Comparaisons de génomes avec gènes dupliqués : étude théorique et algorithmes

Comparative genomics with duplicated genes: theoretical study and algorithms

Angibaud Sébastien

sebastien.angibaud@univ-nantes.fr

Laboratoire d'**I**nformatique de **N**antes **A**tlantique,
UMR CNRS 6241, UFR de Sciences et Techniques de Nantes

www.cnrs.fr

October 7th 2009

Outline

1. Genomes comparison
    - Overview
    - Genomes representation
    - Measures between genomes

Outline

1. Genomes comparison
   - Overview
   - Genomes representation
   - Measures between genomes
2. Theoretical complexity results

Outline

1. Genomes comparison
   - Overview
   - Genomes representation
   - Measures between genomes
2. Theoretical complexity results
3. Algorithms
   - Exact approach
   - Heuristics and hybrid method

Outline

1. Genomes comparison
    - Overview
    - Genomes representation
    - Measures between genomes
2. Theoretical complexity results
3. Algorithms
    - Exact approach
    - Heuristics and hybrid method
4. MATCH&WATCH application
    - Protocol
    - Visualization tool

Outline

1. Genomes comparison
    - Overview
    - Genomes representation
    - Measures between genomes
2. Theoretical complexity results
3. Algorithms
    - Exact approach
    - Heuristics and hybrid method
4. MATCH&WATCH application
    - Protocol
    - Visualization tool
5. Conclusion

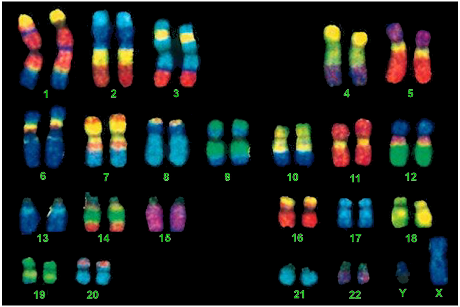
Outline

1. Genomes comparison
   - Overview
   - Genomes representation
   - Measures between genomes
2. Theoretical complexity results
3. Algorithms
   - Exact approach
   - Heuristics and hybrid method
4. MATCH&WATCH application
   - Protocol
   - Visualization tool
5. Conclusion

Genomes and genes

Genome:

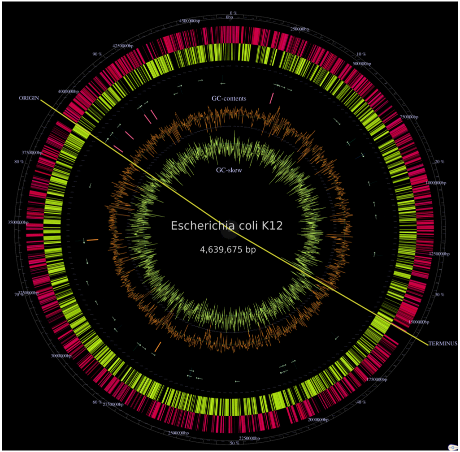
- Composed of one or several *chromosomes*



Genomes and genes

Genome:

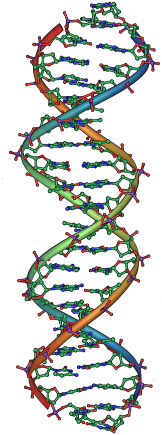
- Composed of one or several *chromosomes*



Genomes and genes

Genome:

- Composed of one or several *chromosomes*
- Sequence(s) of *DNA*
- Hereditary information



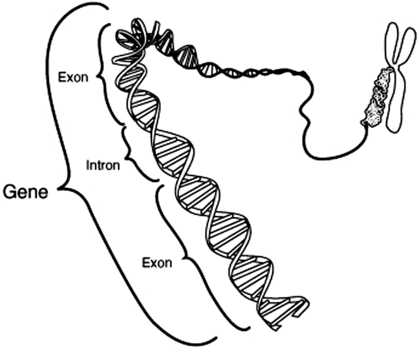
Genomes and genes

Genome:

- Composed of one or several *chromosomes*
- Sequence(s) of *DNA*
- Hereditary information

Gene:

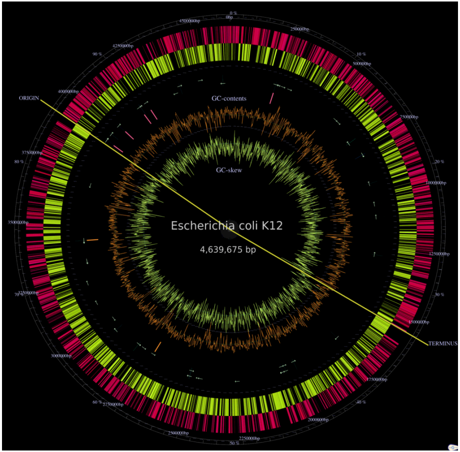
- Sequence of DNA
- Coding one or severals *proteins*
- Gene orientation



Genomes and genes

Genome:

- Composed of one or several *chromosomes*
- Sequence(s) of *DNA*
- Hereditary information



Gene:

- Sequence of DNA
- Coding one or severals *proteins*
- Gene orientation

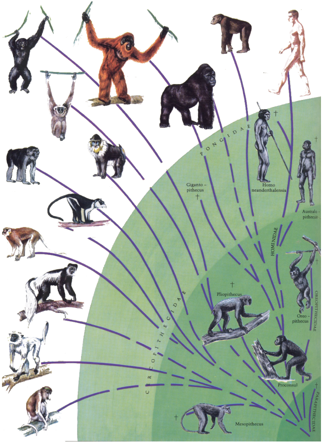
Comparing genomes

Why?

Comparing genomes

Why?

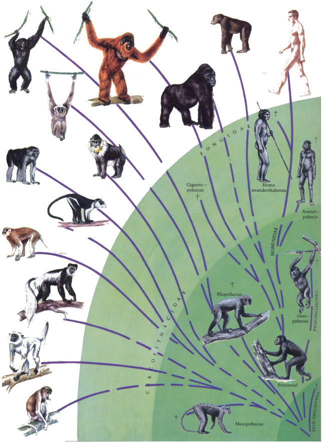
- Phylogenetic trees construction



Comparing genomes

Why?

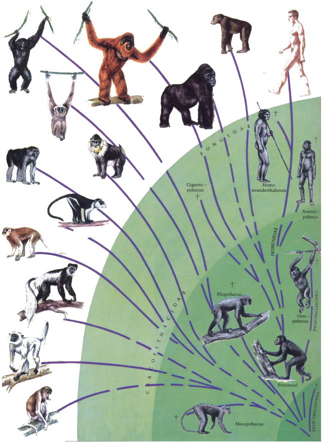
- Phylogenetic trees construction
- Identification of highly conserved sequences



Comparing genomes

Why?

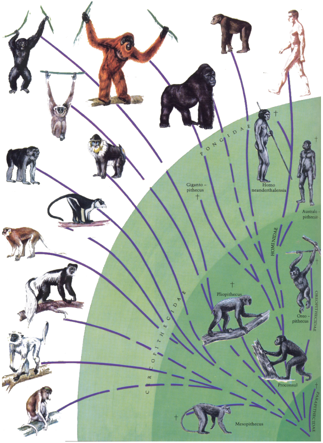
- Phylogenetic trees construction
- Identification of highly conserved sequences
- Help genome annotation



Comparing genomes

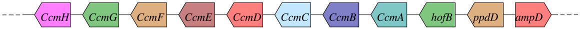
Why?

- Phylogenetic trees construction
- Identification of highly conserved sequences
- Help genome annotation



How?

- Genome modeled as a sequence of genes



Comparing two genomes : two different points of view

Comparison based on the evolution process

- Infer an evolution process from one genome to another
- Several operations can be considered:
  - inversion
  - translocation
  - duplication
  - . . .
- Find a most parsimonious rearrangement scenario

Comparing two genomes : two different points of view

Comparison based on the evolution process

- Infer an evolution process from one genome to another
- Several operations can be considered:
  - inversion
  - translocation
  - duplication
  - ...
- Find a most parsimonious rearrangement scenario

Comparison based on the structure of genomes

- Compare the structure (genes order) of the two genomes
- Compute a (dis)similarity measure between genomes
  - *number of breakpoints/adjacencies*
  - *number of common intervals*
  - *number of conserved intervals*
  - *Sum Adjacency Disruption*
  - ...

Comparing two genomes : two different points of view

Comparison based on the evolution process

- Infer an evolution process from one genome to another
- Several operations can be considered:
  - inversion
  - translocation
  - duplication
  - . . .
- Find a most parsimonious rearrangement scenario

Comparison based on the structure of genomes

- Compare the structure (genes order) of the two genomes
- Compute a (dis)similarity measure between genomes
  - *number of breakpoints/adjacencies*
  - *number of common intervals*
  - *number of conserved intervals*
  - *Sum Adjacency Disruption*
  - . . .

Genomes representation

Representation and notations

1. Unichromosomal genome: sequence of *signed genes*

Example

1. $\boldsymbol{G_0 = +1\ \ +2\ \ -3\ \ -7\ \ +4\ \ +5\ \ +7\ \ -8\ \ +10\ \ -9\ \ +4\ \ -6\ \ -4}$

Genomes representation

Representation and notations

1. Unichromosomal genome: sequence of *signed genes*
2. Alphabet $\boldsymbol{\Sigma} \Leftrightarrow$ *gene families*

Example

1. $G_0 = +1\ \ +2\ \ -3\ \ \text{-7}\ \ +4\ \ +5\ \ \text{+7}\ \ -8\ \ +10\ \ -9\ \ +4\ \ -6\ \ -4$
2. $\boldsymbol{\Sigma} = \{1, 2, 3 \ldots 10\}$

Genomes representation

Representation and notations

1. Unichromosomal genome: sequence of *signed genes*
2. Alphabet $\Sigma \Leftrightarrow$ *gene families*
3. Let $G_0[k]$ be the k^{th} gene (signed integer) of G_0

Example

1. $G_0 = +1\ +2\ -3\ \boxed{-7}\ +4\ +5\ +7\ -8\ +10\ -9\ +4\ -6\ -4$
2. $\Sigma = \{1, 2, 3 \ldots 10\}$
3. $G_0[4] = -7$

Genomes representation

Representation and notations

1. Unichromosomal genome: sequence of *signed genes*
2. Alphabet $\boldsymbol{\Sigma} \Leftrightarrow$ *gene families*
3. Let $G_0[k]$ be the k^{th} gene (signed integer) of G_0
4. Let $occ(G_0)$ be the maximum number of genes in a gene family

Example

1. $G_0 = +1\ +2\ -3\ -7\ \boxed{+4}\ +5\ +7\ -8\ +10\ -9\ \boxed{+4}\ -6\ \boxed{-4}$
2. $\Sigma = \{1, 2, 3 \ldots 10\}$
3. $G_0[4] = -7$
4. $occ(G_0) = 3$

Genomes representation

Representation and notations

1. Unichromosomal genome: sequence of *signed genes*
2. Alphabet $\boldsymbol{\Sigma}$ ⇔ *gene families*
3. Let $\boldsymbol{G_0[k]}$ be the $\boldsymbol{k^{th}}$ gene (signed integer) of $\boldsymbol{G_0}$
4. Let $\boldsymbol{occ(G_0)}$ be the maximum number of genes in a gene family
5. Let η_{G_0} be the number of genes in $\boldsymbol{G_0}$

Example

1. $\boldsymbol{G_0 = +1\ +2\ -3\ -7\ +4\ +5\ +7\ -8\ +10\ -9\ +4\ -6\ -4}$
2. $\boldsymbol{\Sigma = \{1, 2, 3 \ldots 10\}}$
3. $\boldsymbol{G_0[4] = -7}$
4. $\boldsymbol{occ(G_0) = 3}$
5. $\boldsymbol{\eta_{G_0} = 13}$

Measures between two genomes

- **Input:** Two genomes G_0 and G_1 with the same gene contents and without duplicates
- **Output:** A (dis)-similarity measure between G_0 and G_1

- *number of breakpoints/adjacencies* [Watterson et al. 1982]
- *number of common intervals* [Uno and Yagiura, 2000]
- *number of conserved intervals* [Bergeron and Stoye, 2003]

Breakpoint and adjacency

Definition: **adjacency** and **breakpoint** [Watterson et al. 1982]

There exists an adjacency between genes $\boldsymbol{G_0[p]}$ and $\boldsymbol{G_0[p+1]}$ **iff** $(\boldsymbol{G_0[p], G_0[p+1]})$ or $(\boldsymbol{-G_0[p+1], -G_0[p]})$ appears as a pair of consecutive genes in $\boldsymbol{G_1}$.

$$\boldsymbol{G_0 = +1\ \ +2\ \ +3\ \ +4\ \ +5}$$
$$\boldsymbol{G_1 = +3\ \ +4\ \ -5\ \ -2\ \ -1}$$

Breakpoint and adjacency

Definition: **adjacency** and **breakpoint** [Watterson et al. 1982]

There exists an adjacency between genes $\mathbf{G_0[p]}$ and $\mathbf{G_0[p+1]}$ **iff** $(\mathbf{G_0[p], G_0[p+1]})$ or $(-\mathbf{G_0[p+1], -G_0[p]})$ appears as a pair of consecutive genes in $\mathbf{G_1}$.

$$G_0 = \overbrace{+1 \ \ +2}^{\textit{Adjacency}} +3 \ \ +4 \ \ +5$$
$$G_1 = -4 \ \ -3 \ \ -5 \ +1 \ \ +2$$

Breakpoint and adjacency

Definition: **adjacency** and **breakpoint** [Watterson et al. 1982]

