Approche pseudo-booléenne pour le calcul de distance entre génomes avec duplications

S. Angibaud[1] G. Fertin[1] I. Rusu[1]
A. Thévenin[2] S. Vialette[2]

[1]Laboratoire d'Informatique de Nantes-Atlantique (LINA), Nantes

[2]Laboratoire de Recherche en Informatique (LRI), Orsay

28 septembre 2007, Marne-la-Vallée

- A General Framework for Computing Rearrangement Distances between Genomes with Duplicates
  Journal of Computational Biology 2007
  Angibaud, S. and Fertin, G. and Rusu, I. and Vialette, S.

- A Pseudo-boolean Programming Approach for Computing the Breakpoint Distance Between Two Genomes with Duplicate Genes
  Proceedings. RECOMB-CG 2007
  Angibaud, S. and Fertin, G. and Rusu, I. and Thévenin, A. and Vialette, S.

Plan

Plan

Représentation des génomes

séquence signée

- Alphabet $\Sigma \Leftrightarrow$ *familles de gènes*

Exemple

- $\Sigma = \{1, 2, 3 \ldots 10\}$

Représentation des génomes

séquence signée

- Alphabet $\Sigma \Leftrightarrow$ *familles de gènes*
- Génome : séquence de *gènes signés*

Exemple

- $\Sigma = \{1, 2, 3 \ldots 10\}$
- $G_1 = 1\ 2\ -3\ -7\ 4\ 5\ 7\ -8\ 10\ -9\ 4\ -6$

Représentation des génomes

séquence signée

- Alphabet $\Sigma \Leftrightarrow$ *familles de gènes*
- Génome : séquence de *gènes signés*
- Soit $G_1[k]$ le $k^{\text{ième}}$ gène de G_1

Exemple

- $\Sigma = \{1, 2, 3 \ldots 10\}$
- $G_1 = 1\ 2\ -3\ -7\ 4\ 5\ 7\ -8\ 10\ -9\ 4\ -6$
- $G_1[4] = -7$

Représentation des génomes

séquence signée

- Alphabet $\Sigma \Leftrightarrow$ *familles de gènes*
- Génome : séquence de *gènes signés*
- Soit $G_1[k]$ le $k^{\text{ième}}$ gène de G_1
- Deux gènes sont de la même famille s'ils ont la même valeur absolue

Exemple

- $\Sigma = \{1, 2, 3 \ldots 10\}$
- $G_1 = 1\ 2\ -3$ -7 $4\ 5$ 7 $-8\ 10\ -9\ 4\ -6$
- $G_1[4] = -7$
- $G_1[4]$ et $G_1[7]$ sont de la même famille de gènes

Problématique

- **Entrée :** Deux génomes G_1 et G_2 **sans** duplication
- **Sortie :** Une distance entre G_1 et G_2

Problématique

- **Entrée :** Deux génomes G_1 et G_2 **sans** duplication
- **Sortie :** Une distance entre G_1 et G_2

- plusieurs distances existent : *breakpoints*, *intervalles communs*, *intervalles conservés*, SAD . . .

Problématique

- **Entrée :** Deux génomes G_1 et G_2 **sans** duplication
- **Sortie :** Une distance entre G_1 et G_2

- plusieurs distances existent : *breakpoints*, *intervalles communs*, *intervalles conservés*, SAD ...

Définition :

- Une paire de gènes consécutifs (a, b) sur G_1 induit un *breakpoint* si ni (a, b) ni $(\overline{b}, \overline{a})$ n'est une paire de gènes consécutifs sur G_2, où $\overline{g}$ exprime le gène g de signe opposé.

$$\begin{aligned} G_1 = & \quad +1 \ +2 \ +3 \ +4 \ +5 \ +6 \ +7 \ +8 \\ G_2 = & \quad -8 \ -3 \ -2 \ -1 \ +5 \ +6 \ -7 \ +4 \end{aligned}$$

- Le signe des gènes est pris en compte.

Problématique

- **Entrée :** Deux génomes G_1 et G_2 **sans** duplication
- **Sortie :** Une distance entre G_1 et G_2

- plusieurs distances existent : *breakpoints*, *intervalles communs*, *intervalles conservés*, SAD ...

Définition :

- Une paire de gènes consécutifs (a, b) sur G_1 induit un *breakpoint* si ni (a, b) ni $(\overline{b}, \overline{a})$ n'est une paire de gènes consécutifs sur G_2, où $\overline{g}$ exprime le gène g de signe opposé.

$$G_1 = +0 \ +1 \ +2 \ +3 \ +4 \ +5 \ +6 \ +7 \ +8 \ +9$$
$$G_2 = +0 \ -8 \ -3 \ -2 \ -1 \ +5 \ +6 \ -7 \ +4 \ +9$$

- Le signe des gènes est pris en compte.

Problématique

- **Entrée :** Deux génomes G_1 et G_2 **sans** duplication
- **Sortie :** Une distance entre G_1 et G_2

- plusieurs distances existent : *breakpoints*, *intervalles communs*, *intervalles conservés*, SAD ...

Définition :

- Une paire de gènes consécutifs (a, b) sur G_1 induit un *breakpoint* si ni (a, b) ni $(\overline{b}, \overline{a})$ n'est une paire de gènes consécutifs sur G_2, où $\overline{g}$ exprime le gène g de signe opposé.

$$G_1 = +0^{\downarrow} +1 \ +2 \ +3^{\downarrow} +4^{\downarrow} +5 \ +6^{\downarrow} +7^{\downarrow} +8^{\downarrow} +9$$
$$G_2 = +0 \ -8 \ -3 \ -2 \ -1 \ +5 \ +6 \ -7 \ +4 \ +9$$

- Le signe des gènes est pris en compte.

Problématique

- **Entrée :** Deux génomes G_1 et G_2 **sans** duplication
- **Sortie :** Une distance entre G_1 et G_2

Définition :

- Un *intervalle* du génome G_1 est une séquence d'éléments consécutifs de G_1
- Un intervalle I_1 de G_1 est un *intervalle commun* de (G_1, G_2) s'il existe dans G_2 un intervalle I_2 tel que I_2 soit une permutation de I_1.

$$G_1 = +1 \ +2 \ +3 \ \text{+4 +5 +6 +7} \ +8$$
$$G_2 = -8 \ -3 \ -2 \ -1 \ \text{+5 +6 -7 +4}$$

- Les gènes ne sont pas signés

Problématique

- **Entrée :** Deux génomes G_1 et G_2 **avec** duplications
- **Sortie :** Une distance entre G_1 et G_2

Problématique

- **Entrée :** Deux génomes G_1 et G_2 **avec** duplications
- **Sortie :** Une distance entre G_1 et G_2

exemplarisation
[Sankoff, 99]

$-1\ 2\ \ -3\ 4\ \ 1\ \ \ -5$

$2\ \ -4\ 1\ \ -5\ 1\ 3\ \ 1$

Problématique

- **Entrée :** Deux génomes G_1 et G_2 **avec** duplications
- **Sortie :** Une distance entre G_1 et G_2

exemplarisation
[Sankoff, 99]

$-1\ 2\ -3\ 4\ 1\ -5$

$2\ -4\ 1\ -5\ 1\ 3\ 1$

Problématique

- **Entrée :** Deux génomes G_1 et G_2 **avec** duplications
- **Sortie :** Une distance entre G_1 et G_2

exemplarisation
[Sankoff, 99]

$-1\ 2\ -3\ 4\ 1\ -5$

$2\ -4\ 1\ -5\ 1\ 3\ 1$

matching maximal
[Tang & al, 03]

$-1\ 2\ -3\ 4\ 1\ -5$

$2\ -4\ 1\ -5\ 1\ 3\ 1$

Problématique

- **Entrée :** Deux génomes G_1 et G_2 **avec** duplications
- **Sortie :** Une distance entre G_1 et G_2

exemplarisation
[Sankoff, 99]

$-1\ 2\ -3\ 4\ 1\ -5$

$2\ -4\ 1\ -5\ 1\ 3\ 1$

matching maximal
[Tang & al, 03]

$-1\ 2\ -3\ 4\ 1\ -5$

$2\ -4\ 1\ -5\ 1\ 3\ 1$

Problématique

- **Entrée :** Deux génomes G_1 et G_2 **avec** duplications
- **Sortie :** Une distance entre G_1 et G_2

exemplarisation
[Sankoff, 99]

−1 2 − 3 4 1 − 5

2 − 4 1 − 5 1 3 1

matching non-maximal

matching maximal
[Tang & al, 03]

-1 2 − 3 4 1 − 5

2 − 4 1 − 5 1 3 1

Problématique

- **Entrée :** Deux génomes G_1 et G_2 **avec** duplications
- **Sortie :** Une distance entre G_1 et G_2

exemplarisation
[Sankoff, 99]

$-1\ 2\ -3\ 4\ 1\ -5$

$2\ -4\ 1\ -5\ 1\ 3\ 1$

matching non-maximal

matching maximal
[Tang & al, 03]

$-1\ 2\ -3\ 4\ 1\ -5$

$2\ -4\ 1\ -5\ 1\ 3\ 1$

- **Objectif :** Trouver une exemplarisation (ou un matching) optimisant la mesure
- **Complexité** : *breakpoint* et *intervalles communs*
  - Prouvé NP-Complet [Bryant, 00] [Chauve & al, 06] même si $occ(G_1) = 1$ et $occ(G_2) = 2$
  - Preuve d'APX-Difficulté

