le matching qui **minimise** le nombre de breakpoints
- Selon le principe de parcimonie

Outline

Définitions

Nommons les Problèmes

- EBD: **E**xemplar **B**reakpoint **D**istance
- MBD: **M**aximum matching **B**reakpoint **D**istance

Remarques

- Le nom est trompeur: ce ne sont pas des distances au sens mathématique !
- Dénomination "historique"

Définitions

Terminologie

- $occ(G, g)$: nombre d'occurrences du gène g dans le génome G
- $occ(G)$: maximum des $occ(G, g)$ sur toutes les familles g de gènes
- Instance (G_1, G_2) de type (a, b): $occ(G_1) = a$ et $occ(G_2) = b$
- Instance (G_1, G_2) équilibrée: pour tout gène g,

$$occ(G_1, g) = occ(G_2, g)$$

Définitions

Example

$$G_1 = +1\ +2\ +3\ -2\ +5\ +4\ +4\ -3$$

$$G_2 = +3\ -1\ +2\ +1\ +5-3\ +2\ +4\ +3$$

- $occ(G_1, 3) = 2$
- $occ(G_2, 3) = 3$
- (G_1, G_2) est une instance de type $(2, 3)$
- (G_1, G_2) n'est pas une instance équilibrée

Remarque

- Si (G_1, G_2) est une instance équilibrée, alors c'est une instance de type (a, a)
- ...mais l'inverse n'est pas vrai

Définitions

Example

$$G_1 = +1\ +2\ +3\ -2\ +5\ +4\ +4\ -3$$

$$G_2 = +3\ -1\ +2\ +1\ +5-3\ +2\ +4\ +3$$

- $occ(G_1, 3) = 2$
- $occ(G_2, 3) = 3$
- (G_1, G_2) est une instance de type $(2, 3)$
- (G_1, G_2) n'est pas une instance équilibrée

Remarque

- Si (G_1, G_2) est une instance équilibrée, alors c'est une instance de type (a, a)
- ...mais l'inverse n'est pas vrai

Complexité algorithmique du problème EBD

Theorem (Bryant-2000)

EBD *est NP-complet, même pour les instances de type* $(1, 2)$

Theorem (Angibaud et al.-2007)

EBD *is APX-dur, même pour les instances de type* $(1, 2)$

Complexité algorithmique du problème EBD

Theorem (Bryant-2000)

EBD *est NP-complet, même pour les instances de type* $(1, 2)$

Theorem (Angibaud et al.-2007)

EBD *is APX-dur, même pour les instances de type* $(1, 2)$

Peut-on quand même approcher EBD ?

Theorem (Chen et al.-2006)

EBD *ne peut pas être approché en-deçà d'un ratio* 1.36*, même pour les instances de type* $(2, 2)$*.*

Remarque

Résultat déjà présent (implicitement) dans [Bryant-2000]

Peut-on quand même approcher EBD ?

Un nouveau problème: ZEBD

Soit ZEBD le problème suivant: existe-t-il une exemplarisation $(G_1^E, G_2^E, \mathcal{M})$ de (G_1, G_2) telle que $B(G_1^E, G_2^E) = 0$?

Theorem (Chen et al.-2006)

ZEBD *est NP-complet, même pour les instances de type* $(3, 3)$

Theorem (Angibaud et al.-2008)

ZEBD *est NP-complet, même pour les instances de type* $(2, k)$*, où k est non borné*

Peut-on quand même approcher EBD ?

Un nouveau problème: ZEBD

Soit ZEBD le problème suivant: existe-t-il une exemplarisation $(G_1^E, G_2^E, \mathcal{M})$ de (G_1, G_2) telle que $B(G_1^E, G_2^E) = 0$?

Theorem (Chen et al.-2006)

ZEBD *est NP-complet, même pour les instances de type* $(3, 3)$

Theorem (Angibaud et al.-2008)

ZEBD *est NP-complet, même pour les instances de type* $(2, k)$*, où k est non borné*

Peut-on quand même approcher EBD ?

Theorem (Chen et al.-2006, Angibaud et al.-2008)

EBD *ne peut être approché en-deçà d'aucun ratio:*

- *même pour les instances de type* $(3,3)$
- *même pour les instances de type* $(2,k)$*, où* k *est non borné*

EBD: vous n'auriez pas du positif ?