There exists an adjacency between genes $\boldsymbol{G_0[p]}$ and $\boldsymbol{G_0[p+1]}$ **iff** $(\boldsymbol{G_0[p], G_0[p+1]})$ or $(\boldsymbol{-G_0[p+1], -G_0[p]})$ appears as a pair of consecutive genes in $\boldsymbol{G_1}$.

$$\boldsymbol{G_0 = \overbrace{+1\ \ +2}^{\textit{Adjacency}}\ \overbrace{+3\ \ +4}^{\textit{Adjacency}}\ +5}$$
$$\boldsymbol{G_1 = -4\ \ -3\ \ -5\ \ +1\ \ +2}$$

Breakpoint and adjacency

Definition: adjacency and breakpoint [Watterson et al. 1982]

There exists a breakpoint between genes $\boldsymbol{G_0[p]}$ and $\boldsymbol{G_0[p+1]}$ **iff** neither $(\boldsymbol{G_0[p]}, \boldsymbol{G_0[p+1]})$ nor $(-\boldsymbol{G_0[p+1]}, -\boldsymbol{G_0[p]})$ appears as a pair of consecutive genes in $\boldsymbol{G_1}$.

$$\begin{aligned} G_0 &= \overbrace{+1\ \ +2}^{\textit{Adjacency}}\ \blacktriangledown \overbrace{+3\ \ +4}^{\textit{Adjacency}}\ \blacktriangledown\ +5 \\ G_1 &= -4\ \ -3\ \ -5\ \ +1\ \ +2 \end{aligned}$$

Breakpoint and adjacency

Definition: adjacency and breakpoint [Watterson et al. 1982]

There exists a breakpoint between genes $\boldsymbol{G_0[p]}$ and $\boldsymbol{G_0[p+1]}$ **iff** neither $(\boldsymbol{G_0[p], G_0[p+1]})$ nor $(\boldsymbol{-G_0[p+1], -G_0[p]})$ appears as a pair of consecutive genes in $\boldsymbol{G_1}$.

$$\begin{aligned} G_0 &= +0^{\blacktriangledown}\overbrace{+1\ \ +2}^{\textit{Adjacency}}{}^{\blacktriangledown}\overbrace{+3\ \ +4}^{\textit{Adjacency}}{}^{\blacktriangledown}+5^{\blacktriangledown}+6 \\ G_1 &= +0\ \ -4\ \ -3\ \ -5\ \ +1\ \ +2\ +6 \end{aligned}$$

Breakpoint and adjacency

Definition: **adjacency** and **breakpoint** [Watterson et al. 1982]

There exists a breakpoint between genes $\boldsymbol{G_0[p]}$ and $\boldsymbol{G_0[p+1]}$ **iff** neither $(\boldsymbol{G_0[p]}, \boldsymbol{G_0[p+1]})$ nor $(-\boldsymbol{G_0[p+1]}, -\boldsymbol{G_0[p]})$ appears as a pair of consecutive genes in $\boldsymbol{G_1}$.

$$\boldsymbol{G_0 = +0^{\blacktriangledown}\overbrace{+1\ \ +2}^{\textit{Adjacency}}{}^{\blacktriangledown}\overbrace{+3\ \ +4}^{\textit{Adjacency}}{}^{\blacktriangledown} +5^{\blacktriangledown}+6}$$
$$\boldsymbol{G_1 = +0\ \ -4\ \ -3\ \ -5\ \ +1\ \ +2\ \ +6}$$

Two measures:

- *Number of adjacencies:* similarity
- *Number of breakpoints:* dissimilarity

Common interval

Definition: **common interval** [Uno and Yagiura, 2000]

- A substring s_0 of G_0 is a *common interval* of (G_0, G_1) if, in G_1, there is a substring s_1 such that s_1 is a permutation of s_0 (without taking signs into account)

$$G_0 = +1 + 2 + 3 + 4 + 5 \qquad G_1 = +2 - 4 + 3 + 5 + 1$$

Common interval

Definition: **common interval** [Uno and Yagiura, 2000]

- A substring s_0 of G_0 is a *common interval* of (G_0, G_1) if, in G_1, there is a substring s_1 such that s_1 is a permutation of s_0 (without taking signs into account)

$$G_0 = +1 + 2 \text{ +3 +4 +5} \qquad G_1 = +2 \text{ -4 +3 +5} + 1$$

$\Rightarrow s_0 = +3 + 4 + 5 \quad s_1 = -4 + 3 + 5$
Substring s_0 is a common interval of (G_0, G_1).

Common interval

Definition: **common interval** [Uno and Yagiura, 2000]

- A substring s_0 of G_0 is a *common interval* of (G_0, G_1) if, in G_1, there is a substring s_1 such that s_1 is a permutation of s_0 (without taking signs into account)

$$G_0 = +1 + 2 \; \text{+3 +4 +5} \qquad G_1 = +2 \; \text{-4 +3 +5} \; + 1$$

$\Rightarrow s_0 = +3 + 4 + 5 \quad s_1 = -4 + 3 + 5$
Substring s_0 is a common interval of (G_0, G_1).

- **Number of common intervals of (G_0, G_1):**
Similarity measure between two genomes

Conserved interval

Definition: **conserved interval**

Proposed in [Bergeron and Stoye, 2003] for n permutations

- common interval
- same extremities OR reversed extremities

$$G_0 = +0 + 1 + 2 + 3 + 4 + 5$$

$$G_1 = -4 - 3 - 5 + 0 - 1 + 2$$

Conserved interval

Definition: conserved interval
Proposed in [Bergeron and Stoye, 2003] for n permutations

- common interval
- same extremities OR reversed extremities

$$G_0 = +0\ +1\ +2 + \mathbf{3} + \mathbf{4} + \mathbf{5}$$

$$G_1 = -\mathbf{4} - \mathbf{3} - \mathbf{5}\ +0\ -1 +2$$

Conserved interval

Definition: **conserved interval**
Proposed in [Bergeron and Stoye, 2003] for n permutations

- common interval
- same extremities OR reversed extremities

$$G_0 = +0 + 1 + 2 \; +3 \; +4 \; + 5$$

$$G_1 = \; -4 \; -3 \; - 5 + 0 - 1 + 2$$

Conserved interval

Definition: **conserved interval**
Proposed in [Bergeron and Stoye, 2003] for n permutations

- common interval
- same extremities OR reversed extremities

$$G_0 = +0 + 1 + 2 + 3 + 4 + 5$$

$$G_1 = -4 - 3 - 5 + 0 - 1 + 2$$

- **Number of conserved intervals of (G_0, G_1):**
  Similarity measure between two genomes

And with duplicates?

1. Choose a one-to-one correspondence $\mathcal{M}$ of genes (a matching)
2. Rename or remove genes according to $\mathcal{M}$
3. Compute the (dis)-similarity measure

And with duplicates?

1. Choose a one-to-one correspondence $\mathcal{M}$ of genes (a matching)
2. Rename or remove genes according to $\mathcal{M}$
3. Compute the (dis)-similarity measure

*exemplar model (**E**)*
[Sankoff, 99]
one occurrence for each
gene family in $\mathcal{M}$

$$\boldsymbol{G_0 = +0\;+1\;-2\;-1\;-3\;+4}$$

$$\boldsymbol{G_1 = +0\;-1\;+2\;-1\;-3\;-1\;+4}$$

And with duplicates?

1. Choose a one-to-one correspondence $\mathcal{M}$ of genes (a matching)
2. Rename or remove genes according to $\mathcal{M}$
3. Compute the (dis)-similarity measure

*exemplar model (**E**)*
[Sankoff, 99]
one occurrence for each
gene family in $\mathcal{M}$

$\boldsymbol{G_0 = +0\ \boxed{+1}\ -2\ -1\ -3\ +4}$

$\boldsymbol{G_1 = +0\ -1\ +2\ \boxed{-1}\ -3\ -1\ +4}$

And with duplicates?

1. Choose a one-to-one correspondence $\mathcal{M}$ of genes (a matching)
2. Rename or remove genes according to $\mathcal{M}$
3. Compute the (dis)-similarity measure

*exemplar model (**E**)*
[Sankoff, 99]
one occurrence for each
gene family in $\mathcal{M}$

$$\boldsymbol{G_0^E = +0 \; +1 \; -2 \; -3 \; +4}$$

$$\boldsymbol{G_1^E = +0 \; +2 \; -1 \; -3 \; +4}$$

And with duplicates?

1. Choose a one-to-one correspondence $\mathcal{M}$ of genes (a matching)
2. Rename or remove genes according to $\mathcal{M}$
3. Compute the (dis)-similarity measure

*exemplar model (**E**)*
[Sankoff, 99]
one occurrence for each
gene family in $\mathcal{M}$

$$\boldsymbol{G_0^E} = \mathbf{+0^{\blacktriangledown} +1 \ -2^{\blacktriangledown} -3 \ +4}$$

$$\boldsymbol{G_1^E} = \mathbf{+0 \ +2 \ -1 \ -3 \ +4}$$

$$\mathtt{Bkp}(\boldsymbol{G_0^E}, \boldsymbol{G_1^E}) = \mathbf{2}$$

And with duplicates?

1. Choose a one-to-one correspondence $\mathcal{M}$ of genes (a matching)
2. Rename or remove genes according to $\mathcal{M}$
3. Compute the (dis)-similarity measure

*exemplar model (**E**)*
[Sankoff, 99]
one occurrence for each
gene family in $\mathcal{M}$

$$G_0^E = +0^{\blacktriangledown} +1 \; -2^{\blacktriangledown} -3 +4$$

$$G_1^E = +0 \; +2 \; -1 \; -3 \; +4$$

$$\mathtt{Bkp}(G_0^E, G_1^E) = 2$$

*maximum matching model (**M**)*
[Tang & al, 03]
a maximum number of
occurrences in $\mathcal{M}$

$$G_0 = +0 \; +1 \; -2 \; -1 \; -3 \; +4$$

$$G_1 = +0 \; -1 \; +2 \; -1 \; -3 \; -1 \; +4$$

And with duplicates?

1. Choose a one-to-one correspondence $\mathcal{M}$ of genes (a matching)
2. Rename or remove genes according to $\mathcal{M}$
3. Compute the (dis)-similarity measure

*exemplar model (**E**)*
[Sankoff, 99]
one occurrence for each gene family in $\mathcal{M}$

$$G_0^E = +0^{\blacktriangledown} +1 -2^{\blacktriangledown} -3 +4$$

$$G_1^E = +0 +2 -1 -3 +4$$

$$\mathtt{Bkp}(G_0^E, G_1^E) = 2$$

*maximum matching model (**M**)*
[Tang & al, 03]
a maximum number of occurrences in $\mathcal{M}$

$$G_0 = +0 \ \colorbox{green}{+1}\ -2\ \colorbox{cyan}{-1}\ -3 +4$$

$$G_1 = +0\ \colorbox{green}{-1}\ +2\ \colorbox{cyan}{-1}\ -3\ {\color{red}-1} +4$$

And with duplicates?

1. Choose a one-to-one correspondence $\mathcal{M}$ of genes (a matching)
2. Rename or remove genes according to $\mathcal{M}$
3. Compute the (dis)-similarity measure

*exemplar model (**E**)*
[Sankoff, 99]
one occurrence for each gene family in $\mathcal{M}$

$$G_0^E = +0^{\blacktriangledown} +1 \; -2^{\blacktriangledown} -3 \; +4$$

$$G_1^E = +0 \; +2 \; -1 \; -3 \; +4$$

$$\mathtt{Bkp}(G_0^E, G_1^E) = 2$$

*maximum matching model (**M**)*
[Tang & al, 03]
a maximum number of occurrences in $\mathcal{M}$

$$G_0^M = +0 \; +1' \; -2 \; -1'' \; -3 \; +4$$

$$G_1^M = +0 \; -1' \; +2 \; -1'' \; -3 \; +4$$

And with duplicates?

1. Choose a one-to-one correspondence $\mathcal{M}$ of genes (a matching)
2. Rename or remove genes according to $\mathcal{M}$
3. Compute the (dis)-similarity measure

*exemplar model (**E**)*
[Sankoff, 99]
one occurrence for each
gene family in $\mathcal{M}$

$$\boldsymbol{G_0^E} = +0^{\blacktriangledown} +1 -2^{\blacktriangledown} -3 +4$$

$$\boldsymbol{G_1^E} = +0 +2 -1 -3 +4$$

$$\texttt{Bkp}(\boldsymbol{G_0^E}, \boldsymbol{G_1^E}) = 2$$

*maximum matching model (**M**)*
[Tang & al, 03]
a maximum number of
occurrences in $\mathcal{M}$

$$\boldsymbol{G_0^M} = +0^{\blacktriangledown} +1'^{\blacktriangledown} -2^{\blacktriangledown} -1'' -3 +4$$

$$\boldsymbol{G_1^M} = +0 -1' +2 -1'' -3 +4$$

$$\texttt{Bkp}(\boldsymbol{G_0^M}, \boldsymbol{G_1^M}) = 3$$

And with duplicates?

1. Choose a one-to-one correspondence $\mathcal{M}$ of genes (a matching)
2. Rename or remove genes according to $\mathcal{M}$
3. Compute the (dis)-similarity measure

*exemplar model (**E**)*
[Sankoff, 99]
one occurrence for each
gene family in $\mathcal{M}$

*maximum matching model (**M**)*
[Tang & al, 03]
a maximum number of
occurrences in $\mathcal{M}$

*Intermediate model (**I**)*

For each gene family,
at least one gene is kept in $\mathcal{M}$

Several possible matchings?