Plan

Algorithme exact : transformation

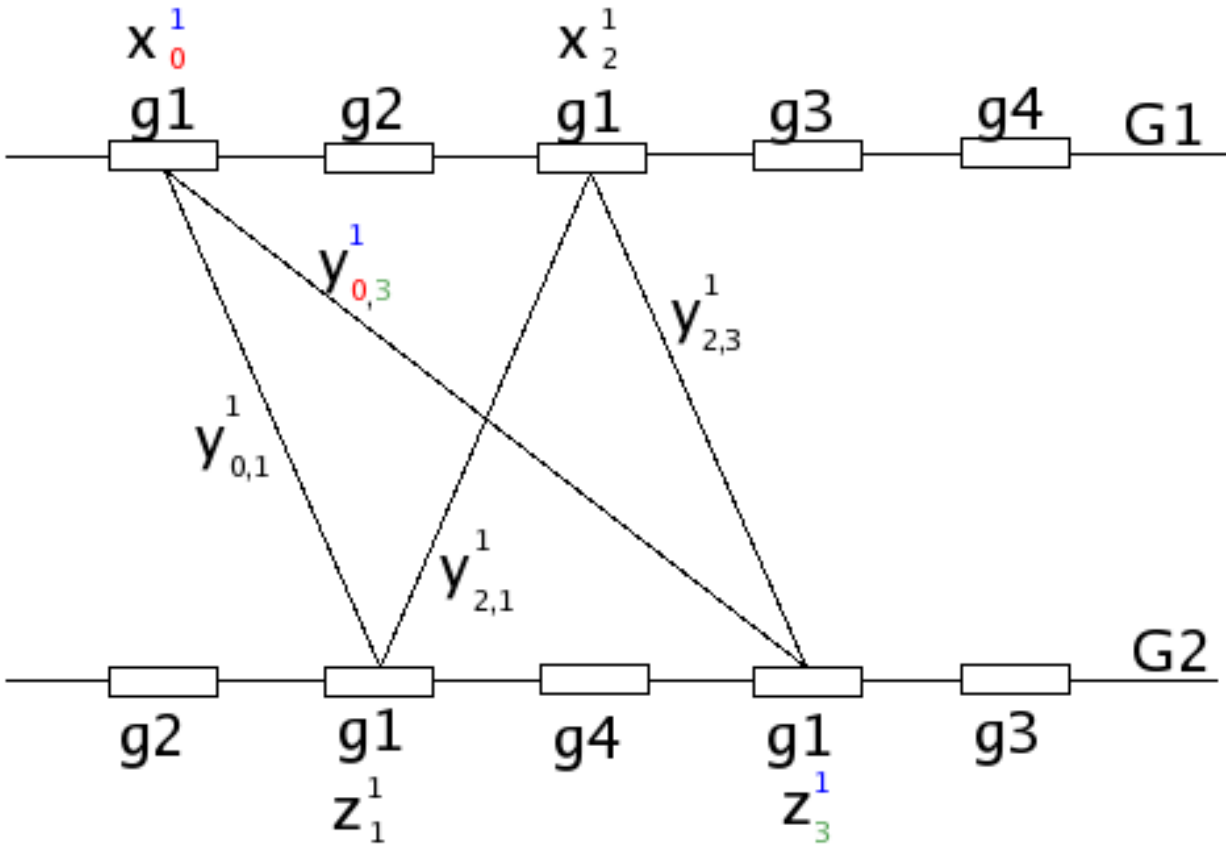
- Transformation en un problème **pseudo-booléen**
  Variables : booléennes
  Contraintes : inégalités entre sommes pondérees de variables
  Fonction Objectif : somme pondérée de variables

- Solvers puissants pour ce type de problème.
  Utilisation de `Minisat+`

Algorithme exact : transformation

Intervalles Communs :

- Variables x, y, z et I :



$I_{k,l,m,n}$ vrai $\Leftrightarrow$ $[k, l]$ dans G_1 est un intervalle commun de (G_1, G_2), et $[m, n]$ dans G_2 est une permutation de $[k, l]$

Algorithme exact : Contraintes

C1 : Une occurrence de chaque gène

$i \quad F, exactlyone(x^i_1, \ldots, x^i_{occ(i,G_1)})$

$i \quad F, exactlyone(z^i_1, \ldots, z^i_{occ(i,G_2)})$

Algorithme exact : Contraintes

C1 : Une occurrence de chaque gène

$i \quad F, exactlyone(x^i_1, \ldots, x^i_{occ(i,G_1)})$

$i \quad F, exactlyone(z^i_1, \ldots, z^i_{occ(i,G_2)})$

C2 : Validité des variables $l_{k,l,m,n}$

$[k,l] \quad G_1, \quad [m,n] \quad G_2, \quad x^i_j \quad [k,l],$

$$l_{klmn} \quad (\overline{x^i_j} \quad z^i_{\phi_1} \quad \ldots \quad z^i_{\phi_p})$$

avec $s \quad [1,p], z^i_{\phi_s} \quad [m,n]$

Algorithme exact : Contraintes

C1 : Une occurrence de chaque gène

$i \quad F, exactlyone(x^i_1, \ldots, x^i_{occ(i,G_1)})$

$i \quad F, exactlyone(z^i_1, \ldots, z^i_{occ(i,G_2)})$

C2 : Validité des variables $l_{k,l,m,n}$

$[k,l] \quad G_1, \quad [m,n] \quad G_2, \quad x^i_j \quad [k,l],$

$$l_{klmn} \quad (\overline{x^i_j} \quad z^i_{\phi_1} \quad \ldots \quad z^i_{\phi_p})$$

avec $s \quad [1,p], z^i_{\phi_s} \quad [m,n]$

C3 : Extrémités des intervalles

$[k,l] \quad G_1, \quad [m,n] \quad G_2,$

$l_{klmn} \quad x_k^{G_1[k]} \quad x_l^{G_1[l]} \quad z_m^{G_2[m]} \quad z_n^{G_2[n]}$

Algorithme exact : fonction objectif

Fonction objectif :

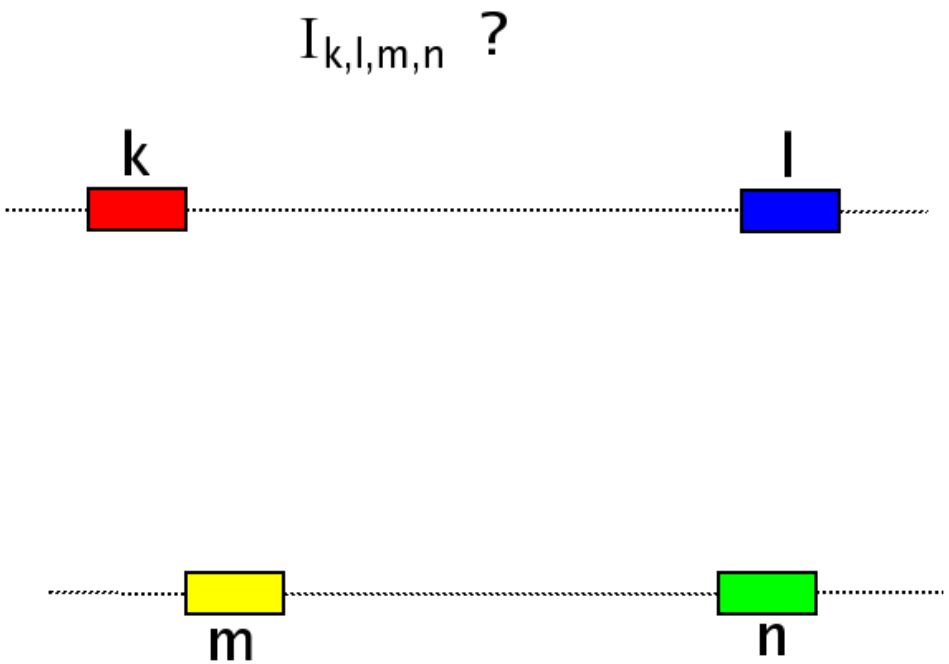
$$\text{MAX} \sum_{k,l,m,n} I_{k,l,m,n}$$

Algorithme exact : fonction objectif

Fonction objectif :

$$\text{MAX} \sum_{k,l,m,n} I_{k,l,m,n}$$

Exemple : modèle *exemplarisation*

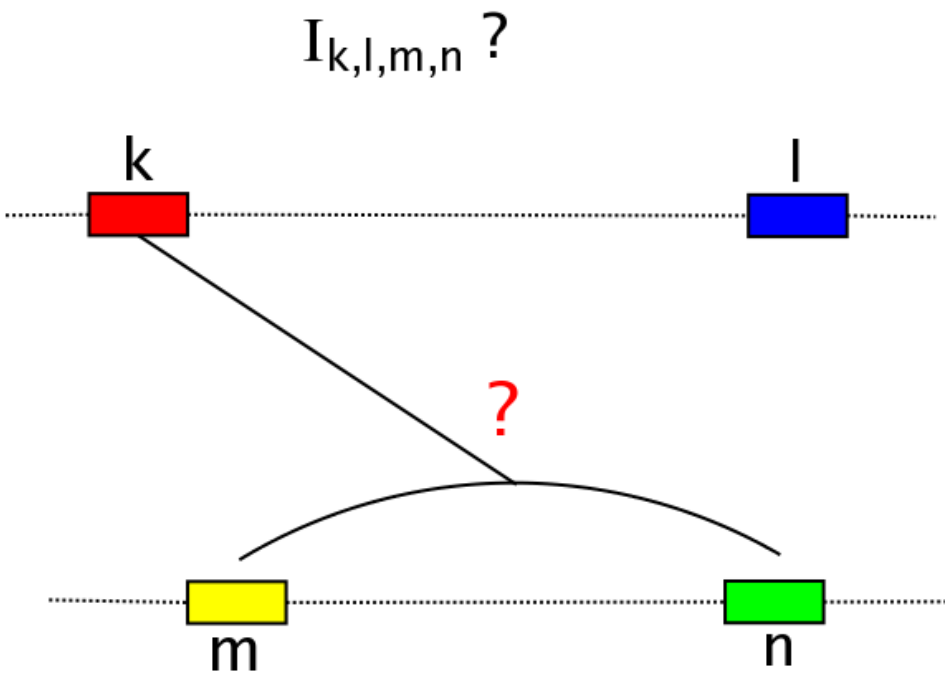


Algorithme exact : fonction objectif

Fonction objectif :