Heuristiques

- Branch and Bound [Sankoff-1999]
- Divide and Conquer + Branch and Bound [Thach Nguyen et al.-2005]
- ELCS: Algorithme glouton basé sur le LCS (Longest Common Subsequence) [Angibaud et al.-2007]

ELCS: le principe

1. Trouver S, le LCS de G_1 et G_2 (à un renversement total - ordre et signe - près)
2. Exemplariser S arbitrairement: on obtient S'
3. Matcher les gènes de S'
4. Retirer du reste de G_1 (resp. G_2) les gènes de S'
5. Itérer le processus jusqu'à avoir matché tous les gènes

EBD: vous n'auriez pas du positif ?

Heuristiques

- Branch and Bound [Sankoff-1999]
- Divide and Conquer + Branch and Bound [Thach Nguyen et al.-2005]
- ELCS: Algorithme glouton basé sur le LCS (Longest Common Subsequence) [Angibaud et al.-2007]

ELCS: le principe

1. Trouver S, le LCS de G_1 et G_2 (à un renversement total - ordre et signe - près)
2. Exemplariser S arbitrairement: on obtient S'
3. Matcher les gènes de S'
4. Retirer du reste de G_1 (resp. G_2) les gènes de S'
5. Itérer le processus jusqu'à avoir matché tous les gènes

EBD: vous n'auriez pas du positif ?

Résultats exacts obtenus par programmation linéaire en $(0,1)$

- Approche utilisée dans [Angibaud et al.-2007]
- Transformer le problème EBD en un problème de type SAT + fonction objectif
- Utiliser un solveur SAT pour calculer la solution
- Ré-importer la solution dans notre problème
- Donne des résultats exacts... mais pas toujours:
  - 61 résultats obtenus
  - sur les 66 attendus dans notre jeu de données

EBD: vous n'auriez pas du positif ?

Heuristique ELCS vs Résultats exacts

- 12 génomes de γ-protéobactéries
- Génomes de taille $\sim$500 à $\sim$5500 gènes
- 66 comparaisons de génomes deux à deux
- 61 résultats exacts obtenus, et comparés à l'heuristique ELCS

Exemplarisation: Exact vs ELCS		
Pire Cas	Meilleur Cas	Moyenne (61 instances)
96.51%	100%	99.36%

EBD: vous n'auriez pas du positif ?

Heuristique ELCS vs Résultats exacts

- 12 génomes de γ-protéobactéries
- Génomes de taille $\sim$500 à $\sim$5500 gènes
- 66 comparaisons de génomes deux à deux
- 61 résultats exacts obtenus, et comparés à l'heuristique ELCS

Exemplarisation: Exact vs ELCS		
Pire Cas	**Meilleur Cas**	**Moyenne (61 instances)**
96.51%	100%	99.36%

Complexité algorithmique du problème MBD

Theorem (Bryant-2000, Angibaud et al.-2007)

- MBD *est NP-complet, même pour les instances de type* $(1, 2)$
- MBD *est APX-dur, même pour les instances de type* $(1, 2)$

Proof.

Pour toute instance de type $(1, b)$,

exemplarisation = maximum matching

$\Rightarrow$ Instances de type $(1, b)$: les résultats pour EBD sont valables pour MBD $\square$

Complexité algorithmique du problème MBD

Theorem (Bryant-2000, Angibaud et al.-2007)

- MBD *est NP-complet, même pour les instances de type* $(1, 2)$
- MBD *est APX-dur, même pour les instances de type* $(1, 2)$

Proof.

Pour toute instance de type $(1, b)$,

exemplarisation = maximum matching

$\Rightarrow$ Instances de type $(1, b)$: les résultats pour EBD sont valables pour MBD $\square$

Peut-on quand même approcher MBD ?

Contrairement à EBD, je ne connais pas de résultat indiquant que MBD ne peut pas être approché en-deçà d'un ratio r donné.
D'ailleurs...

Un nouveau problème: ZMBD

Soit ZMBD le problème suivant: existe-t-il un matching maximum $(G_1^M, G_2^M, \mathcal{M})$ de (G_1, G_2) tel que $B(G_1^M, G_2^M) = 0$?