*maximum matching model (**M**)*
[Tang & al, 03]
a maximum number of occurrences in $\mathcal{M}$

$$\boldsymbol{G_0 = +0\ {+1}\ -2\ {-1}\ -3\ +4}$$

$$\boldsymbol{G_1 = +0\ {-1}\ +2\ {-1}\ -3\ -1\ +4}$$

Several possible matchings?

*maximum matching model (**M**)*
[Tang & al, 03]
a maximum number of occurrences in $\mathcal{M}$

$$G_0 = +0 \; +1 \; -2 \; -1 \; -3 \; +4$$

$$G_1 = +0 \; -1 \; +2 \; -1 \; -3 \; -1 \; +4$$

Measure between genomes with duplicates

Problem

- **Input:**
  - Two genomes $\boldsymbol{G_0}$ and $\boldsymbol{G_1}$
  - A model $\boldsymbol{X} \in \{\boldsymbol{E}, \boldsymbol{M}, \boldsymbol{I}\}$
- **Output:** Find a matching $\mathcal{M}$ which satisfies the model $\boldsymbol{X}$, and which optimizes the measure between $\boldsymbol{G_0^X}$ and $\boldsymbol{G_1^X}$

Measure between genomes with duplicates

Problem

- **Input:**
  - Two genomes $\boldsymbol{G_0}$ and $\boldsymbol{G_1}$
  - A model $\boldsymbol{X} \in \{\boldsymbol{E}, \boldsymbol{M}, \boldsymbol{I}\}$
- **Output:** Find a matching $\mathcal{M}$ which satisfies the model $\boldsymbol{X}$, and which optimizes the measure between $\boldsymbol{G_0^X}$ and $\boldsymbol{G_1^X}$

measure	problem
common interval	$ICOM_X$
conserved interval	$ICONS_X$
breakpoint	BD_X
adjacency	ADJ_X

Measure between genomes with duplicates

Problem

- **Input:**
  - Two genomes $\boldsymbol{G_0}$ and $\boldsymbol{G_1}$
  - A model $\boldsymbol{X} \in \{\boldsymbol{E}, \boldsymbol{M}, \boldsymbol{I}\}$
- **Question:** Are there $\boldsymbol{G_0^X}$ and $\boldsymbol{G_1^X}$ which satisfy the model $\boldsymbol{X}$, and which imply no breakpoint ?

measure	problem	
common interval	$ICOM_X$	ZBD_X
conserved interval	$ICONS_X$	
breakpoint	BD_X	
adjacency	ADJ_X	

Outline

1 Genomes comparison

2 Theoretical complexity results

3 Algorithms

4 MATCH&WATCH application

5 Conclusion

What do we know?

<table>
<tr><th></th><th><i>exemplar
model</i></th><th><i>maximum matching
model</i></th><th><i>intermediate
model</i></th></tr>
<tr><td>$ICOM_X$
$ICONS_X$</td><td colspan="3"><b>NP</b>-Complete [Chauve et al.] (instance $\mathbf{(1, 2)}$)</td></tr>
<tr><td rowspan="2">BD_X</td><td colspan="3"><b>NP</b>-Complete [Bryant] (instance $\mathbf{(1, 2)}$)</td></tr>
<tr><td></td><td><b>NP</b>-Complete [Blin et al.] *</td><td></td></tr>
<tr><td>ZBD_X</td><td><b>NP</b>-Complete
[Chen et al.]
(instance $\mathbf{(3, 3)}$)</td><td>?</td><td>?</td></tr>
</table>

instance $\boldsymbol{(a, b)} \Leftrightarrow \boldsymbol{occ(G_0) = a}$ and $\boldsymbol{occ(G_1) = b}$

* only one family contains several occurrences

Definition

α-approximation and PTAS

- Let ***P*** be an optimization problem
- Let ***I*** be an instance of ***P***
- A polynomial algorithm ***A*** is an α-*approximation* iff
  - If ***P*** is a problem of minimization, then $\boldsymbol{A(I)} \leqslant \alpha \cdot \boldsymbol{optimal(I)}$
  - If ***P*** is a problem of maximization, then $\boldsymbol{A(I)} \geqslant \frac{1}{\alpha} \cdot \boldsymbol{optimal(I)}$

Definition

α-approximation and PTAS

- Let ***P*** be an optimization problem
- Let ***I*** be an instance of ***P***
- A polynomial algorithm ***A*** is an α-*approximation* iff
  - If ***P*** is a problem of minimization, then $\boldsymbol{A(I)} \leqslant \alpha \cdot \boldsymbol{optimal(I)}$
  - If ***P*** is a problem of maximization, then $\boldsymbol{A(I)} \geqslant \frac{1}{\alpha} \cdot \boldsymbol{optimal(I)}$
- A polynomial algorithm ***B*** is a *Polynomial-Time Approximation Scheme* (PTAS) **iff** $\forall \epsilon > \mathbf{0}$
  - If ***P*** is a problem of minimization, then $\boldsymbol{B(I)} \leqslant (\mathbf{1} + \epsilon) \cdot \boldsymbol{optimal(I)}$
  - If ***P*** is a problem of maximization, then $\boldsymbol{B(I)} \geqslant \frac{\mathbf{1}}{\mathbf{1}+\epsilon} \cdot \boldsymbol{optimal(I)}$

Definition

α-approximation and PTAS

- Let P be an optimization problem
- Let I be an instance of P
- A polynomial algorithm A is an *α-approximation* iff
  - If P is a problem of minimization, then $A(I) \leqslant \alpha \cdot optimal(I)$
  - If P is a problem of maximization, then $A(I) \geqslant \frac{1}{\alpha} \cdot optimal(I)$
- A polynomial algorithm B is a *Polynomial-Time Approximation Scheme* (PTAS) **iff** $\forall \epsilon > 0$
  - If P is a problem of minimization, then $B(I) \leqslant (1 + \epsilon) \cdot optimal(I)$
  - If P is a problem of maximization, then $B(I) \geqslant \frac{1}{1+\epsilon} \cdot optimal(I)$

APX-Hard Class

- If a problem P is **APX**-Hard then P does not admit a **PTAS**

New results

	exemplar model	*maximum matching model*	*intermediate model*
$ICOM_X$ $ICONS_X$	**NP**-Complete [Chauve et al.] (instance $\mathbf{(1,2)}$) **APX**-Hard (instance $\mathbf{(1,2)}$) *		
BD_X	**NP**-Complete [Bryant] (instance $\mathbf{(1,2)}$) **APX**-Hard (instance $\mathbf{(1,2)}$) *	**NP**-Complete [Blin et al.]	
ZBD_X	**NP**-Complete [Chen et al.] (instance $\mathbf{(3,3)}$) (instance $\mathbf{(2,k)}$) * [Blin et al.] (instance $\mathbf{(2,2)}$)	polynomial *	$ZBD_I \equiv ZBD_E$ *
ADJ_X	$ADJ_E \simeq BD_E$ *	$ADJ_M \simeq BD_M$ *	$ADJ_I \neq BD_I$ *

* S. Angibaud, G. Fertin, I. Rusu, A. Thévenin et et S. Vialette
On the Approximability of Comparing Genomes with Duplicates
Journal of Graph Algorithms and Applications, Vol. 13(1), pages 19-53, 2009

New results

	exemplar model	*maximum matching model*	*intermediate model*
$ICOM_X$ $ICONS_X$	**NP**-Complete [Chauve et al.] (instance **(1, 2)**) **APX**-Hard (instance **(1, 2)**)		
BD_X	**NP**-Complete [Bryant] (instance **(1, 2)**) **APX**-Hard (instance **(1, 2)**)	**NP**-Complete [Blin et al.]	
ZBD_X	**NP**-Complete [Chen et al.] (instance **(3, 3)**) (instance **(2, *k*)**) [Blin et al.] (instance **(2, 2)**)	polynomial	$ZBD_I \equiv ZBD_E$
ADJ_X	$ADJ_E \simeq BD_E$	$ADJ_M \simeq BD_M$	$ADJ_I \not\simeq BD_I$

A $\simeq$ ***B*** : An optimal solution for ***A*** is an optimal solution for ***B***

A $\not\simeq$ ***B*** : An optimal solution for ***A*** *is not necessarily* an optimal solution for ***B***

New results

	exemplar model	*maximum matching model*	*intermediate model*
$ICOM_X$ $ICONS_X$	**NP**-Complete [Chauve et al.] (instance $\mathbf{(1,2)}$) **APX**-Hard (instance $\mathbf{(1,2)}$)		
BD_X	**NP**-Complete [Bryant] (instance $\mathbf{(1,2)}$) **APX**-Hard (instance $\mathbf{(1,2)}$)	**NP**-Complete [Blin et al.]	
ZBD_X	**NP**-Complete [Chen et al.] (instance $\mathbf{(3,3)}$) (instance $\mathbf{(2,k)}$) [Blin et al.] (instance $\mathbf{(2,2)}$)	polynomial	$ZBD_I \equiv ZBD_E$
ADJ_X	$ADJ_E \simeq BD_E$	$ADJ_M \simeq BD_M$	$ADJ_I \not\simeq BD_I$

$\boldsymbol{A} \simeq \boldsymbol{B}$: An optimal solution for ***A*** is an optimal solution for ***B***

$\boldsymbol{A} \not\simeq \boldsymbol{B}$: An optimal solution for ***A*** *is not necessarily* an optimal solution for ***B***

New results

	exemplar model	*maximum matching model*	*intermediate model*
$ICOM_X$ $ICONS_X$	**NP**-Complete [Chauve et al.] (instance $\mathbf{(1,2)}$) **APX**-Hard (instance $\mathbf{(1,2)}$)		
BD_X	**NP**-Complete [Bryant] (instance $\mathbf{(1,2)}$) **APX**-Hard (instance $\mathbf{(1,2)}$)	**NP**-Complete [Blin et al.]	
ZBD_X	**NP**-Complete [Chen et al.] (instance $\mathbf{(3,3)}$) (instance $\mathbf{(2,k)}$) [Blin et al.] (instance $\mathbf{(2,2)}$)	polynomial	$ZBD_I \equiv ZBD_E$
ADJ_X	$ADJ_E \simeq BD_E$	$ADJ_M \simeq BD_M$	$ADJ_I \neq BD_I$

⇒ Bad news : $ICOM_X$, $ICONS_X$ and BD_X do not admit a polynomial-time approximation scheme (PTAS)

New results

	exemplar model	*maximum matching model*	*intermediate model*
$ICOM_X$ $ICONS_X$	**NP**-Complete [Chauve et al.] (instance $\mathbf{(1,2)}$) **APX**-Hard (instance $\mathbf{(1,2)}$)		
BD_X	**NP**-Complete [Bryant] (instance $\mathbf{(1,2)}$)		
		NP-Complete [Blin et al.]	
	APX-Hard (instance $\mathbf{(1,2)}$)		
ZBD_X	**NP**-Complete [Chen et al.] (instance $\mathbf{(3,3)}$) (instance $\mathbf{(2,k)}$) [Blin et al.] (instance $\mathbf{(2,2)}$)	polynomial	$ZBD_I \equiv ZBD_E$
ADJ_X	$ADJ_E \simeq BD_E$	$ADJ_M \simeq BD_M$	$ADJ_I \not\equiv BD_I$

$\Rightarrow$ Bad news : BD_E and BD_I do not admit any α-approximation, unless **P** = **NP**

New results

	exemplar model	*maximum matching model*	*intermediate model*
$ICOM_X$ $ICONS_X$	**NP**-Complete [Chauve et al.] (instance $\mathbf{(1,2)}$) **APX**-Hard (instance $\mathbf{(1,2)}$)		
BD_X	**NP**-Complete [Bryant] (instance $\mathbf{(1,2)}$) **APX**-Hard (instance $\mathbf{(1,2)}$)	**NP**-Complete [Blin et al.]	
ZBD_X	**NP**-Complete [Chen et al.] (instance $\mathbf{(3,3)}$) (instance $\mathbf{(2,\mathit{k})}$) [Blin et al.] (instance $\mathbf{(2,2)}$)	polynomial	$ZBD_I \equiv ZBD_E$
ADJ_X	$ADJ_E \simeq BD_E$	$ADJ_M \simeq BD_M$	$ADJ_I \not\simeq BD_I$

$\Rightarrow$ Good news : BD_M **could** admit an α-approximation

Outline

1. Genomes comparison

2. Theoretical complexity results

3. Algorithms
   - Exact approach
     - Pseudo boolean problem
     - Pseudo-boolean transformation for $\boldsymbol{ICOM_E}$
     - Experimental results
   - Heuristics and hybrid method
     - $\mathit{IILCS_X}$
     - Hybrid method
     - Experimental results

4. MATCH&WATCH application

Exact algorithm

Problem

- **Input:**
  - Two genomes G_0 and G_1
  - A model $X \in \{E, M, I\}$
- **Output:** Find a matching $\mathcal{M}$ which satisfies the model X, and which optimizes the measure between G_0^X and G_1^X

Idea: transformation into a pseudo boolean linear problem

Pseudo-boolean linear problem

Definition

- **Variables**: domain $= \{0, 1\}$
- **Constraints**: inequalities between weighted sum of variables
- **Objective function**: weighted sum of variables

Example

- **Variables**: $x \in \{0, 1\}, y \in \{0, 1\}, z \in \{0, 1\}$
- **Constraints**:
  - $x + 2 \cdot y \geqslant 2$
  - $z + y \leqslant 1$
- **Objective function**:
  maximize $x + 2 \cdot y - z$

Pseudo-boolean linear problem

Definition

- **Variables**: boolean
- **Constraints**: inequalities between weighted sum of variables
- **Objective function**: weighted sum of variables