$$\text{MAX} \sum_{k,l,m,n} I_{k,l,m,n}$$

Exemple : modèle *exemplarisation*

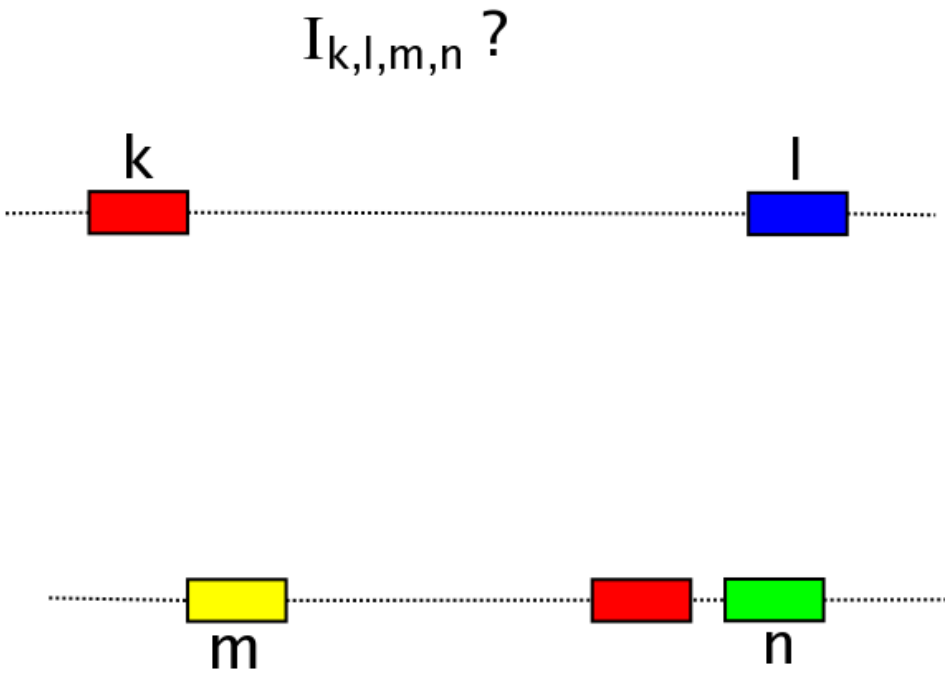


Algorithme exact : fonction objectif

Fonction objectif :

$$\text{MAX} \sum_{k,l,m,n} I_{k,l,m,n}$$

Exemple : modèle *exemplarisation*

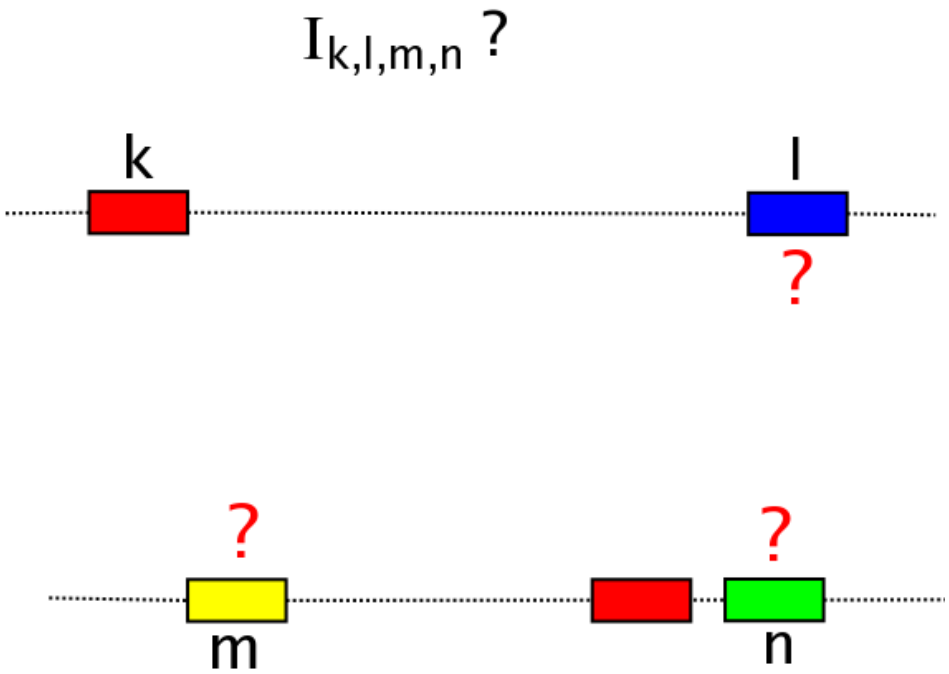


Algorithme exact : fonction objectif

Fonction objectif :

$$\text{MAX} \sum_{k,l,m,n} I_{k,l,m,n}$$

Exemple : modèle *exemplarisation*

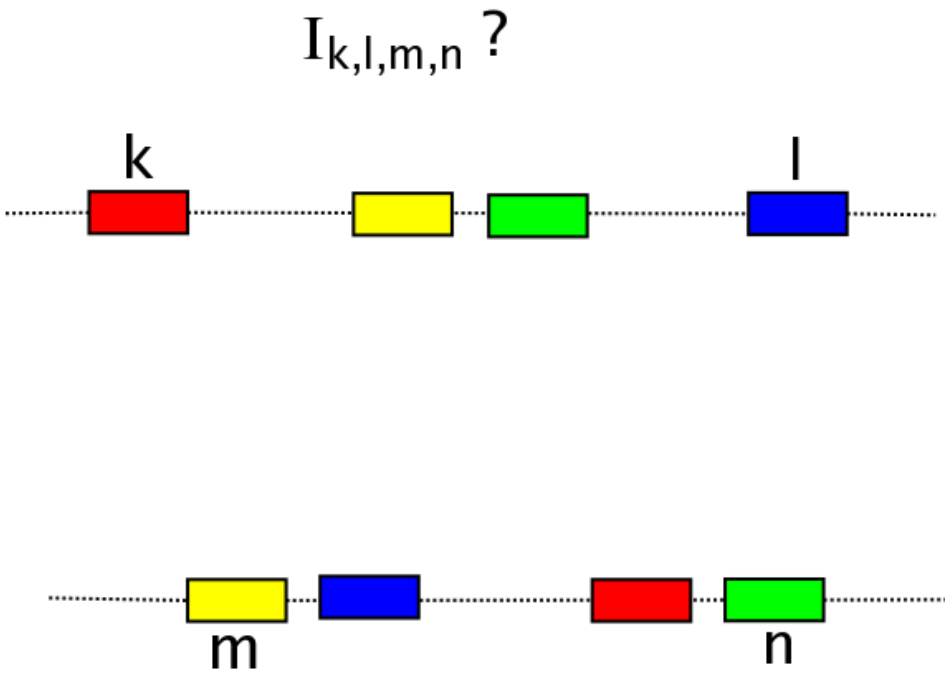


Algorithme exact : fonction objectif

Fonction objectif :

$$\text{MAX} \sum_{k,l,m,n} I_{k,l,m,n}$$

Exemple : modèle *exemplarisation*

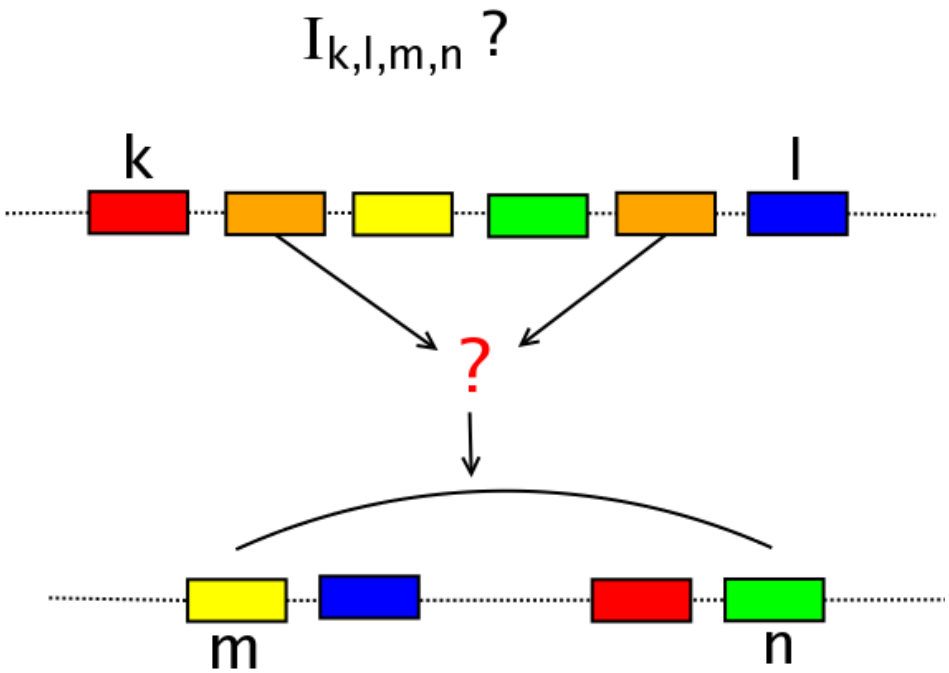


Algorithme exact : fonction objectif

Fonction objectif :