Theorem (Angibaud et al.-2008)

ZMBD *est polynomial*

$\Rightarrow$ côté approximation, il y a donc de l'espoir pour MBD

Peut-on quand même approcher MBD ?

Contrairement à EBD, je ne connais pas de résultat indiquant que MBD ne peut pas être approché en-deçà d'un ratio r donné.
D'ailleurs...

Un nouveau problème: ZMBD

Soit ZMBD le problème suivant: existe-t-il un matching maximum $(G_1^M, G_2^M, \mathcal{M})$ de (G_1, G_2) tel que $B(G_1^M, G_2^M) = 0$?

Theorem (Angibaud et al.-2008)

ZMBD *est polynomial*

$\Rightarrow$ côté approximation, il y a donc de l'espoir pour MBD

Peut-on quand même approcher MBD ?

Contrairement à EBD, je ne connais pas de résultat indiquant que MBD ne peut pas être approché en-deçà d'un ratio r donné.
D'ailleurs...

Un nouveau problème: ZMBD

Soit ZMBD le problème suivant: existe-t-il un matching maximum $(G_1^M, G_2^M, \mathcal{M})$ de (G_1, G_2) tel que $B(G_1^M, G_2^M) = 0$?

Theorem (Angibaud et al.-2008)

ZMBD *est polynomial*

$\Rightarrow$ côté approximation, il y a donc de l'espoir pour MBD

lina LABORATOIRE D'INFORMATIQUE DE NANTES ATLANTIQUE

De l'espoir ?

Approximation pour MBD: de l'espoir, mais...

- Aucun résultat dans le cas général (comme pour EBD)
- Raison essentielle dans les deux cas: dès qu'on supprime des gènes, il est difficile d'assurer que le nombre de breakpoints n'augmente pas de façon déraisonnable
- $\Rightarrow$ Solution: regarder des instances équilibrées en matching maximum

De l'espoir ?

Approximation pour MBD: de l'espoir, mais...

- Aucun résultat dans le cas général (comme pour EBD)
- Raison essentielle dans les deux cas: dès qu'on supprime des gènes, il est difficile d'assurer que le nombre de breakpoints n'augmente pas de façon déraisonnable
- $\Rightarrow$ Solution: regarder des instances équilibrées en matching maximum

De l'espoir ?

Approximation pour MBD: de l'espoir, mais...

- Aucun résultat dans le cas général (comme pour EBD)
- Raison essentielle dans les deux cas: dès qu'on supprime des gènes, il est difficile d'assurer que le nombre de breakpoints n'augmente pas de façon déraisonnable
- $\Rightarrow$ Solution: regarder des instances équilibrées en matching maximum

MBD pour les instances équilibrées

Definition

MBD pour les instances équilibrées: problème MBD_{Eq}

Theorem (Goldstein et al.-2004, Kolman et al.-2006)

Il existe des algorithmes d'approximation pour MBD_{Eq}*:*

- *De ratio* 1.1037 *pour les instances équilibrées de type* $(2,2)$
- *De ratio* 4 *pour les instances équilibrées de type* $(3,3)$
- *De ratio* $O(k)$ *pour les instances équilibrées de type* (k,k)

MBD pour les instances équilibrées

Definition

MBD pour les instances équilibrées: problème MBD_{Eq}

Theorem (Goldstein et al.-2004, Kolman et al.-2006)

Il existe des algorithmes d'approximation pour MBD_{Eq}*:*

- *De ratio* 1.1037 *pour les instances équilibrées de type* $(2,2)$
- *De ratio* 4 *pour les instances équilibrées de type* $(3,3)$
- *De ratio* $O(k)$ *pour les instances équilibrées de type* (k,k)

MBD pour les instances équilibrées

Definition

MBD pour les instances équilibrées: problème MBD_{Eq}

Theorem (Goldstein et al.-2004, Kolman et al.-2006)

Il existe des algorithmes d'approximation pour MBD_{Eq}*:*

- *De ratio* 1.1037 *pour les instances équilibrées de type* $(2,2)$
- *De ratio* 4 *pour les instances équilibrées de type* $(3,3)$
- *De ratio* $O(k)$ *pour les instances équilibrées de type* (k,k)

MBD_{Eq} reste un problème difficile

Theorem (Goldstein et al.-2004)