Example

- **Variables**: $x \in \{0,1\}, y \in \{0,1\}, z \in \{0,1\}$
- **Constraints**:
  - $x + 2 \cdot y \geqslant 2$
  - $z + y \leqslant 1$
- **Objective function**:
  maximize $x + 2 \cdot y - z$

Pseudo-boolean linear problem

Definition

- **Variables**: boolean
- **Constraints**: inequalities between weighted sum of variables
- **Objective function**: weighted sum of variables

Example

- **Variables**: $x \in \{0,1\}, y \in \{0,1\}, z \in \{0,1\}$
- **Constraints**:
    - $x + 2 \cdot y \geqslant 2$
    - $z + y \leqslant 1$
- **Objective function**:
  maximize $x + 2 \cdot \; y - z$

Pseudo-boolean linear problem

Definition

- **Variables**: boolean
- **Constraints**: inequalities between weighted sum of variables
- **Objective function**: weighted sum of variables

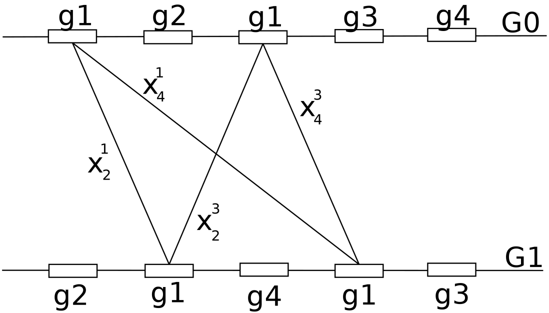
Example

- **Variables**: $x \in \{0, 1\}, y \in \{0, 1\}, z \in \{0, 1\}$
- **Constraints**:
  - $x + 2 \cdot y \geqslant 2$
  - $z + y \leqslant 1$
- **Objective function**:
  maximize $x + 2 \cdot y - z$

⇒ Powerful solvers for this type of problem

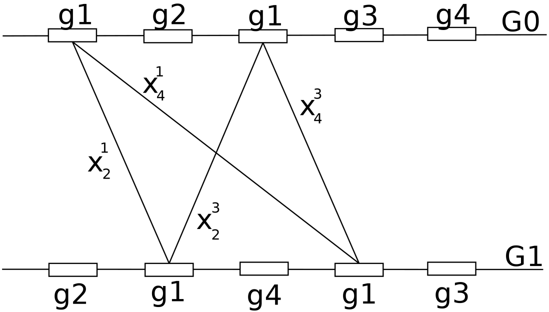
Transformation for $ICOM_E$: variables

- Variables $\boldsymbol{x}$ and $\boldsymbol{l}$:



Transformation for $ICOM_E$: variables

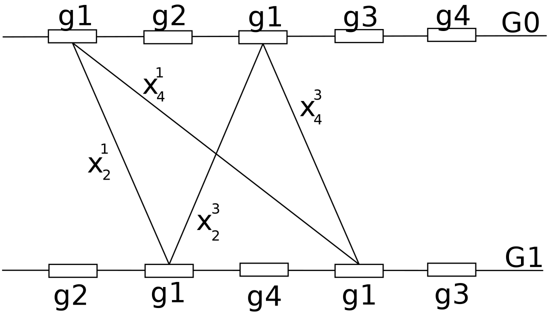
- Variables $\boldsymbol{x}$ and $\boldsymbol{l}$:



$\boldsymbol{x_b^a}$ true ⇔ gene $\boldsymbol{G_0[a]}$ and $\boldsymbol{G_1[b]}$ are matched

Transformation for $ICOM_E$: variables

- Variables $\boldsymbol{x}$ and $\boldsymbol{l}$:



$l_{k,l,m,n}$ true ⇔ $[k, l]$ in G_0 is a common interval of (G_0, G_1), and $[m, n]$ in G_1 is a permutation of $[k, l]$

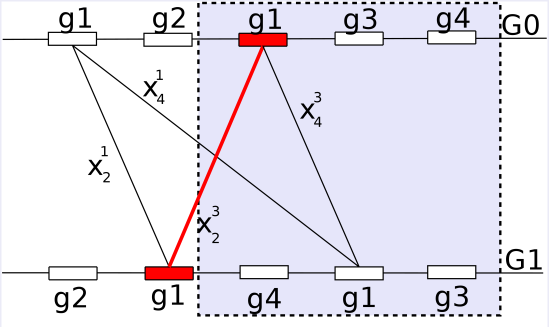
Transformation for $ICOM_E$: constraints

Exemplar model:
for each genome, only one occurrence of each gene family

$$\text{C1: } \forall \boldsymbol{f} \in \mathcal{F}_{G_0} \cup \mathcal{F}_{G_1}, \quad \sum_{\substack{1 \leqslant a \leqslant \eta_{G_0} \\ G_0[a]=f}} \; \sum_{\substack{1 \leqslant b \leqslant \eta_{G_1} \\ G_1[b]=f}} x_b^a = 1$$

Transformation for $ICOM_E$: constraints

Validity of variables $I_{k,l,m,n}$



$$I_{k,\ell,m,n} + x_2^3 \leqslant 1$$

Transformation for $ICOM_E$

Objective function:

$$\text{Maximize} \sum_{\boldsymbol{k,l,m,n}} \boldsymbol{l_{k,l,m,n}}$$

Transformation for $ICOM_E$

Variables:

$$\mathcal{I} = \{ I_{k,l,m,n} : 1 \leqslant k \leqslant \ell \leqslant \eta_{G_0} \wedge 1 \leqslant m \leqslant n \leqslant \eta_{G_1} \}$$
$$\mathcal{X} = \{ x_b^a : 1 \leqslant a \leqslant \eta_{G_0} \wedge 1 \leqslant b \leqslant \eta_{G_1} \wedge G_0[a] = G_1[b] \}$$

Constraints:

(C.01) $\forall f \in \mathcal{F}_{G_0} \cup \mathcal{F}_{G_1}, \quad \sum_{\substack{1 \leqslant a \leqslant \eta_{G_0} \\ G_0[a]=f}} \; \sum_{\substack{1 \leqslant b \leqslant \eta_{G_1} \\ G_1[b]=f}} x_b^a = 1$

(C.02) $\forall I_{k,l,m,n} \in \mathcal{I}, \ \forall k < p < \ell, \ \forall 1 \leqslant r < m, \quad G_0[p] = G_1[r], \quad I_{k,l,m,n} + x_r^p \leqslant 1$

(C.03) $\forall I_{k,l,m,n} \in \mathcal{I}, \ \forall k < p < \ell, \ \forall n < r \leqslant \eta_{G_1}, \ G_0[p] = G_1[r], \quad I_{k,l,m,n} + x_r^p \leqslant 1$

(C.04) $\forall I_{k,l,m,n} \in \mathcal{I}, \ \forall m < r < n, \ \forall 1 \leqslant p < k, \quad G_0[p] = G_1[r], \quad I_{k,l,m,n} + x_r^p \leqslant 1$

(C.05) $\forall I_{k,l,m,n} \in \mathcal{I}, \ \forall m < r < n, \ \forall \ell < p \leqslant \eta_{G_0}, \ G_0[p] = G_1[r], \quad I_{k,l,m,n} + x_r^p \leqslant 1$

(C.06) $\forall I_{k,l,m,n} \in \mathcal{I}, \quad 4\, I_{k,l,m,n} - \sum_{\substack{m \leqslant r \leqslant n \\ G_0[k]=G_1[r]}} x_r^k - \sum_{\substack{m \leqslant s \leqslant n \\ G_0[\ell]=G_1[s]}} x_s^\ell - \sum_{\substack{k \leqslant p \leqslant \ell \\ G_0[p]=G_1[m]}} x_m^p - \sum_{\substack{k \leqslant q \leqslant \ell \\ G_0[q]=G_1[n]}} x_n^q \leqslant 0$

Objective function:

Maximize $\sum_{k,l,m,n} I_{k,l,m,n}$

Transformation for $ICOM_E$

Variables:

$$\mathcal{I} = \{I_{k,l,m,n} : 1 \leqslant k \leqslant \ell \leqslant \eta_{G_0} \wedge 1 \leqslant m \leqslant n \leqslant \eta_{G_1}\}$$
$$\mathcal{X} = \{x_b^a : 1 \leqslant a \leqslant \eta_{G_0} \wedge 1 \leqslant b \leqslant \eta_{G_1} \wedge G_0[a] = G_1[b]\}$$

Constraints:

(C.01) $\forall f \in \mathcal{F}_{G_0} \cup \mathcal{F}_{G_1}, \quad \sum_{\substack{1 \leqslant a \leqslant \eta_{G_0} \\ G_0[a]=f}} \sum_{\substack{1 \leqslant b \leqslant \eta_{G_1} \\ G_1[b]=f}} x_b^a = 1$

(C.02) $\forall I_{k,l,m,n} \in \mathcal{I}, \ \forall k < p < \ell, \ \forall 1 \leqslant r < m, \quad G_0[p] = G_1[r], \quad I_{k,l,m,n} + x_r^p \leqslant 1$

(C.03) $\forall I_{k,l,m,n} \in \mathcal{I}, \ \forall k < p < \ell, \ \forall n < r \leqslant \eta_{G_1}, \ G_0[p] = G_1[r], \quad I_{k,l,m,n} + x_r^p \leqslant 1$

(C.04) $\forall I_{k,l,m,n} \in \mathcal{I}, \ \forall m < r < n, \ \forall 1 \leqslant p < k, \quad G_0[p] = G_1[r], \quad I_{k,l,m,n} + x_r^p \leqslant 1$

(C.05) $\forall I_{k,l,m,n} \in \mathcal{I}, \ \forall m < r < n, \ \forall \ell < p \leqslant \eta_{G_0}, \ G_0[p] = G_1[r], \quad I_{k,l,m,n} + x_r^p \leqslant 1$

(C.06) $\forall I_{k,l,m,n} \in \mathcal{I}, \quad 4\, I_{k,l,m,n} - \sum_{\substack{m \leqslant r \leqslant n \\ G_0[k]=G_1[r]}} x_r^k - \sum_{\substack{m \leqslant s \leqslant n \\ G_0[\ell]=G_1[s]}} x_s^\ell - \sum_{\substack{k \leqslant p \leqslant \ell \\ G_0[p]=G_1[m]}} x_m^p - \sum_{\substack{k \leqslant q \leqslant \ell \\ G_0[q]=G_1[n]}} x_n^q \leqslant 0$

Objective function:

Maximize $\sum_{k,l,m,n} I_{k,l,m,n}$

Transformation for $ICOM_E$

Variables:

$$\mathcal{I} = \{I_{k,l,m,n} : 1 \leqslant k \leqslant \ell \leqslant \eta_{G_0} \wedge 1 \leqslant m \leqslant n \leqslant \eta_{G_1}\}$$
$$\mathcal{X} = \{x_b^a : 1 \leqslant a \leqslant \eta_{G_0} \wedge 1 \leqslant b \leqslant \eta_{G_1} \wedge G_0[a] = G_1[b]\}$$

Constraints:

(C.01) $\forall f \in \mathcal{F}_{G_0} \cup \mathcal{F}_{G_1}, \quad \sum_{\substack{1\leqslant a\leqslant \eta_{G_0} \\ G_0[a]=f}} \sum_{\substack{1\leqslant b\leqslant \eta_{G_1} \\ G_1[b]=f}} x_b^a = 1$

(C.02) $\forall I_{k,l,m,n} \in \mathcal{I}, \ \forall k < p < \ell, \ \forall 1 \leqslant r < m, \quad G_0[p] = G_1[r], \quad I_{k,l,m,n} + x_r^p \leqslant 1$

(C.03) $\forall I_{k,l,m,n} \in \mathcal{I}, \ \forall k < p < \ell, \ \forall n < r \leqslant \eta_{G_1}, \ G_0[p] = G_1[r], \quad I_{k,l,m,n} + x_r^p \leqslant 1$

(C.04) $\forall I_{k,l,m,n} \in \mathcal{I}, \ \forall m < r < n, \ \forall 1 \leqslant p < k, \quad G_0[p] = G_1[r], \quad I_{k,l,m,n} + x_r^p \leqslant 1$

(C.05) $\forall I_{k,l,m,n} \in \mathcal{I}, \ \forall m < r < n, \ \forall \ell < p \leqslant \eta_{G_0}, \ G_0[p] = G_1[r], \quad I_{k,l,m,n} + x_r^p \leqslant 1$

(C.06) $\forall I_{k,l,m,n} \in \mathcal{I}, \quad 4\, I_{k,l,m,n} - \sum_{\substack{m\leqslant r\leqslant n \\ G_0[k]=G_1[r]}} x_r^k - \sum_{\substack{m\leqslant s\leqslant n \\ G_0[\ell]=G_1[s]}} x_s^\ell - \sum_{\substack{k\leqslant p\leqslant \ell \\ G_0[p]=G_1[m]}} x_m^p - \sum_{\substack{k\leqslant q\leqslant \ell \\ G_0[q]=G_1[n]}} x_n^q \leqslant 0$

Objective function:

Maximize $\sum_{k,l,m,n} I_{k,l,m,n}$

Transformation for $ICOM_E$

Variables:

$$\mathcal{I} = \{I_{k,l,m,n} : 1 \leqslant k \leqslant \ell \leqslant \eta_{G_0} \wedge 1 \leqslant m \leqslant n \leqslant \eta_{G_1}\}$$
$$\mathcal{X} = \{x_b^a : 1 \leqslant a \leqslant \eta_{G_0} \wedge 1 \leqslant b \leqslant \eta_{G_1} \wedge G_0[a] = G_1[b]\}$$

Constraints:

(C.01) $\forall f \in \mathcal{F}_{G_0} \cup \mathcal{F}_{G_1}, \quad \sum\limits_{\substack{1\leqslant a\leqslant \eta_{G_0}\\ G_0[a]=f}} \sum\limits_{\substack{1\leqslant b\leqslant \eta_{G_1}\\ G_1[b]=f}} x_b^a = 1$

(C.02) $\forall I_{k,l,m,n} \in \mathcal{I}, \ \forall k < p < \ell, \ \forall 1 \leqslant r < m, \quad G_0[p] = G_1[r], \quad I_{k,l,m,n} + x_r^p \leqslant 1$

(C.03) $\forall I_{k,l,m,n} \in \mathcal{I}, \ \forall k < p < \ell, \ \forall n < r \leqslant \eta_{G_1}, \ G_0[p] = G_1[r], \quad I_{k,l,m,n} + x_r^p \leqslant 1$

(C.04) $\forall I_{k,l,m,n} \in \mathcal{I}, \ \forall m < r < n, \ \forall 1 \leqslant p < k, \quad G_0[p] = G_1[r], \quad I_{k,l,m,n} + x_r^p \leqslant 1$

(C.05) $\forall I_{k,l,m,n} \in \mathcal{I}, \ \forall m < r < n, \ \forall \ell < p \leqslant \eta_{G_0}, \ G_0[p] = G_1[r], \quad I_{k,l,m,n} + x_r^p \leqslant 1$

(C.06) $\forall I_{k,l,m,n} \in \mathcal{I}, \quad 4\, I_{k,l,m,n} - \sum\limits_{\substack{m\leqslant r\leqslant n\\ G_0[k]=G_1[r]}} x_r^k - \sum\limits_{\substack{m\leqslant s\leqslant n\\ G_0[\ell]=G_1[s]}} x_s^\ell - \sum\limits_{\substack{k\leqslant p\leqslant \ell\\ G_0[p]=G_1[m]}} x_m^p - \sum\limits_{\substack{k\leqslant q\leqslant \ell\\ G_0[q]=G_1[n]}} x_n^q \leqslant 0$

Objective function:

Maximize $\sum\limits_{k,l,m,n} I_{k,l,m,n}$

Pseudo boolean transformation

Other problems ?

- **other models:** modify constraints C1
- **conserved intervals:** restriction on variables $\boldsymbol{I_{k,\ell,m,n}}$
- **breakpoint and adjacency:** new variables and constraints

- ***ICOM$_X$* and *ICONS$_X$***
  S. Angibaud, G. Fertin, I. Rusu et S. Vialette.
  A pseudo-boolean general framework for computing rearrangement distances between genomes with duplicates

  Journal of Computational Biology, Vol. 14(4), pages 379-393. 2007

- ***BD$_X$* and *ADJ$_X$***
  S. Angibaud, G. Fertin, I. Rusu, A. Thévenin et 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, Vol. 15(8), pages 1093-1115. 2008

Experimental results

Dataset

- Twelve genomes of γ-*Proteobacteria* [Lerat et al. 2003]

Name	Genbank identifier	size
Buchnera aphidicola APS	NC_002528	564
Escherichia coli K12	NC_000913	4183
Haemophilus influenzae Rd	NC_000907	1709
Pseudomonas aeruginosa PA01	NC_002516	5540
Pasteurella multocida Pm70	NC_002663	2015
Salmonella typhimurium LT2	NC_003197	4203
Wigglesworthia glossinidia brevipalpis	NC_004344	653
Xanthomonas axonopodis pv. citri 306	NC_003919	4192
Xanthomonas campestris	NC_0 03902	4029
Xylella fastidiosa 9a5c	NC_002488	2680
Yersinia pestis CO_92	NC_003143	3599
Yersinia pestis KIM5 P12	NC_004088	3879
	average:	3104

Experimental results

Dataset

- Twelve genomes of γ-*Proteobacteria* [Lerat et al. 2003]
- 66 possible pairs of genomes

Number of results:

	model		
	Exemplar	maximum matching	intermediate
ADJ_X	61/66	66/66	63/66
$ICOM_X$	21/66	40/66	21/66

Experimental results

Dataset

- Twelve genomes of γ-*Proteobacteria* [Lerat et al. 2003]
- 66 possible pairs of genomes

Number of results:

	model		
	Exemplar	maximum matching	intermediate
ADJ_X	61/66	66/66	63/66
$ICOM_X$	21/66	40/66	21/66

$\Rightarrow$ Efficient approach for ADJ_X

Experimental results

Dataset

- Twelve genomes of γ-*Proteobacteria* [Lerat et al. 2003]
- 66 possible pairs of genomes

Number of results:

	model		
	Exemplar	maximum matching	intermediate
ADJ_X	61/66	66/66	63/66
$ICOM_X$	21/66	40/66	21/66

⇒ Efficient approach for ADJ_X

⇒ Limit is attained for $ICOM_X$

⇒ Heuristics

Outline

1 Genomes comparison

2 Theoretical complexity results

3 Algorithms
- Exact approach
  - Pseudo boolean problem
  - Pseudo-boolean transformation for $\textbf{\textit{ICOM}}_E$
  - Experimental results
- Heuristics and hybrid method
  - $\text{IILCS}_{\boldsymbol{x}}$
  - Hybrid method
  - Experimental results

4 MATCH&WATCH application

IILCS$_M$ heuristic

- Based on ILCS$_M$ heuristic [Tichy, 82]
- **Idea:** Match genes of a Longest Common Substring (LCS)

IILCS$_M$ heuristic

- Based on ILCS$_M$ heuristic [Tichy, 82]
- **Idea:** Match genes of a Longest Common Substring (LCS)

IILCS$_M$ heuristic

1. Compute the Longest Common Substring ***S***

Example

$$+1\ +2\ +3\ \ +4\ +5\ +6\ +7$$

$$+6\ -7\ +4\ +5\ +1\ +6\ \ -3\ -2\ -1$$

IILCS$_M$ heuristic

- Based on ILCS$_M$ heuristic [Tichy, 82]
- **Idea:** Match genes of a Longest Common Substring (LCS)

IILCS$_M$ heuristic

1. Compute the Longest Common Substring S
2. Match all the genes of S accordingly

Example

+1 +2 +3 + 4 + 5 + 6 + 7

+6 − 7 + 4 + 5 + 1 + 6 -3 -2 -1

IILCS$_M$ heuristic

- Based on ILCS$_M$ heuristic [Tichy, 82]
- **Idea:** Match genes of a Longest Common Substring (LCS)

IILCS$_M$ heuristic

1. Compute the Longest Common Substring ***S***
2. Match all the genes of ***S*** accordingly
3. Remove genes that cannot be matched

Example

+1 +2 +3 **+ 4 + 5 + 6 + 7**

+6 − 7 + 4 + 5 +1 **+ 6** -3 -2 -1

IILCS$_M$ heuristic

- Based on ILCS$_M$ heuristic [Tichy, 82]
- **Idea:** Match genes of a Longest Common Substring (LCS)

IILCS$_M$ heuristic

1. Compute the Longest Common Substring $\boldsymbol{S}$
2. Match all the genes of $\boldsymbol{S}$ accordingly
3. Remove genes that cannot be matched

Example

+1 +2 +3 $+\mathbf{4}$ $+\mathbf{5}$ $+\mathbf{6}$ $+\mathbf{7}$

$+\mathbf{6}$ $-\mathbf{7}$ $+\mathbf{4}$ $+\mathbf{5}$ $+\mathbf{6}$ -3 -2 -1

IILCS$_M$ heuristic

- Based on ILCS$_M$ heuristic [Tichy, 82]
- **Idea:** Match genes of a Longest Common Substring (LCS)

IILCS$_M$ heuristic

1. Compute the Longest Common Substring S
2. Match all the genes of S accordingly
3. Remove genes that cannot be matched
4. Iterate the process until saturation

Example

+1 +2 +3 $+4$ $+5$ $+6$ $+7$

+6 -7 $+4$ $+5$ $+6$ -3 -2 -1

IILCS$_M$ heuristic

- Based on ILCS$_M$ heuristic [Tichy, 82]
- **Idea:** Match genes of a Longest Common Substring (LCS)

IILCS$_M$ heuristic

1. Compute the Longest Common Substring $\boldsymbol{S}$
2. Match all the genes of $\boldsymbol{S}$ accordingly
3. Remove genes that cannot be matched
4. Iterate the process until saturation

Example

+1 +2 +3 +4 +5 +6 $+$**7**

$+$**6** $-$**7** +4 +5 +6 -3 -2 -1

IILCS$_M$ heuristic

- Based on ILCS$_M$ heuristic [Tichy, 82]
- **Idea:** Match genes of a Longest Common Substring (LCS)

IILCS$_M$ heuristic

1. Compute the Longest Common Substring ***S***
2. Match all the genes of ***S*** accordingly
3. Remove genes that cannot be matched
4. Iterate the process until saturation

Example

+1 +2 +3 +4 +5 +6 $+$ **7**

+6 $-$ **7** +4 +5 +6 -3 -2 -1

IILCS$_M$ heuristic

- Based on ILCS$_M$ heuristic [Tichy, 82]
- **Idea:** Match genes of a Longest Common Substring (LCS)

IILCS$_M$ heuristic

1. Compute the Longest Common Substring $\boldsymbol{S}$
2. Match all the genes of $\boldsymbol{S}$ accordingly
3. Remove genes that cannot be matched
4. Iterate the process until saturation

Example

+1 +2 +3 +4 +5 +6 $+\mathbf{7}$

$-\mathbf{7}$ +4 +5 +6 -3 -2 -1

IILCS$_M$ heuristic

- Based on ILCS$_M$ heuristic [Tichy, 82]
- **Idea:** Match genes of a Longest Common Substring (LCS)

IILCS$_M$ heuristic

1. Compute the Longest Common Substring S
2. Match all the genes of S accordingly
3. Remove genes that cannot be matched
4. Iterate the process until saturation

Example

+1 +2 +3 +4 +5 +6 +7

-7 +4 +5 +6 -3 -2 -1

IILCS$_M$ heuristic

- Based on ILCS$_M$ heuristic [Tichy, 82]
- **Idea:** Match genes of a Longest Common Substring (LCS)

IILCS$_M$ heuristic

1. Compute the Longest Common Substring ***S***
2. Match all the genes of ***S*** accordingly
3. Remove genes that cannot be matched
4. Iterate the process until saturation
5. Compute the measure

Example

+1 +2 +3 | +4 +5 +6 | +7

-7 | +4 +5 +6 | -3 -2 -1

Hybrid method

Algorithm HYB$_X$(k)

- **Idea:** Associate exact method and IILCS$_X$ heuristic
- **Parameter k:** Bound on **LCS** size

```
1. Compute an LCS $S$ of $(G_0, G_1)$
2. If $|S| \geqslant k$
   Then
       Match all the genes of $S$
       Remove genes that cannot be matched
       Return to 1
   Else Apply the exact method: transformation into a pseudo-boolean linear problem
```

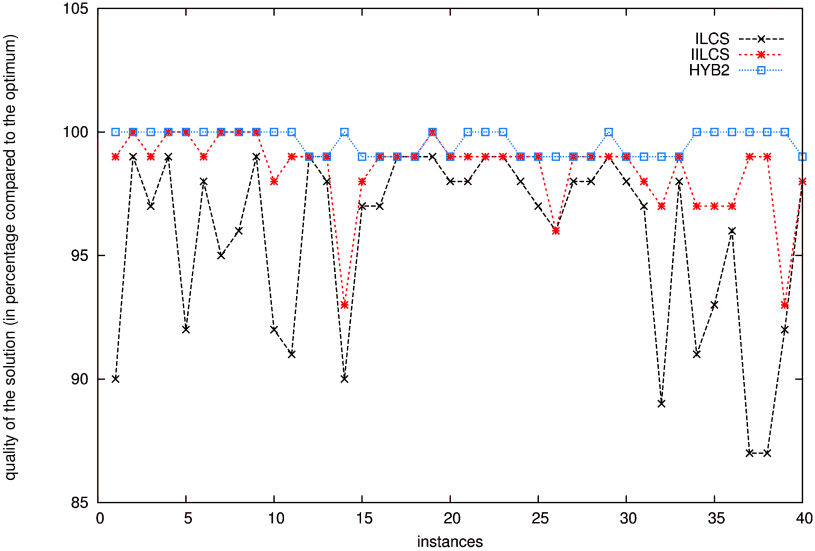
Experimental results

Dataset

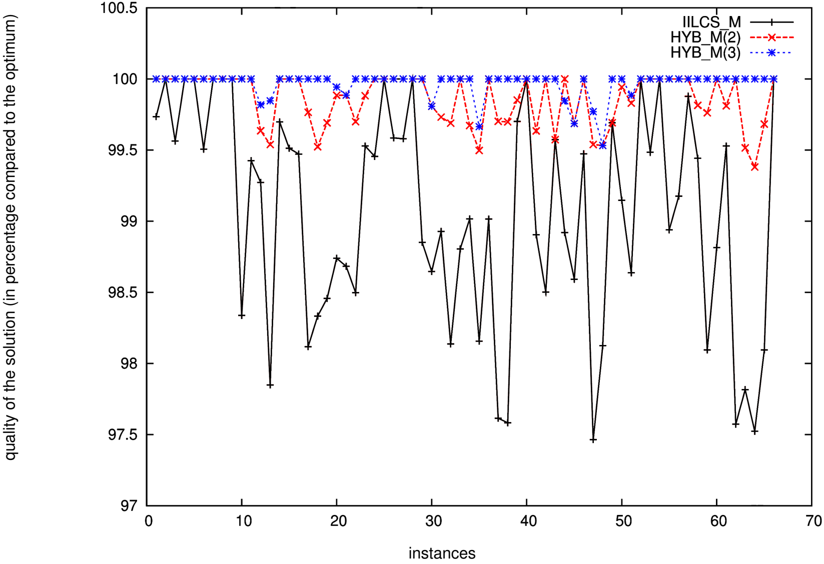
- Twelve genomes of γ-*Proteobacteria* [Lerat et al. 2003]
- 66 possible pairs of genomes

EXACT	model		
	Exemplar	maximum matching	intermediate
ADJ_X	61/66	66/66	63/66
$ICOM_X$	21/66	40/66	21/66

Experimental results: $ICOM_M$



Experimental results: ADJ_M



Outline

1. Genomes comparison
2. Theoretical complexity results
3. Algorithms
4. MATCH&WATCH application
   - Protocol
   - Visualization tool
5. Conclusion

Goal

Problem

- **Input:** two circular genomes G_1 and G_2
- **Output:** List of common intervals between G_1 and G_2

Goal

- Compute common intervals
- Provide a tool to visualize and analyze results

S. Angibaud, D. Éveillard, G. Fertin et I. Rusu
Comparing Bacterial Genomes by Searching Their Common Intervals
In Proc. 1st International Conference on Bioinformatics and Computational Biology
LNBI Vol. 5462, pages 102-113. 2009

Protocol

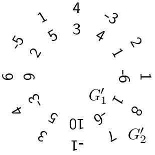
Step 1) Input:
Two genomes G_1 and G_2 are obtained on the `NCBI` website in FASTA format

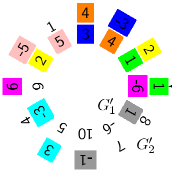
G_1.fa file
16127995
MKRISTTI...
16127996
MRVLKFG...
...

homologies file
16127995;9786796
1612798;154627;154786
...