$$\text{MAX} \sum_{k,l,m,n} I_{k,l,m,n}$$

Exemple : modèle *exemplarisation*

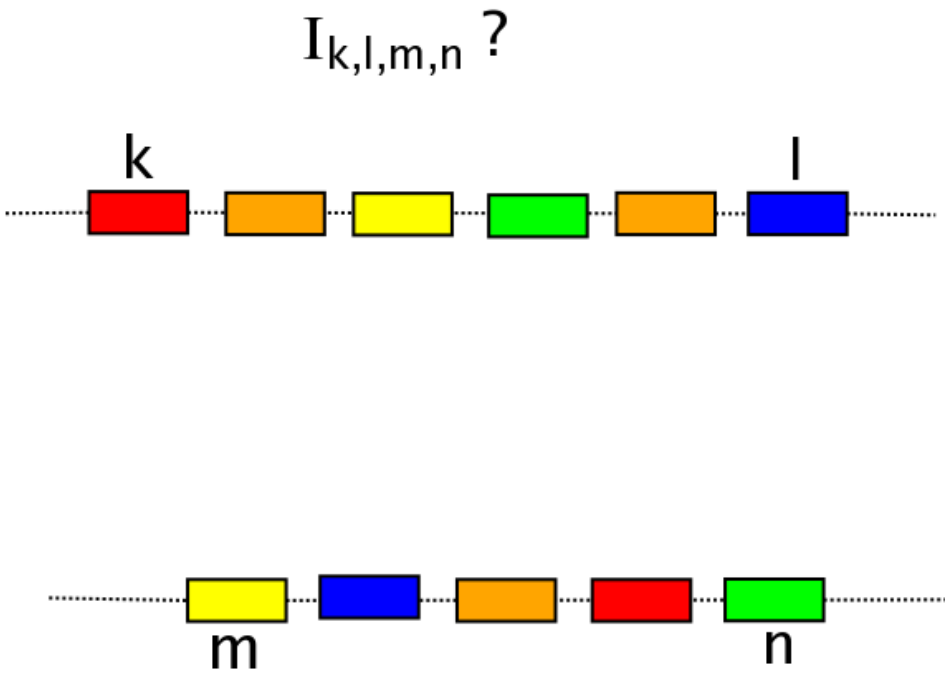


Algorithme exact : fonction objectif

Fonction objectif :

$$\text{MAX} \sum_{k,l,m,n} I_{k,l,m,n}$$

Exemple : modèle *exemplarisation*



Algorithme exact : résultats expérimentaux

- **Jeu de données :** 12 génomes de γ-*Proteobacteries*

Algorithme exact : résultats expérimentaux

- **Jeu de données :** 12 génomes de γ-*Proteobacteries*

Résultats : breakpoints

- modèle *exemplaire* :
  49 mesures exactes obtenues sur les 66 paires de génomes possibles
- modèle *matching maximal* :
  66 mesures exactes

Algorithme exact : résultats expérimentaux

- **Jeu de données :** 12 génomes de γ-*Proteobacteries*

Résultats : breakpoints

- modèle *exemplaire* :
  49 mesures exactes obtenues sur les 66 paires de génomes possibles
- modèle *matching maximal* :
  66 mesures exactes

Résultats : intervalles communs

- modèle *matching maximal*
  39 mesures exactes obtenues sur les 66 paires de génomes possibles
  - 23 distances obtenues en moins de 20 secondes
  - 14 distances obtenues en moins de 4 minutes
  - 2 distances obtenues en moins de 5 jours

Plan

Méthodes approchées : heuristique ILCS

LCS : Longest Common Substring [Marron & al, 04]

1 2 3 4 5 6 7

6 7 4 5 1 6 3 2 1

Méthodes approchées : heuristique ILCS

LCS : Longest Common Substring [Marron & al, 04]

1 2 3 4 5 6 7

6 7 4 5 1 6 3 2 1

Heuristique ILCS

Idée : Matcher les gènes des LCS jusqu'à saturation

Méthodes approchées : heuristique ILCS

LCS : Longest Common Substring [Marron & al, 04]

1 2 3 4 5 6 7

6 7 4 5 1 6 3 2 1

Heuristique ILCS

Idée : Matcher les gènes des LCS jusqu'à saturation

1. Calculer un LCS S

Méthodes approchées : heuristique ILCS

LCS : Longest Common Substring [Marron & al, 04]

1 2 3 4 5 6 7

6 7 4 5 1 6 3 2 1

Heuristique ILCS

Idée : Matcher les gènes des LCS jusqu'à saturation

1. Calculer un LCS S
2. Matcher tous les gènes de S

Méthodes approchées : heuristique ILCS

LCS : Longest Common Substring [Marron & al, 04]

1 2 3 4 5 6 7

6 7 4 5 1 6 3 2 1

Heuristique ILCS

Idée : Matcher les gènes des LCS jusqu'à saturation

1. Calculer un LCS S
2. Matcher tous les gènes de S
3. Recommencer jusqu'à saturation

Méthodes approchées : heuristique ILCS

LCS : Longest Common Substring [Marron & al, 04]

1 2 3 4 5 6 7

6 7 4 5 1 6 3 2 1

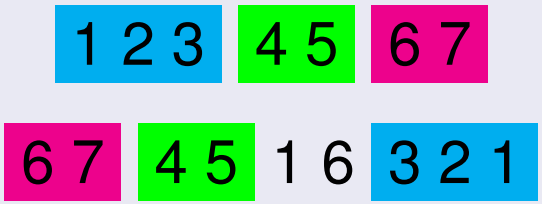
Heuristique ILCS

Idée : Matcher les gènes des LCS jusqu'à saturation

1. Calculer un LCS S
2. Matcher tous les gènes de S
3. Recommencer jusqu'à saturation

Méthodes approchées : heuristique ILCS

LCS : Longest Common Substring [Marron & al, 04]



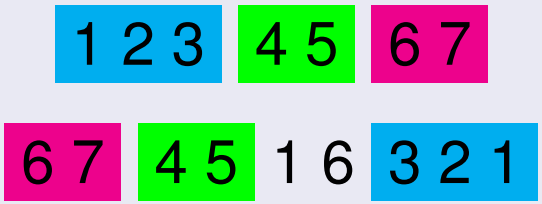
Heuristique ILCS

Idée : Matcher les gènes des LCS jusqu'à saturation

1. Calculer un LCS S
2. Matcher tous les gènes de S
3. Recommencer jusqu'à saturation

Méthodes approchées : heuristique ILCS

LCS : Longest Common Substring [Marron & al, 04]



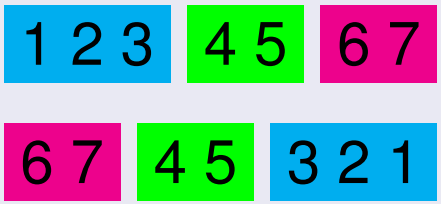
Heuristique ILCS

Idée : Matcher les gènes des LCS jusqu'à saturation

1. Calculer un LCS S
2. Matcher tous les gènes de S
3. Recommencer jusqu'à saturation
4. Retirer tous les gènes qui n'ont pas été matchés

Méthodes approchées : heuristique ILCS

LCS : Longest Common Substring [Marron & al, 04]



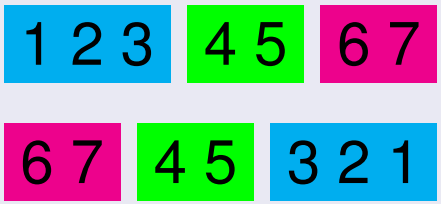
Heuristique ILCS

Idée : Matcher les gènes des LCS jusqu'à saturation

1. Calculer un LCS S
2. Matcher tous les gènes de S
3. Recommencer jusqu'à saturation
4. Retirer tous les gènes qui n'ont pas été matchés

Méthodes approchées : heuristique ILCS

LCS : Longest Common Substring [Marron & al, 04]



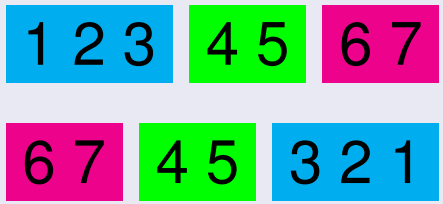
Heuristique ILCS

Idée : Matcher les gènes des LCS jusqu'à saturation

1. Calculer un LCS S
2. Matcher tous les gènes de S
3. Recommencer jusqu'à saturation
4. Retirer tous les gènes qui n'ont pas été matchés
5. Calculer la mesure

Méthodes approchées : heuristique ILCS

LCS : Longest Common Substring [Marron & al, 04]



nombre d'intervalles communs $= 19$

Heuristique ILCS

Idée : Matcher les gènes des LCS jusqu'à saturation

1. Calculer un LCS S
2. Matcher tous les gènes de S
3. Recommencer jusqu'à saturation
4. Retirer tous les gènes qui n'ont pas été matchés
5. Calculer la mesure

Méthodes approchées : heuristique IILCS

LCS : Longest Common Substring [Marron & al, 04]

1 2 3 4 5 6 7

6 7 4 5 1 6 3 2 1

Heuristique IILCS

Idée : Retirer les gènes ne pouvant plus être matchés

Méthodes approchées : heuristique IILCS

LCS : Longest Common Substring [Marron & al, 04]

1 2 3 4 5 6 7

6 7 4 5 1 6 3 2 1

Heuristique IILCS

Idée : Retirer les gènes ne pouvant plus être matchés

1. Calculer un LCS S

Méthodes approchées : heuristique IILCS

LCS : Longest Common Substring [Marron & al, 04]

1 2 3 4 5 6 7

6 7 4 5 1 6 3 2 1

Heuristique IILCS

Idée : Retirer les gènes ne pouvant plus être matchés

1. Calculer un LCS S

Méthodes approchées : heuristique IILCS

LCS : Longest Common Substring [Marron & al, 04]

1 2 3 4 5 6 7

6 7 4 5 1 6 3 2 1

Heuristique IILCS

Idée : Retirer les gènes ne pouvant plus être matchés

1. Calculer un LCS S
2. Matcher tous les gènes de S

Méthodes approchées : heuristique IILCS

LCS : Longest Common Substring [Marron & al, 04]