- MBD_{Eq} *est NP-complet, même pour les instances de type* $(2, 2)$
- MBD_{Eq} *est APX-dur, même pour les instances de type* $(2, 2)$

$\mathbf{MBD}_{Eq}$ reste un problème difficile

Theorem (Goldstein et al.-2004)

- MBD_{Eq} *est NP-complet, même pour les instances de type* $(2,2)$
- MBD_{Eq} *est APX-dur, même pour les instances de type* $(2,2)$

Retour sur MBD

Heuristiques

MLCS: Algorithme glouton basé sur le LCS (Longest Common Subsequence)

Heuristique MLCS vs Résultats exacts

- 12 génomes de γ-protéobactéries
- Génomes de taille $\sim$500 à $\sim$5500 gènes
- 66 comparaisons de génomes deux à deux
- Les 66 résultats exacts ont été obtenus, et comparés à l'heuristique MLCS

Matching Maximum: Exact vs MLCS		
Pire Cas	**Meilleur Cas**	**Moyenne (66 instances)**
95.19%	100%	99.00%

Retour sur MBD

Heuristiques

MLCS: Algorithme glouton basé sur le LCS (Longest Common Subsequence)

Heuristique MLCS vs Résultats exacts

- 12 génomes de γ-protéobactéries
- Génomes de taille $\sim$500 à $\sim$5500 gènes
- 66 comparaisons de génomes deux à deux
- Les 66 résultats exacts ont été obtenus, et comparés à l'heuristique MLCS

Matching Maximum: Exact vs MLCS		
Pire Cas	Meilleur Cas	Moyenne (66 instances)
95.19%	100%	99.00%

Retour sur MBD

Heuristiques

MLCS: Algorithme glouton basé sur le LCS (Longest Common Subsequence)

Heuristique MLCS vs Résultats exacts

- 12 génomes de γ-protéobactéries
- Génomes de taille $\sim$500 à $\sim$5500 gènes
- 66 comparaisons de génomes deux à deux
- Les 66 résultats exacts ont été obtenus, et comparés à l'heuristique MLCS

Matching Maximum: Exact vs MLCS		
Pire Cas	**Meilleur Cas**	**Moyenne (66 instances)**
95.19%	100%	99.00%

Et si on comptait les adjacences ?

Definition (Problèmes EADJ et MADJ)

- $Adj(G_1, G_2)$=nombre d'adjacences entre G_1 et G_2
- EADJ: Trouver une exemplarisation $(G_1^E, G_2^E, \mathcal{M})$ qui maximise $Adj(G_1^E, G_2^E)$
- MADJ: Trouver un matching maximum $(G_1^M, G_2^M, \mathcal{M})$ qui maximise $Adj(G_1^M, G_1^M)$

Et si on comptait les adjacences ?

Definition (Problèmes EADJ et MADJ)

- $Adj(G_1, G_2)$=nombre d'adjacences entre G_1 et G_2
- EADJ: Trouver une exemplarisation $(G_1^E, G_2^E, \mathcal{M})$ qui maximise $Adj(G_1^E, G_2^E)$
- MADJ: Trouver un matching maximum $(G_1^M, G_2^M, \mathcal{M})$ qui maximise $Adj(G_1^M, G_1^M)$

Et si on comptait les adjacences ?

Definition (Problèmes EADJ et MADJ)

- $Adj(G_1, G_2)$=nombre d'adjacences entre G_1 et G_2
- EADJ: Trouver une exemplarisation $(G_1^E, G_2^E, \mathcal{M})$ qui maximise $Adj(G_1^E, G_2^E)$
- MADJ: Trouver un matching maximum $(G_1^M, G_2^M, \mathcal{M})$ qui maximise $Adj(G_1^M, G_1^M)$

Et si on comptait les adjacences ?

Theorem

- EADJ *et* EBD *sont équivalents*
- MADJ *et* MBD *sont équivalents*

Proof.

Soit deux génomes (G_1, G_2) sans duplication, ayant le même contenu en gènes, avec $|G_1| = |G_2| = n$. Alors,

$$B(G_1, G_2) + Adj(G_1, G_2) = n - 1$$

□

Remarque

Problèmes équivalents si on veut optimiser, pas si on veut approcher !

Et si on comptait les adjacences ?