Step 2) Homologies detection:
`InParanoid` is applied to detect homologies between genes of G_1 and G_2

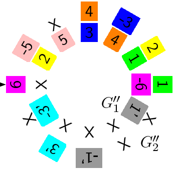
Step 3) Intermediate genomes construction:
Genes are renamed according to homologies to construct G'_1 and G'_2





Step 4) Matching choice:
A matching between G'_1 and G'_2 is obtained

Step 5) Matching application:
G''_1, G''_2 are constructed by renaming genes and removing unmatched genes



intervals file
$([3,5],[3,5])$
$([2,5],[3,6])$
...

Step 6) Measure computation:
Common intervals positions of (G''_1, G''_2) are listed

Step 7) Visualization

Protocol

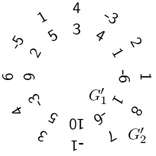
Step 1) Input:
Two genomes G_1 and G_2 are obtained on the `NCBI` website in FASTA format

G_1.fa file
16127995
MKRISTTI...
16127996
MRVLKFG...
...

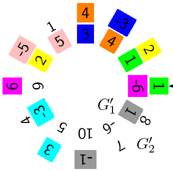
homologies file
16127995;9786796
1612798;154627;154786
...

Step 2) Homologies detection:
`InParanoid` is applied to detect homologies between genes of G_1 and G_2

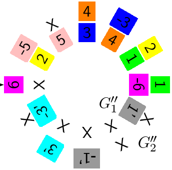
Step 3) Intermediate genomes construction:
Genes are renamed according to homologies to construct G'_1 and G'_2



Step 4) Matching choice:
A matching between G'_1 and G'_2 is obtained



Step 5) Matching application:
G''_1, G''_2 are constructed by renaming genes and removing unmatched genes

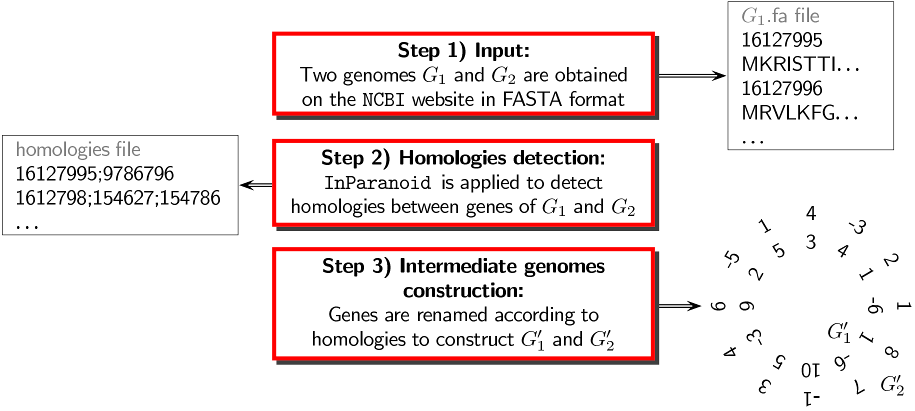


intervals file
$([3,5],[3,5])$
$([2,5],[3,6])$
...

Step 6) Measure computation:
Common intervals positions of (G''_1, G''_2) are listed

Step 7) Visualization

Homologies computation



Step 1) Input:
Two genomes G_1 and G_2 are obtained on the `NCBI` website in FASTA format

G_1.fa file
16127995
MKRISTTI...
16127996
MRVLKFG...
...

Step 2) Homologies detection:
`InParanoid` is applied to detect homologies between genes of G_1 and G_2

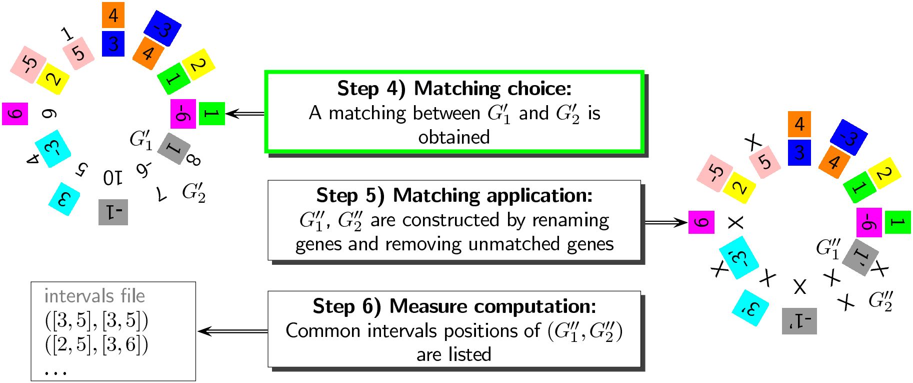
homologies file
16127995;9786796
1612798;154627;154786
...

Step 3) Intermediate genomes construction:
Genes are renamed according to homologies to construct G'_1 and G'_2

Inparanoid [Storm et al. 2001]

- Proposed in 2001 by Storm, Remm and Sonnhammer
- Compute clusters of homologous genes

Step 4: choose a matching



Step 4) Matching choice: A matching between G'_1 and G'_2 is obtained

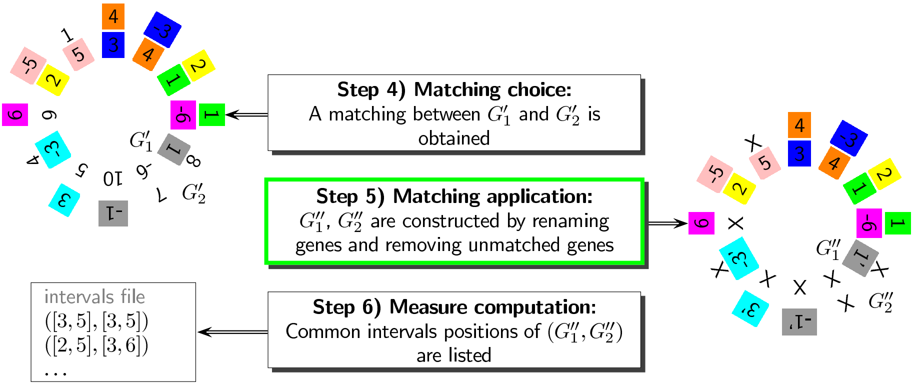
Step 5) Matching application: G''_1, G''_2 are constructed by renaming genes and removing unmatched genes

Step 6) Measure computation: Common intervals positions of (G''_1, G''_2) are listed

intervals file
$([3,5],[3,5])$
$([2,5],[3,6])$
$\ldots$

- Exact method: Pseudo boolean transformation
- IILCS$_X$ heuristic
- Hybrid method

Step 5: Matching application



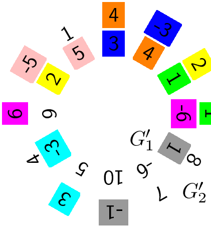
Step 4) Matching choice:
A matching between G'_1 and G'_2 is obtained

Step 5) Matching application:
G''_1, G''_2 are constructed by renaming genes and removing unmatched genes

Step 6) Measure computation:
Common intervals positions of (G''_1, G''_2) are listed

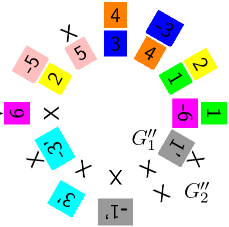
intervals file
$([3,5],[3,5])$
$([2,5],[3,6])$
$\dots$

Step 6: common intervals computation



Step 4) Matching choice:
A matching between G'_1 and G'_2 is obtained

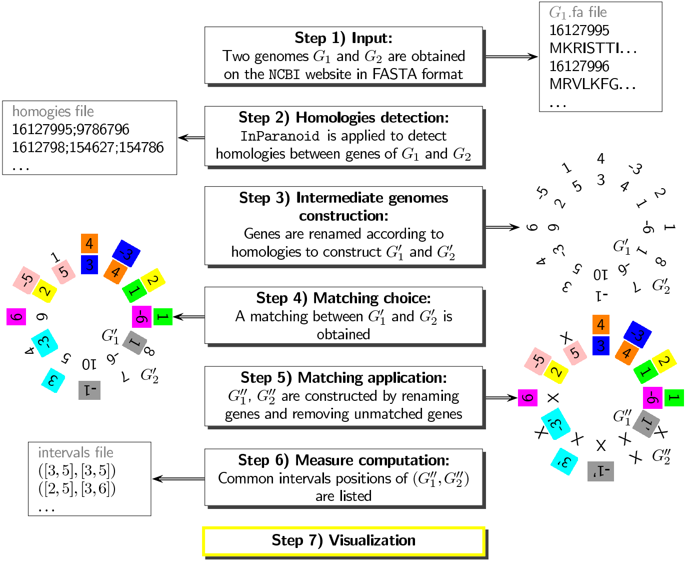
Step 5) Matching application:
G''_1, G''_2 are constructed by renaming genes and removing unmatched genes



Step 6) Measure computation:
Common intervals positions of (G''_1, G''_2) are listed

intervals file
$([3,5],[3,5])$
$([2,5],[3,6])$
$\ldots$

Seven steps



Step 1) Input:
Two genomes G_1 and G_2 are obtained on the `NCBI` website in FASTA format

G_1.fa file
16127995
MKRISTTI...
16127996
MRVLKFG...
...

Step 2) Homologies detection:
`InParanoid` is applied to detect homologies between genes of G_1 and G_2

homogies file
16127995;9786796
1612798;154627;154786
...

Step 3) Intermediate genomes construction:
Genes are renamed according to homologies to construct G'_1 and G'_2

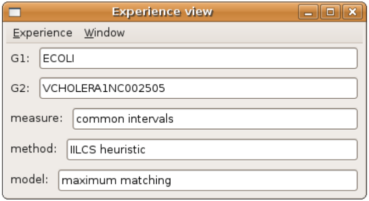
Step 4) Matching choice:
A matching between G'_1 and G'_2 is obtained

Step 5) Matching application:
G''_1, G''_2 are constructed by renaming genes and removing unmatched genes

Step 6) Measure computation:
Common intervals positions of (G''_1, G''_2) are listed

intervals file
$([3,5],[3,5])$
$([2,5],[3,6])$
...