1 2 3 4 5 6 7

6 7 4 5 1 6 3 2 1

Heuristique IILCS

Idée : Retirer les gènes ne pouvant plus être matchés

1. Calculer un LCS S
2. Matcher tous les gènes de S
3. Retirer les gènes ne pouvant plus être matchés

Méthodes approchées : heuristique IILCS

LCS : Longest Common Substring [Marron & al, 04]

1 2 3 4 5 6 7

6 7 4 5 1 6 3 2 1

Heuristique IILCS

Idée : Retirer les gènes ne pouvant plus être matchés

1. Calculer un LCS S
2. Matcher tous les gènes de S
3. Retirer les gènes ne pouvant plus être matchés

Méthodes approchées : heuristique IILCS

LCS : Longest Common Substring [Marron & al, 04]

1 2 3 4 5 6 7

6 7 4 5 6 3 2 1

Heuristique IILCS

Idée : Retirer les gènes ne pouvant plus être matchés

1. Calculer un LCS S
2. Matcher tous les gènes de S
3. Retirer les gènes ne pouvant plus être matchés

Méthodes approchées : heuristique IILCS

LCS : Longest Common Substring [Marron & al, 04]

1 2 3 4 5 6 7

6 7 4 5 6 3 2 1

Heuristique IILCS

Idée : Retirer les gènes ne pouvant plus être matchés

1. Calculer un LCS S
2. Matcher tous les gènes de S
3. Retirer les gènes ne pouvant plus être matchés
4. Recommencer jusqu'à saturation

Méthodes approchées : heuristique IILCS

LCS : Longest Common Substring [Marron & al, 04]

1 2 3 4 5 6 7

6 7 4 5 6 3 2 1

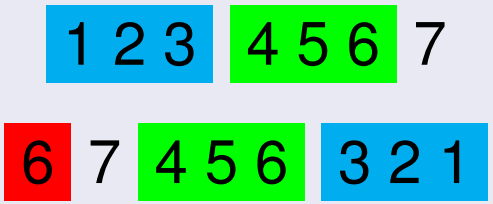
Heuristique IILCS

Idée : Retirer les gènes ne pouvant plus être matchés

1. Calculer un LCS S
2. Matcher tous les gènes de S
3. Retirer les gènes ne pouvant plus être matchés
4. Recommencer jusqu'à saturation

Méthodes approchées : heuristique IILCS

LCS : Longest Common Substring [Marron & al, 04]



Heuristique IILCS

Idée : Retirer les gènes ne pouvant plus être matchés

1. Calculer un LCS S
2. Matcher tous les gènes de S
3. Retirer les gènes ne pouvant plus être matchés
4. Recommencer jusqu'à saturation

Méthodes approchées : heuristique IILCS

LCS : Longest Common Substring [Marron & al, 04]

1 2 3 4 5 6 7

7 4 5 6 3 2 1

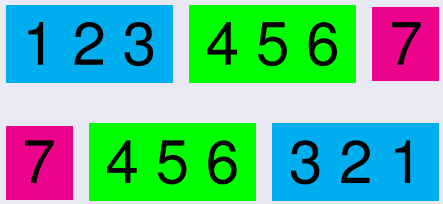
Heuristique IILCS

Idée : Retirer les gènes ne pouvant plus être matchés

1. Calculer un LCS S
2. Matcher tous les gènes de S
3. Retirer les gènes ne pouvant plus être matchés
4. Recommencer jusqu'à saturation

Méthodes approchées : heuristique IILCS

LCS : Longest Common Substring [Marron & al, 04]



Heuristique IILCS

Idée : Retirer les gènes ne pouvant plus être matchés

1. Calculer un LCS S
2. Matcher tous les gènes de S
3. Retirer les gènes ne pouvant plus être matchés
4. Recommencer jusqu'à saturation

Méthodes approchées : heuristique IILCS

LCS : Longest Common Substring [Marron & al, 04]

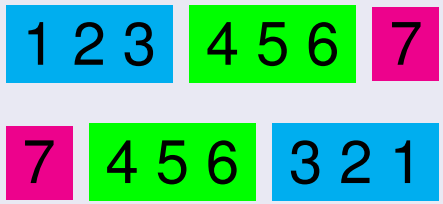
Heuristique IILCS

Idée : Retirer les gènes ne pouvant plus être matchés

1. Calculer un LCS S
2. Matcher tous les gènes de S
3. Retirer les gènes ne pouvant plus être matchés
4. Recommencer jusqu'à saturation
5. Calculer la mesure

Méthodes approchées : heuristique IILCS

LCS : Longest Common Substring [Marron & al, 04]



nombre d'intervalles communs $= 20$

Heuristique IILCS

Idée : Retirer les gènes ne pouvant plus être matchés

1. Calculer un LCS S
2. Matcher tous les gènes de S
3. Retirer les gènes ne pouvant plus être matchés
4. Recommencer jusqu'à saturation
5. Calculer la mesure

Méthodes approchées : méthode hybride

Algorithme $\texttt{HYB}_k$

- **Idée :** Associer la méthode exacte et l'heuristique `IILCS`
- **Paramètre** k **:** Borne de taille des **LCS**

1 Calculer un **LCS** S de (G_1, G_2)

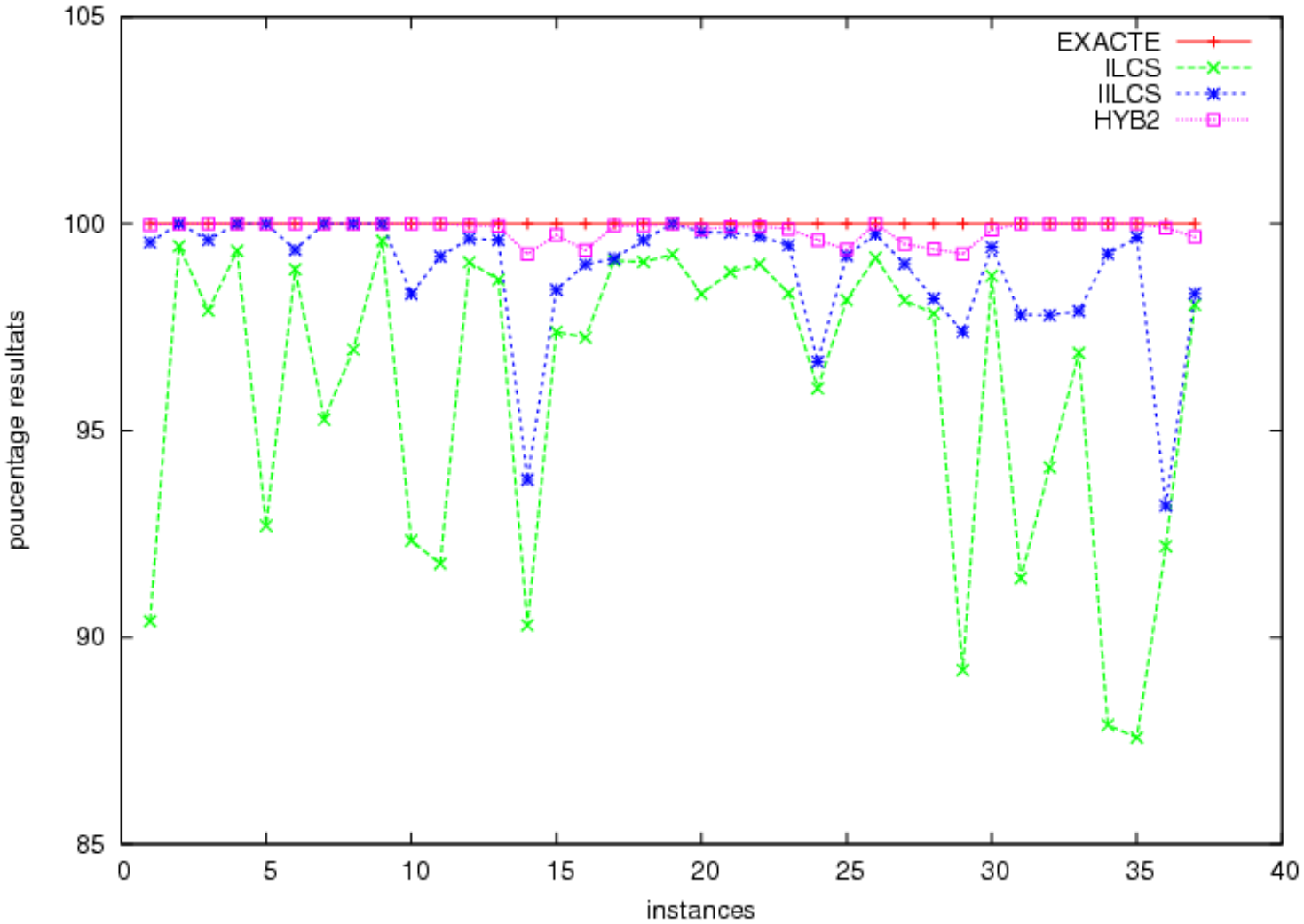
2 **Si** $|S| \geqslant k$
Alors

Matcher tous les gènes de S
Retirer les gènes ne pouvant plus être matchés
Recommencer en 1