Theorem

- EADJ *et* EBD *sont équivalents*
- MADJ *et* MBD *sont équivalents*

Proof.

Soit deux génomes (G_1, G_2) sans duplication, ayant le même contenu en gènes, avec $|G_1| = |G_2| = n$. Alors,

$$B(G_1, G_2) + Adj(G_1, G_2) = n - 1$$

□

Remarque

Problèmes équivalents si on veut optimiser, pas si on veut approcher !

Et si on comptait les adjacences ?

Theorem

- EADJ *et* EBD *sont équivalents*
- MADJ *et* MBD *sont équivalents*

Proof.

Soit deux génomes (G_1, G_2) sans duplication, ayant le même contenu en gènes, avec $|G_1| = |G_2| = n$. Alors,

$$B(G_1, G_2) + Adj(G_1, G_2) = n - 1$$

□

Remarque

Problèmes équivalents si on veut optimiser, pas si on veut approcher !

Et si on comptait les adjacences ?

$MADJ_{Eq}$ = MADJ restreint aux instances équilibrées

Theorem (Angibaud et al.-2008)

Il existe des algorithmes d'approximation pour $MADJ_{Eq}$*:*

- *De ratio* 1.1442 *pour les instances équilibrées de type* $(2, 2)$
- *De ratio* 3 *pour les instances équilibrées de type* $(3, 3)$
- *De ratio* 4 *pour les instances équilibrées de type* (k, k)

Et si on comptait les adjacences ?

$MADJ_{Eq}$ = MADJ restreint aux instances équilibrées

Theorem (Angibaud et al.-2008)

Il existe des algorithmes d'approximation pour $MADJ_{Eq}$*:*

- *De ratio 1.1442 pour les instances équilibrées de type* $(2, 2)$
- *De ratio 3 pour les instances équilibrées de type* $(3, 3)$
- *De ratio 4 pour les instances équilibrées de type* (k, k)

Et si on comptait les adjacences ?

$MADJ_{Eq}$ = MADJ restreint aux instances équilibrées

Theorem (Angibaud et al.-2008)

Il existe des algorithmes d'approximation pour $MADJ_{Eq}$*:*

- *De ratio 1.1442 pour les instances équilibrées de type* $(2,2)$
- *De ratio 3 pour les instances équilibrées de type* $(3,3)$
- *De ratio 4 pour les instances équilibrées de type* (k,k)

Et si on comptait les adjacences ?

MADJ_{Eq} = MADJ restreint aux instances équilibrées

Theorem (Angibaud et al.-2008)

Il existe des algorithmes d'approximation pour MADJ_{Eq}*:*

- *De ratio* 1.1442 *pour les instances équilibrées de type* $(2,2)$
- *De ratio* 3 *pour les instances équilibrées de type* $(3,3)$
- *De ratio* 4 *pour les instances équilibrées de type* (k,k)

Conclusion

Résumé du spectacle

- EBD:
  - APX-dur dans tous les cas
  - Impossible d'approcher à un quelconque ratio pour les instances de type (a, b), $a, \geq 3$
  - Existence d'heuristiques efficaces
- MBD:
  - APX-dur dans tous les cas
  - Rien de connu en terme d'(in)approximabilité
  - Existence d'heuristiques efficaces
  - Cas particulier des instances équilibrées: algorithmes d'approximation existants
- $MADJ_{Eq}$: algorithmes d'approximation existants

Conclusion

Résumé du spectacle

- EBD:
  - APX-dur dans tous les cas
  - Impossible d'approcher à un quelconque ratio pour les instances de type (a, b), $a, \geq 3$
  - Existence d'heuristiques efficaces
- MBD:
  - APX-dur dans tous les cas
  - Rien de connu en terme d'(in)approximabilité
  - Existence d'heuristiques efficaces
  - Cas particulier des instances équilibrées: algorithmes d'approximation existants
- $MADJ_{Eq}$: algorithmes d'approximation existants

Conclusion

Résumé du spectacle

- EBD:
  - APX-dur dans tous les cas
  - Impossible d'approcher à un quelconque ratio pour les instances de type (a, b), $a, \geq 3$
  - Existence d'heuristiques efficaces
- MBD:
  - APX-dur dans tous les cas
  - Rien de connu en terme d'(in)approximabilité
  - Existence d'heuristiques efficaces
  - Cas particulier des instances équilibrées: algorithmes d'approximation existants
- MADJ_{Eq}: algorithmes d'approximation existants

Conclusion

Problèmes ouverts

- Approximation de EBD: que peut-on dire des "petits cas", par exemple des instances de type $(2, 3)$?
- Approximation de MBD: quid du cas général, càd non nécessairement équilibré ?
  $\Rightarrow$ Difficulté liée à la suppression des gènes
- Question entêtante à propos de ZEBD (savoir si'il existe une exemplarisation à 0 breakpoint): qui de la complexité des instances de type $(2, 2)$?