Step 7) Visualization


Experience view

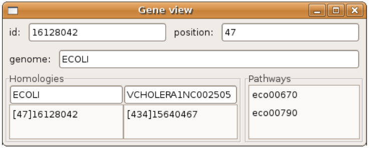
G1: ECOLI

G2: VCHOLERA1NC002505

measure: common intervals

method: IILCS heuristic

model: maximum matching


Gene view

id: 16128042

position: 47

genome: ECOLI

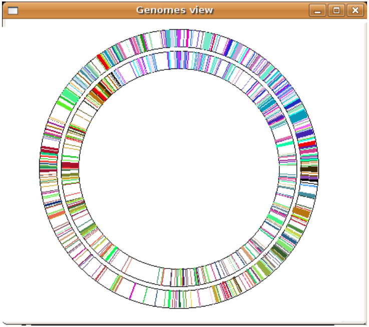
Homologies

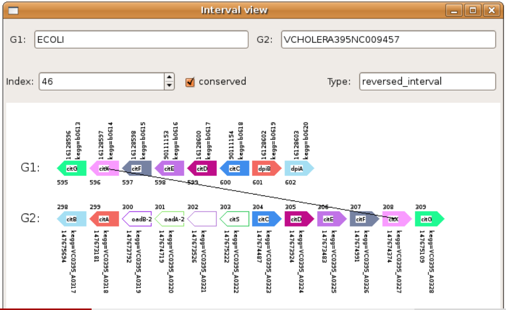
ECOLI	VCHOLERA1NC002505
[47]16128042	[434]15640467

Pathways

eco00670

eco00790


Genomes view


Interval view

G1: ECOLI

G2: VCHOLERA395NC009457

Index: 46

☑ conserved

Type: reversed_interval

G1:
16128596 kegg=b0613 citG 595
16128597 kegg=b0614 citX 596
16128598 kegg=b0615 citF 597
90111153 kegg=b0616 citE 598
16128600 kegg=b0617 citD 599
90111154 kegg=b0618 citC 600
16128602 kegg=b0619 dpiB 601
16128603 kegg=b0620 dpiA 602
G2:
298 citB 147675694 kegg=VC0395_A0317
299 citA 147673181 kegg=VC0395_A0318
300 oadB-2 147673792 kegg=VC0395_A0319
301 oadA-2 147674719 kegg=VC0395_A0320
302 147673526 kegg=VC0395_A0321
303 citS 147675222 kegg=VC0395_A0322
304 citC 147674487 kegg=VC0395_A0323
305 citD 147673924 kegg=VC0395_A0324
306 citE 147673483 kegg=VC0395_A0325
307 citF 147674991 kegg=VC0395_A0326
308 citX 147674374 kegg=VC0395_A0327
309 citG 147675109 kegg=VC0395_A0328

Outline

1 Genomes comparison

2 Theoretical complexity results

3 Algorithms

4 MATCH&WATCH application

5 Conclusion

Contributions

- Better knowledge of problems
  - **APX**-Hardness of BD_X, $ICOM_X$ and $ICONS_X$
  - **NP**-Completeness of ZBD_E and ZBD_I
  - Polynomiality of ZBD_M

Contributions

- Better knowledge of problems
  - **APX**-Hardness of BD_X, $ICOM_X$ and $ICONS_X$
  - **NP**-Completeness of ZBD_E and ZBD_I
  - Polynomiality of ZBD_M
- Three new algorithms
  - An exact approach based on a transformation into a pseudo-boolean problem
    - Efficient approach for BD_X and ADJ_X
    - Limited for $ICOM_X$

Contributions

- Better knowledge of problems
  - **APX**-Hardness of BD_X, $ICOM_X$ and $ICONS_X$
  - **NP**-Completeness of ZBD_E and ZBD_I
  - Polynomiality of ZBD_M
- Three new algorithms
  - An exact approach based on a transformation into a pseudo-boolean problem
    - Efficient approach for BD_X and ADJ_X
    - Limited for $ICOM_X$
  - IILCS$_{\boldsymbol{X}}$ heuristic and Hybrid method
    - Promising results on a real dataset for each problem

Perspectives

- Work on MATCH&WATCH
  - First experimentation on six chromosomes of γ-Proteobacteria
  - Analyze in details the common intervals obtained
  - Add functionalities according to biologists

Perspectives

- Work on MATCH&WATCH
  - First experimentation on six chromosomes of γ-Proteobacteria
  - Analyze in details the common intervals obtained
  - Add functionalities according to biologists
- Multi-chromosomal genome comparison
- Multiple genome comparison

Perspectives

- Work on MATCH&WATCH
  - First experimentation on six chromosomes of γ-Proteobacteria
  - Analyze in details the common intervals obtained
  - Add functionalities according to biologists
- Multi-chromosomal genome comparison
- Multiple genome comparison
- New algorithms
  - α-approximation for BD_E and BD_I when $\boldsymbol{occ(G_0) = 1}$?
  - α-approximation or PTAS for $ICOM_X$ on balanced genomes?

Perspectives

- Work on MATCH&WATCH
  - First experimentation on six chromosomes of γ-Proteobacteria
  - Analyze in details the common intervals obtained
  - Add functionalities according to biologists
- Multi-chromosomal genome comparison
- Multiple genome comparison
- New algorithms
  - α-approximation for BD_E and BD_I when $\boldsymbol{occ(G_0) = 1}$?
  - α-approximation or PTAS for $ICOM_X$ on balanced genomes?
- Partially ordered genomes

Acknowledgement

- *Directors*
  - Irena Rusu
  - Guillaume Fertin
- *Co-authors*
  - Damien Éveillard (LINA, Université de Nantes)
  - Annelyse Thévenin (LRI, Université Paris-Sud)
  - Stéphane Vialette (IGM, Université Paris-Est Marne-la-Vallée)

Pictures

- http://www.mun.ca/biology/scarr/FISH_chromosomes_300dpi.jpg
- http://agaudi.files.wordpress.com/2008/09/dna_overview_es.png
- http://joachimj.club.fr/imagesmada2004bis/PlanchePhylogeniedesprimates.jpg
- http://http://fr.wikipedia.org/wiki/Gene
- http://www.g-language.org/g3/

Acknowledgement

- *Directors*
  - Irena Rusu
  - Guillaume Fertin

Thank you

Merci

- *Co-authors*
  - Damien Éveillard (LINA, Université de Nantes)
  - Annelyse Thévenin (LRI, Université Paris-Sud)
  - Stéphane Vialette (IGM, Université Paris-Est Marne-la-Vallée)

Pictures

- `http://www.mun.ca/biology/scarr/FISH_chromosomes_300dpi.jpg`
- `http://agaudi.files.wordpress.com/2008/09/dna_overview_es.png`
- `http://joachimj.club.fr/imagesmada2004bis/PlanchePhylogeniedesprimates.jpg`
- `http://http://fr.wikipedia.org/wiki/Gene`
- `http://www.g-language.org/g3/`

Appendix

1. Appendix
   - Pseudo boolean transformation for other problems
   - ILCS$_\boldsymbol{X}$ and IILCS$_\boldsymbol{X}$
   - Visualization tool
   - Common intervals filtering
   - First experimental results

Appendix

1 Appendix
- Pseudo boolean transformation for other problems
- ILCS$_X$ and IILCS$_X$
- Visualization tool
- Common intervals filtering
- First experimental results

Transformation for *ICOM$_E$*: objective function

Objective:

$$\text{maximize} \sum_{\boldsymbol{k,l,m,n}} \boldsymbol{l_{k,l,m,n}}$$

Transformation for $ICOM_E$: objective function

Objective:

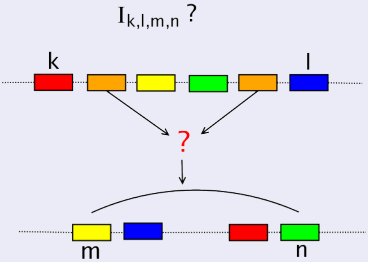
$$\text{maximize} \sum_{k,l,m,n} I_{k,l,m,n}$$

Improvements:

- Add rules to decrease the size of the instance

If all orange genes are located between the red and green one

We must have at least one orange gene to validate $I_{k,l,m,n}$



Transformation for $ICOM_E$: objective function

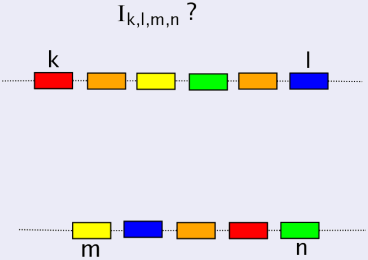
Objective:

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

Improvements:

- Add rules to decrease the size of the instance

Else, we do not generate variable $I_{k,l,m,n}$



Other problems ?

Other models

- C1: (Exemplar model)

$$\forall f \in \mathcal{F}_{G_0} \cup \mathcal{F}_{G_1}, \quad \sum_{\substack{1 \leqslant a \leqslant \eta_{G_0} \\ G_0[a]=f}} \sum_{\substack{1 \leqslant b \leqslant \eta_{G_1} \\ G_1[b]=f}} x_b^a = 1$$

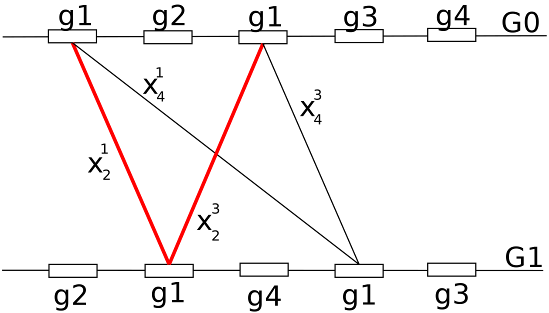
- C1': (Maximal matching model)

$$\forall f \in \mathcal{F}_{G_0} \cup \mathcal{F}_{G_1}, \quad \sum_{\substack{1 \leqslant a \leqslant \eta_{G_0} \\ G_0[a]=f}} \sum_{\substack{1 \leqslant b \leqslant \eta_{G_1} \\ G_1[b]=f}} x_b^a = \min\{occ(f, G_0), occ(f, G_1)\}$$

- C1": (Intermediate matching model)

$$\forall f \in \mathcal{F}_{G_0} \cup \mathcal{F}_{G_1}, \quad \sum_{\substack{1 \leqslant a \leqslant \eta_{G_0} \\ G_0[a]=f}} \sum_{\substack{1 \leqslant b \leqslant \eta_{G_1} \\ G_1[b]=f}} x_b^a \geqslant 1$$

Other models



- $\forall a = 1, 2, \ldots, \eta_{G_0}, \quad \sum_{\substack{1 \leqslant b \leqslant \eta_{G_1} \\ G_0[a] = G_1[b]}} x_b^a \leqslant 1$
- $\forall b = 1, 2, \ldots, \eta_{G_1}, \quad \sum_{\substack{1 \leqslant a \leqslant \eta_{G_0} \\ G_0[a] = G_1[b]}} x_b^a \leqslant 1$

Other problems ?

Other measures

- $ICONS_X$:
  Generate only variables $\boldsymbol{l_{k,l,m,n}}$ such that
  $\boldsymbol{(\ (\ G_0[k] = G_1[m] \wedge G_0[\ell] = G_1[n]\ ) \vee}$
  $\boldsymbol{(\ G_0[k] = -G_1[n] \wedge G_0[\ell] = -G_1[m]\ )\ )\}}$

Other problems ?

Other measures

- $ICONS_X$:
  Generate only variables $\boldsymbol{I_{k,l,m,n}}$ such that
  $\boldsymbol{(\ (\ G_0[k] = G_1[m] \wedge G_0[\ell] = G_1[n]\ ) \vee}$
  $\boldsymbol{(\ G_0[k] = -G_1[n] \wedge G_0[\ell] = -G_1[m]\ )\ )\}}$

- BD_X and ADJ_X:
  Other transformation

Appendix

1. Appendix
   - Pseudo boolean transformation for other problems
   - ILCS$_X$ and IILCS$_X$
   - Visualization tool
   - Common intervals filtering
   - First experimental results

ILCS$_M$ heuristic

LCS: Longest Common Substring [Tichy, 84]

1 2 3 **4 5 6 7**

6 7 4 5 1 6 3 2 1

ILCS$_M$ heuristic

LCS: Longest Common Substring [Tichy, 84]

1 2 3 **4 5 6 7**

6 7 4 5 1 6 3 2 1

ILCS$_M$ heuristic

Idea: Match genes of the LCS until saturation

ILCS$_M$ heuristic

LCS: Longest Common Substring [Tichy, 84]

1 2 3 **4 5 6 7**

6 7 4 5 1 6 3 2 1

ILCS$_M$ heuristic

Idea: Match genes of the LCS until saturation

1. Compute the Longest Common Substring S

ILCS$_M$ heuristic

LCS: Longest Common Substring [Tichy, 84]

1 2 3 **4 5 6 7**

6 7 4 5 1 6 3 2 1

ILCS$_M$ heuristic

Idea: Match genes of the LCS until saturation

1. Compute the Longest Common Substring S
2. Match all the genes of S accordingly

ILCS$_M$ heuristic

LCS: Longest Common Substring [Tichy, 84]

1 2 3 4 5 6 7

6 7 4 5 1 6 3 2 1

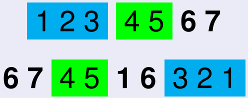
ILCS$_M$ heuristic

Idea: Match genes of the LCS until saturation

1. Compute the Longest Common Substring S
2. Match all the genes of S accordingly
3. Iterate the process until saturation

ILCS$_M$ heuristic

LCS: Longest Common Substring [Tichy, 84]



1 2 3 4 5 6 7

6 7 4 5 1 6 3 2 1

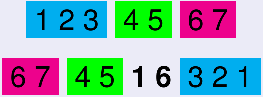
ILCS$_M$ heuristic

Idea: Match genes of the LCS until saturation

1. Compute the Longest Common Substring S
2. Match all the genes of S accordingly
3. Iterate the process until saturation

ILCS$_M$ heuristic

LCS: Longest Common Substring [Tichy, 84]



1 2 3 4 5 6 7

6 7 4 5 **1 6** 3 2 1

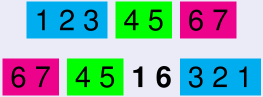
ILCS$_M$ heuristic

Idea: Match genes of the LCS until saturation

1. Compute the Longest Common Substring ***S***
2. Match all the genes of ***S*** accordingly
3. Iterate the process until saturation

ILCS$_M$ heuristic

LCS: Longest Common Substring [Tichy, 84]



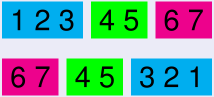
ILCS$_M$ heuristic

Idea: Match genes of the LCS until saturation

1. Compute the Longest Common Substring S
2. Match all the genes of S accordingly
3. Iterate the process until saturation
4. Remove all the genes that have not been matched

ILCS$_M$ heuristic

LCS: Longest Common Substring [Tichy, 84]



1 2 3 4 5 6 7

6 7 4 5 3 2 1

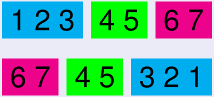
ILCS$_M$ heuristic

Idea: Match genes of the LCS until saturation

1. Compute the Longest Common Substring S
2. Match all the genes of S accordingly
3. Iterate the process until saturation
4. Remove all the genes that have not been matched

ILCS_M heuristic

LCS: Longest Common Substring [Tichy, 84]