Sinon Appliquer la méthode exacte

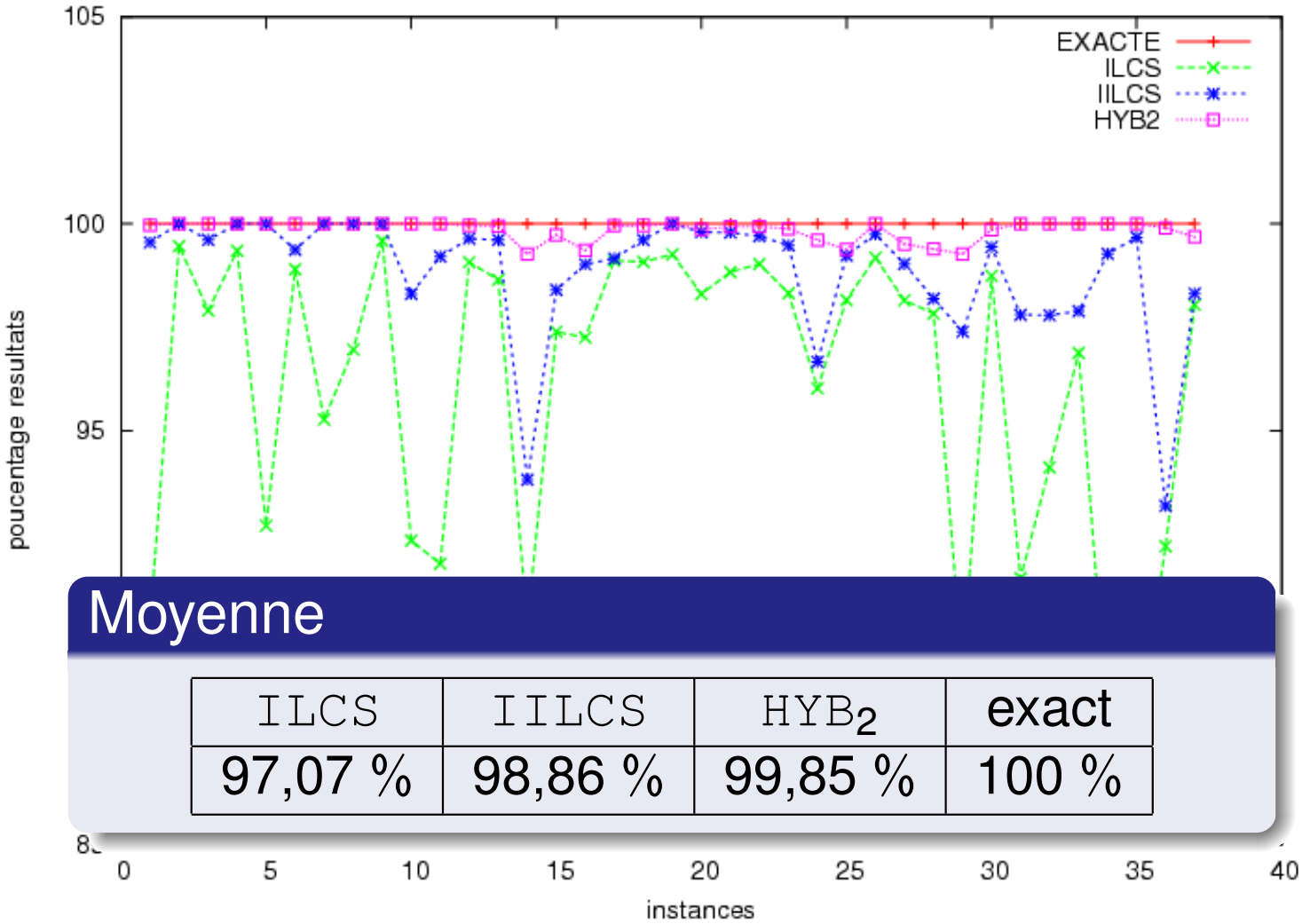
Intervalles communs : résultats expérimentaux