Conclusion

Problèmes ouverts

- Approximation de EBD: que peut-on dire des "petits cas", par exemple des instances de type $(2, 3)$?
- Approximation de MBD: quid du cas général, càd non nécessairement équilibré ?
  $\Rightarrow$ Difficulté liée à la suppression des gènes
- Question entêtante à propos de ZEBD (savoir si'il existe une exemplarisation à 0 breakpoint): qui de la complexité des instances de type $(2, 2)$?

Conclusion

Problèmes ouverts

- Approximation de EBD: que peut-on dire des "petits cas", par exemple des instances de type $(2,3)$?
- Approximation de MBD: quid du cas général, càd non nécessairement équilibré ?
  $\Rightarrow$ Difficulté liée à la suppression des gènes
- Question entêtante à propos de ZEBD (savoir si'il existe une exemplarisation à 0 breakpoint): qui de la complexité des instances de type $(2,2)$?

Conclusion

Problèmes ouverts

- Approximation de EBD: que peut-on dire des "petits cas", par exemple des instances de type $(2, 3)$?
- Approximation de MBD: quid du cas général, càd non nécessairement équilibré ?
  $\Rightarrow$ Difficulté liée à la suppression des gènes
- Question entêtante à propos de ZEBD (savoir si'il existe une exemplarisation à 0 breakpoint): qui de la complexité des instances de type $(2, 2)$?

Bibliography

D. Sankoff

Genome Rearrangement with Gene Families
Bioinformatics, vol. 15, pp. 909–917, 1999

D. Bryant

The Complexity of Calculating Exemplar Distances
Comparative Genomics, pp. 207–212, 2000

A. Goldstein, P. Kolman, J. Zheng

Minimum Common String Partition Problem: Hardness and Approximations
In Proc. **ISAAC 2004**, LNCS vol. 3341, pp. 484–495, 2004

C. Thach Nguyen, Y.C. Tai, L. Zhang

Divide-and-Conquer Approach for the Exemplar Breakpoint Distance
Bioinformatics, vol. 21, pp. 2171–2176, 2006

Z. Chen, B. Fu, B. Zhu

The Approximability of the Exemplar Breakpoint Distance Problem
In Proc. **AAIM 2006**, LNCS vol. 4041, pp. 291–302, 2006

P. Kolman, T. Waleń

Reversal Distance for Strings with Duplicates: Linear Time Approximation using Hitting Set
In Proc. **WAOA 2006**, LNCS vol. 4368, pp. 281–291, 2006

S. Angibaud, G. Fertin, I. Rusu, A. Thévenin, S. Vialette

Efficient Tools for Computing the Number of Breakpoints and the Number of Adjacencies between two Genomes with Duplicate Genes
Journal of Computational Biology, 2007. Submitted.

S. Angibaud, G. Fertin, I. Rusu, A. Thévenin, S. Vialette

Annexe: Caractéristiques des Genomes étudiés

Caractéristiques principales			
Genome	Nombre de gènes	Nombre de familles de gènes	Pourcentage de duplications
Baphi	564	549	2.66
Ecoli	4183	3423	18.17
Haein	1709	1531	10.42
Paeru	5540	4500	18.77
Pmult	2015	1811	10.12
Salty	4203	3456	17.77
Wglos	653	627	3.98
Xaxon	4192	3634	13.31
Xcamp	4029	3468	13.92
Xfast	2680	2346	12.46
Ypest-CO92	3599	3021	16.06
Ypest-KIM	3879	3236	16.58

Table: Caractéristiques principales des 12 génomes de γ-Protéobactérie étudiés

lina LABORATOIRE D'INFORMATIQUE DE NANTES ATLANTIQUE