Résultats - modèle *matching maximal*



Intervalles communs : résultats expérimentaux

Résultats - modèle *matching maximal*

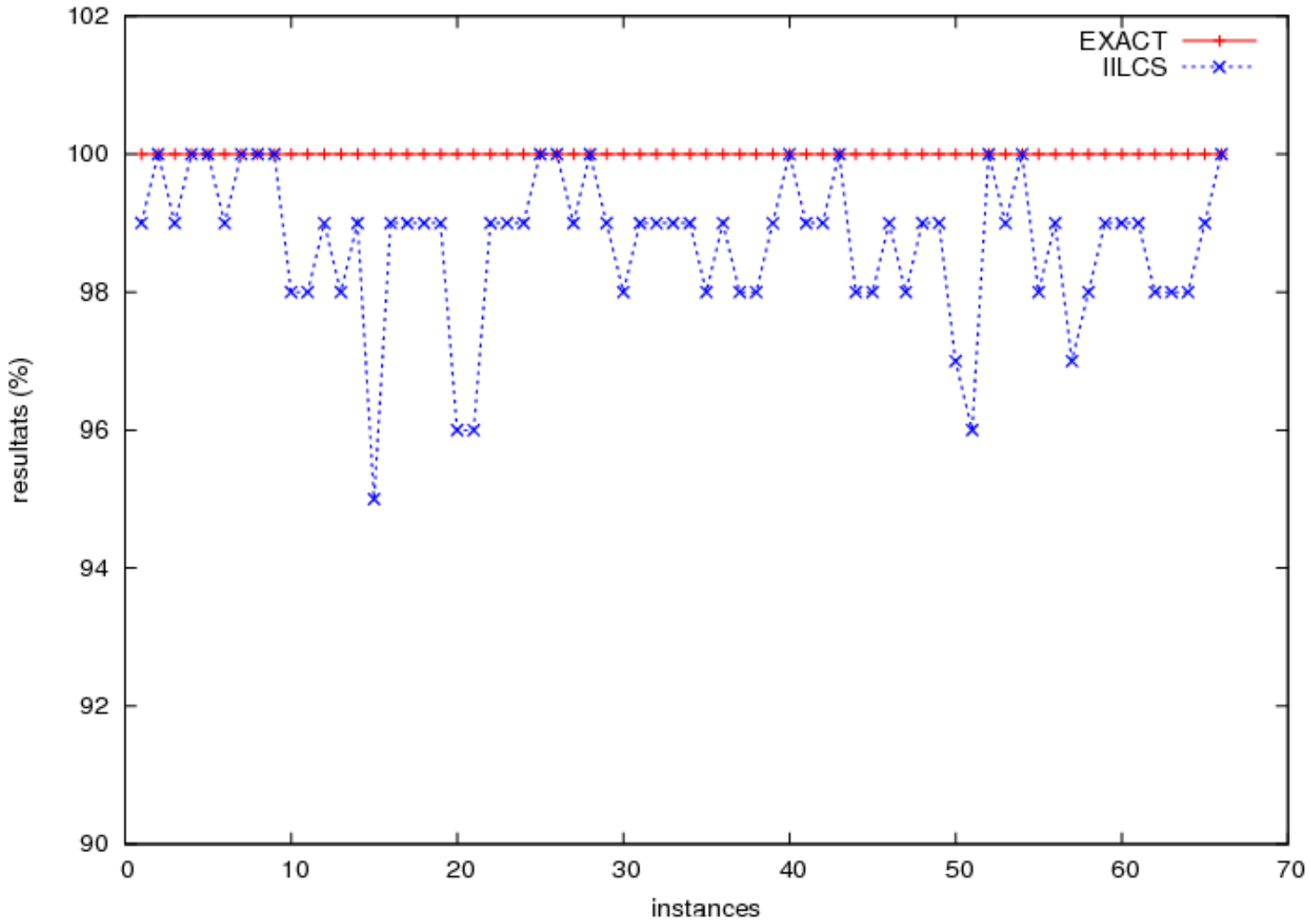


Moyenne

ILCS	IILCS	HYB_2	exact
97,07 %	98,86 %	99,85 %	100 %

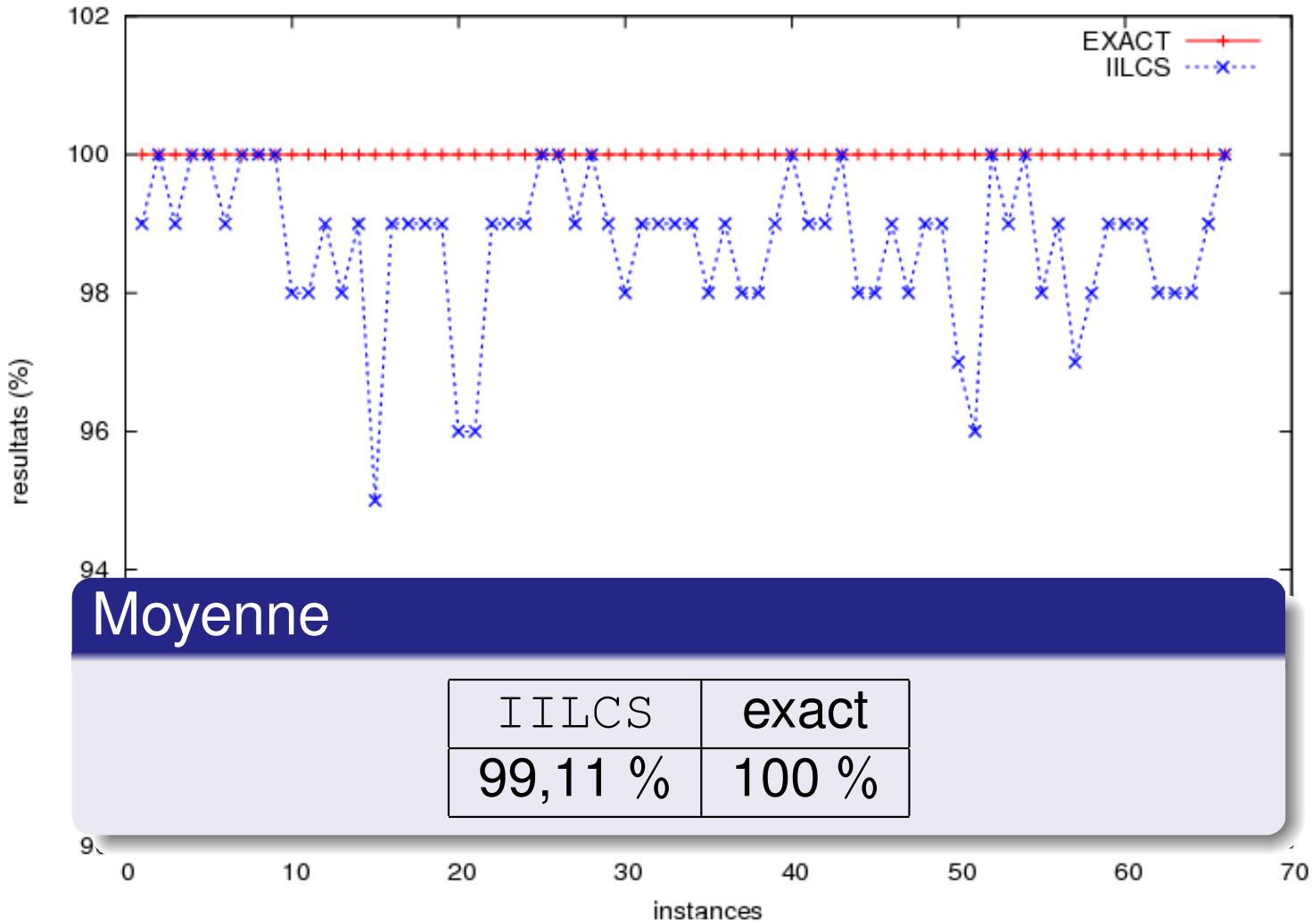
Breakpoints : résultats expérimentaux

Résultats - modèle *matching maximal*



Breakpoints : résultats expérimentaux

Résultats - modèle *matching maximal*

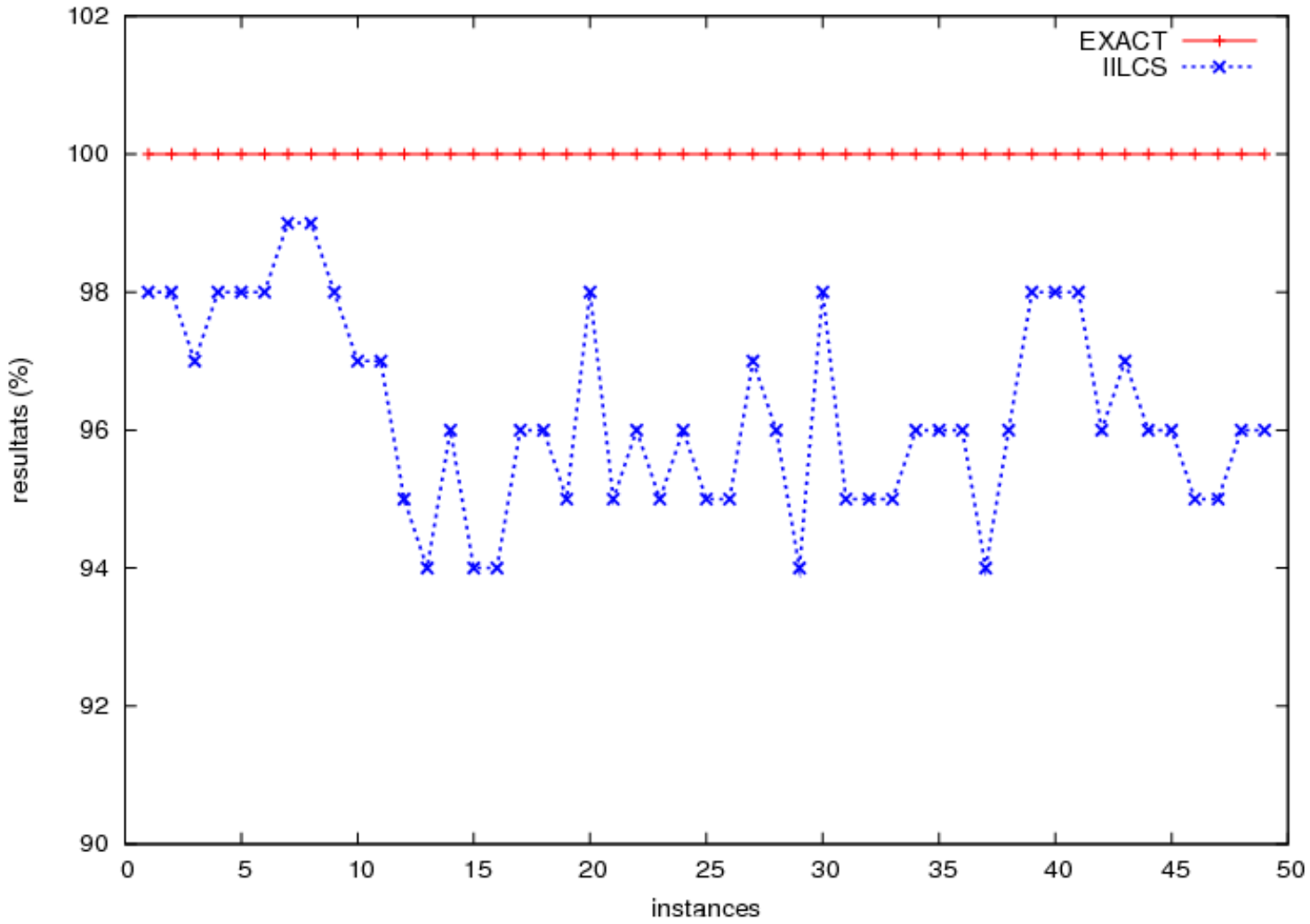


Moyenne

IILCS	exact
99,11 %	100 %

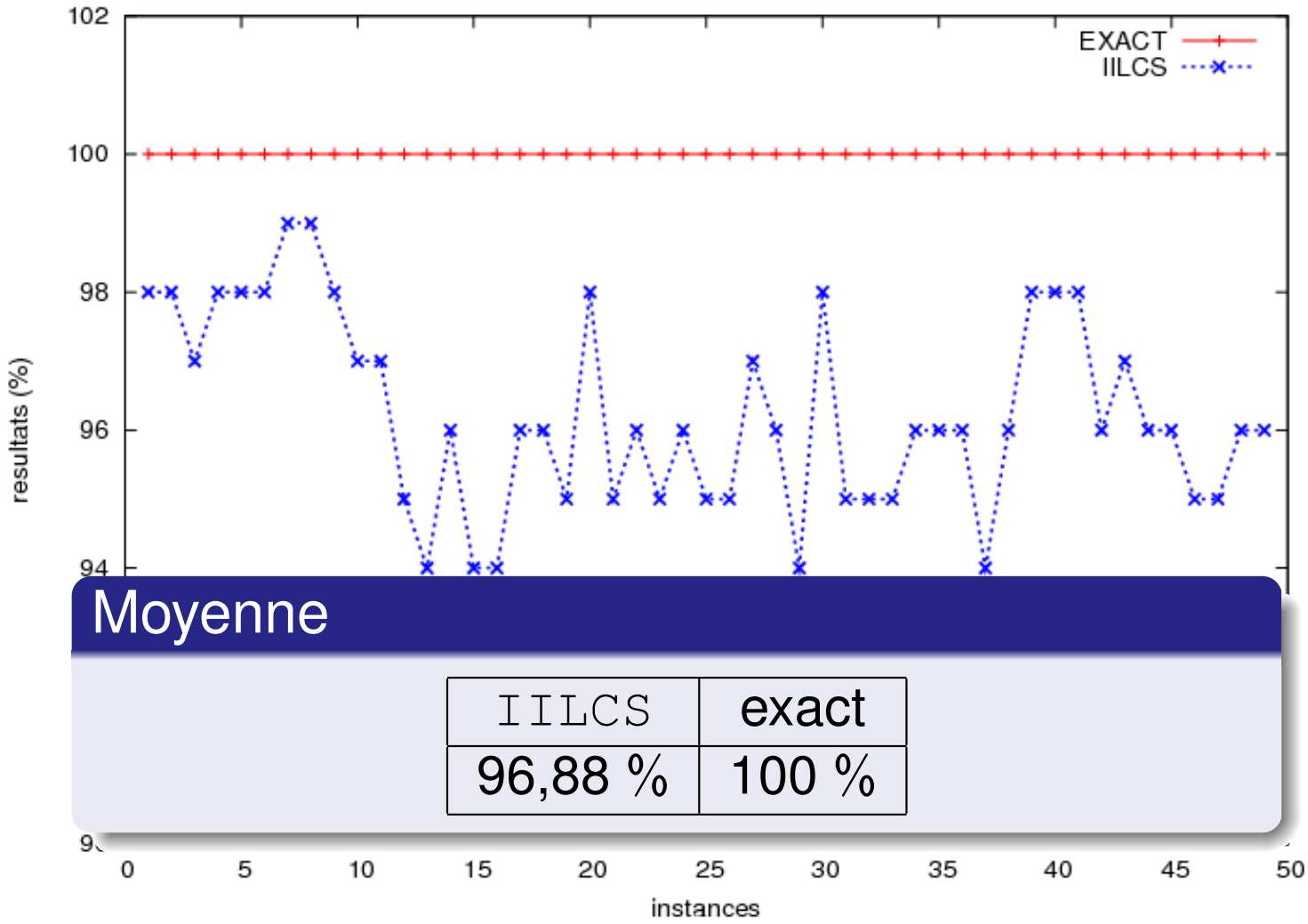
Breakpoints : résultats expérimentaux

Résultats - modèle *exemplaire*



Breakpoints : résultats expérimentaux

Résultats - modèle *exemplaire*

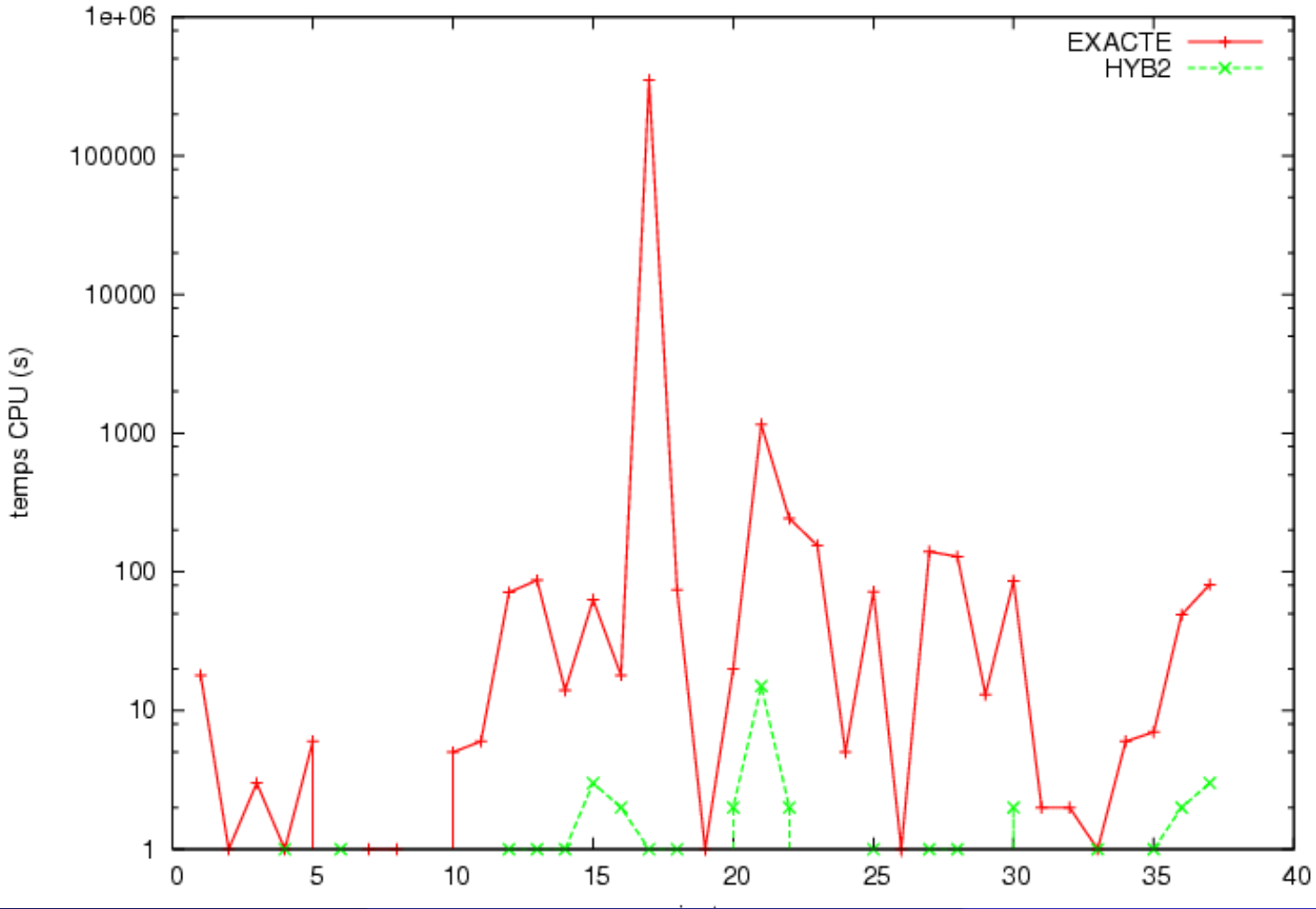


Moyenne

IILCS	exact
96,88 %	100 %

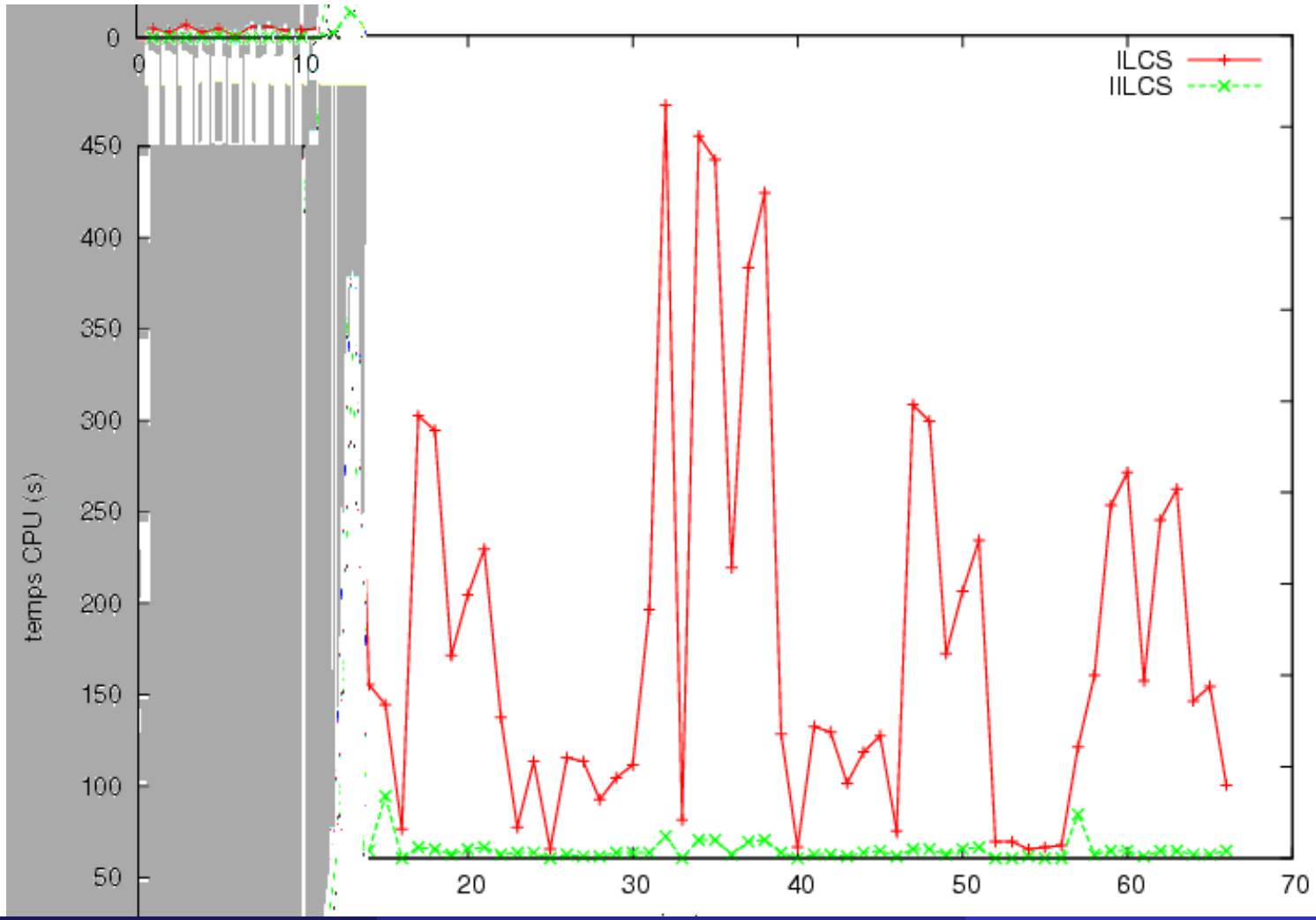
Méthodes approchées : résultats expérimentaux

Intervalles communs :
Temps d'exécution - modèle *matching maximal*



Méthodes approchées : résultats expérimentaux

Intervalles communs :
Temps d'exécution - modèle *matching maximal*



Plan

Perspectives

- **Analyse approfondie des résultats**
  Recherche d'une corrélation (Entrée - temps d'exécution)
  Compréhension du temps d'exécution pour le modèle *exemplaire*
- **Amélioration de la méthode exacte**
  - Ajout de règles afin de réduire la taille du problème
  - Reformulation de certaines contraintes
- **Expérimentations sur d'autres jeux de données**
- **Application de la méthode exacte pour d'autres distances**

Références

S. Angibaud, G. Fertin, I. Rusu, A. Thévenin, and S. Vialette.
A pseudo-boolean programming approach for computing the breakpoint distance between two genomes with duplicate genes.
In *Proc. RECOMB-CG 2007*, volume 4751 of *LNCS*, pages 16–29. Springer, 2007.

S. Angibaud, G. Fertin, I. Rusu, and S. Vialette.
A general framework for computing rearrangement distances between genomes with duplicates.
Journal of Computational Biology, 14(4) :379–393, 2007.

D. Bryant.
The complexity of calculating exemplar distances.
In *Comparative Genomics : Empirical and Analytical Approaches to Gene Order Dynamics, Map Alignement, and the Evolution of Gene Families*, pages 207–212. Kluwer Academic Publisher, 2000.

C. Chauve, G. Fertin, R. Rizzi, and S. Vialette.
Genomes containing duplicates are hard to compare.
In *Proc. IWBRA 2006*, volume 3992 of *LNCS*, pages 783–790. Springer, 2006.

M. Marron, K. M. Swenson, and B. M. E. Moret.
Genomic distances under deletions and insertions.
Theoretical Computer Science, 325(3) :347–360, 2004.

D. Sankoff.
Genome rearrangement with gene families.
Bioinformatics, 15(11) :909–917, 1999.

Approche pseudo-booléenne pour le calcul de distance entre génomes avec duplications

S. Angibaud[1] G. Fertin[1] I. Rusu[1]
A. Thévenin[2] S. Vialette[2]

[1]Laboratoire d'Informatique de Nantes-Atlantique (LINA), Nantes

[2]Laboratoire de Recherche en Informatique (LRI), Orsay

28 septembre 2007, Marne-la-Vallée