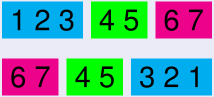
ILCS_M heuristic

Idea: Match genes of the LCS until saturation

1. Compute the Longest Common Substring S
2. Match all the genes of S accordingly
3. Iterate the process until saturation
4. Remove all the genes that have not been matched
5. Compute the number of common intervals

ILCS$_M$ heuristic

LCS: Longest Common Substring [Tichy, 84]



$\Rightarrow$ number of common intervals $= 19$

ILCS$_M$ heuristic

Idea: Match genes of the LCS until saturation

1. Compute the Longest Common Substring S
2. Match all the genes of S accordingly
3. Iterate the process until saturation
4. Remove all the genes that have not been matched
5. Compute the number of common intervals

IILCS$_M$ heuristic

LCS: Longest Common Substring [Tichy, 84]

1 2 3 **4 5 6 7**

6 7 4 5 1 6 3 2 1

IILCS$_M$ heuristic

Idea: Remove genes that cannot be matched

1. Compute the Longest Common Substring S
2. Match all the genes of S accordingly
3. Remove genes that cannot be matched

IILCS$_M$ heuristic

LCS: Longest Common Substring [Tichy, 84]

1 2 3 4 5 6 7

6 7 4 5 1 6 3 2 1

IILCS$_M$ heuristic

Idea: Remove genes that cannot be matched

1. Compute the Longest Common Substring S
2. Match all the genes of S accordingly
3. Remove genes that cannot be matched

IILCS$_M$ heuristic

LCS: Longest Common Substring [Tichy, 84]

1 2 3 4 5 6 7

6 7 4 5 6 3 2 1

IILCS$_M$ heuristic

Idea: Remove genes that cannot be matched

1. Compute the Longest Common Substring ***S***
2. Match all the genes of ***S*** accordingly
3. Remove genes that cannot be matched

IILCS$_M$ heuristic

LCS: Longest Common Substring [Tichy, 84]

1 2 3 4 5 6 7

6 7 4 5 6 3 2 1

IILCS$_M$ heuristic

Idea: Remove genes that cannot be matched

1. Compute the Longest Common Substring S
2. Match all the genes of S accordingly
3. Remove genes that cannot be matched
4. Iterate the process until saturation

IILCS$_M$ heuristic

LCS: Longest Common Substring [Tichy, 84]

1 2 3 4 5 6 **7**

6 7 4 5 6 3 2 1

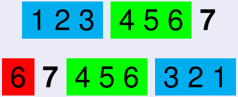
IILCS$_M$ heuristic

Idea: Remove genes that cannot be matched

1. Compute the Longest Common Substring S
2. Match all the genes of S accordingly
3. Remove genes that cannot be matched
4. Iterate the process until saturation

IILCS$_M$ heuristic

LCS: Longest Common Substring [Tichy, 84]



1 2 3 4 5 6 7

6 7 4 5 6 3 2 1

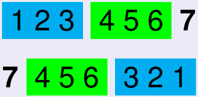
IILCS$_M$ heuristic

Idea: Remove genes that cannot be matched

1. Compute the Longest Common Substring S
2. Match all the genes of S accordingly
3. Remove genes that cannot be matched
4. Iterate the process until saturation

IILCS$_M$ heuristic

LCS: Longest Common Substring [Tichy, 84]



1 2 3 4 5 6 7

7 4 5 6 3 2 1

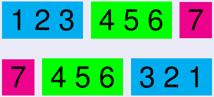
IILCS$_M$ heuristic

Idea: Remove genes that cannot be matched

1. Compute the Longest Common Substring ***S***
2. Match all the genes of ***S*** accordingly
3. Remove genes that cannot be matched
4. Iterate the process until saturation

IILCS$_M$ heuristic

LCS: Longest Common Substring [Tichy, 84]



1 2 3 4 5 6 7

7 4 5 6 3 2 1

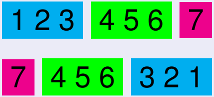
IILCS$_M$ heuristic

Idea: Remove genes that cannot be matched

1. Compute the Longest Common Substring S
2. Match all the genes of S accordingly
3. Remove genes that cannot be matched
4. Iterate the process until saturation

IILCS$_M$ heuristic

LCS: Longest Common Substring [Tichy, 84]



1 2 3 4 5 6 7

7 4 5 6 3 2 1

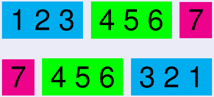
IILCS$_M$ heuristic

Idea: Remove genes that cannot be matched

1. Compute the Longest Common Substring ***S***
2. Match all the genes of ***S*** accordingly
3. Remove genes that cannot be matched
4. Iterate the process until saturation
5. Compute the number of common intervals

IILCS$_M$ heuristic

LCS: Longest Common Substring [Tichy, 84]



⇒ number of common intervals = 20

IILCS$_M$ heuristic

Idea: Remove genes that cannot be matched

1. Compute the Longest Common Substring S
2. Match all the genes of S accordingly
3. Remove genes that cannot be matched
4. Iterate the process until saturation
5. Compute the number of common intervals

Heuristics: adaptation for other models

exemplar model

- For each gene family, we keep only the first occurrence in an LCS
- At each iteration, we remove all genes that cannot be matched

Heuristics: adaptation for other models

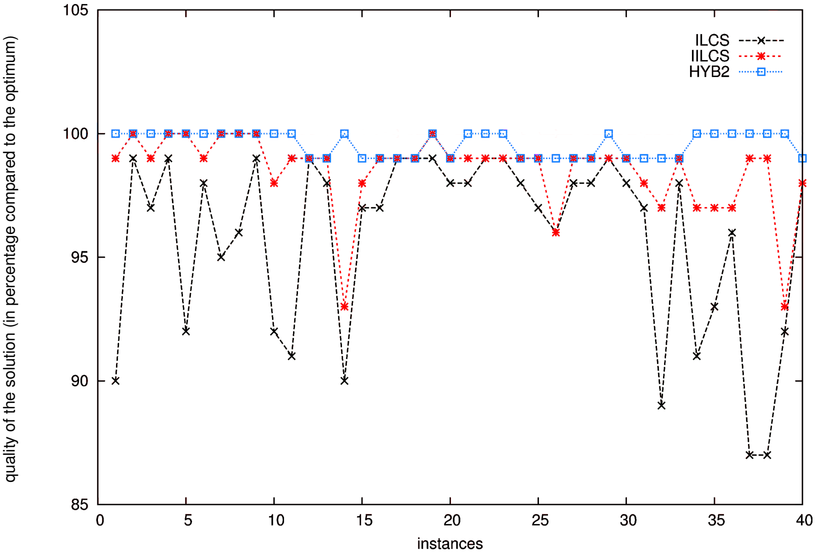
exemplar model

- For each gene family, we keep only the first occurrence in an LCS
- At each iteration, we remove all genes that cannot be matched

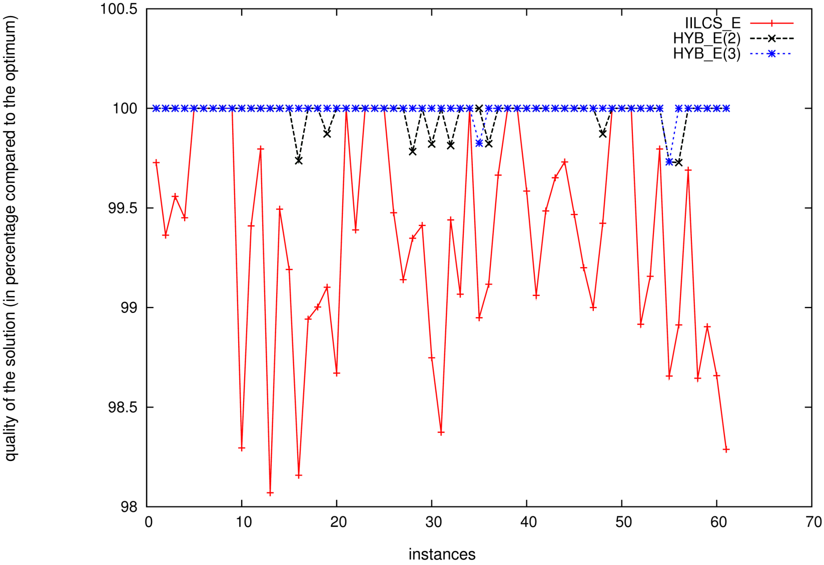
intermediate model

- We stop if, for each gene family, there exists at least one occurrence in the matching

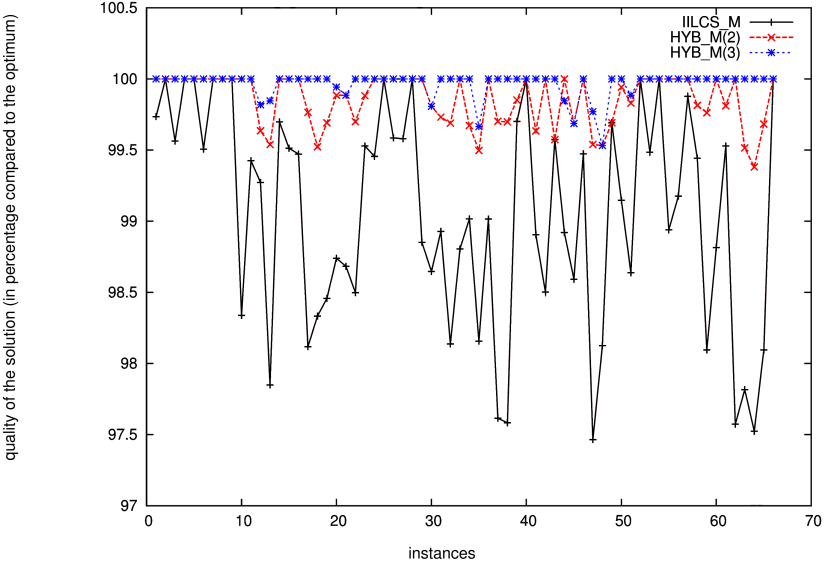
Experimental results: $ICOM_M$



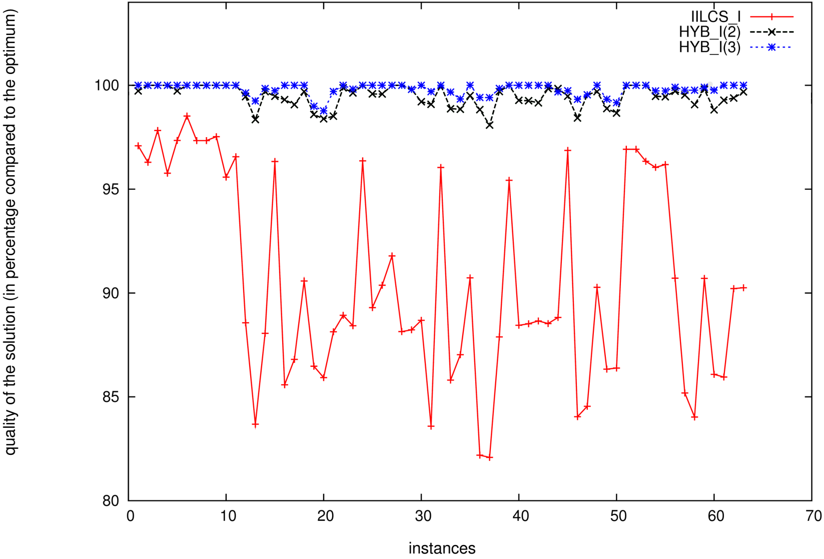
Experimental results: ADJ_E



Experimental results: ADJ_M



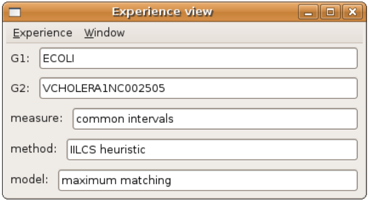
Experimental results: ADJ_I



Appendix

1 Appendix

- Pseudo boolean transformation for other problems
- ILCS_X and IILCS_X
- Visualization tool
- Common intervals filtering
- First experimental results



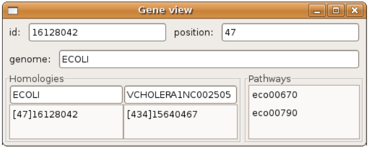
G1: ECOLI

G2: VCHOLERA1NC002505

measure: common intervals

method: IILCS heuristic

model: maximum matching



id: 16128042

position: 47

genome: ECOLI

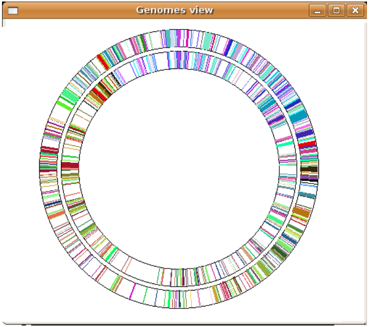
Homologies

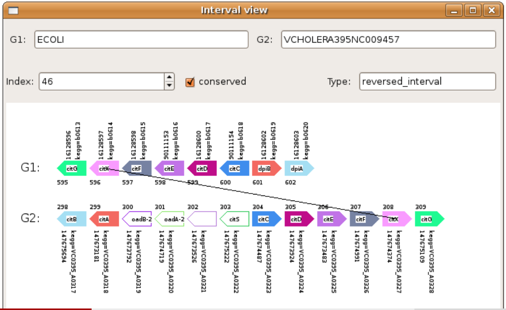
ECOLI	VCHOLERA1NC002505
[47]16128042	[434]15640467

Pathways

eco00670

eco00790





G1: ECOLI

G2: VCHOLERA395NC009457

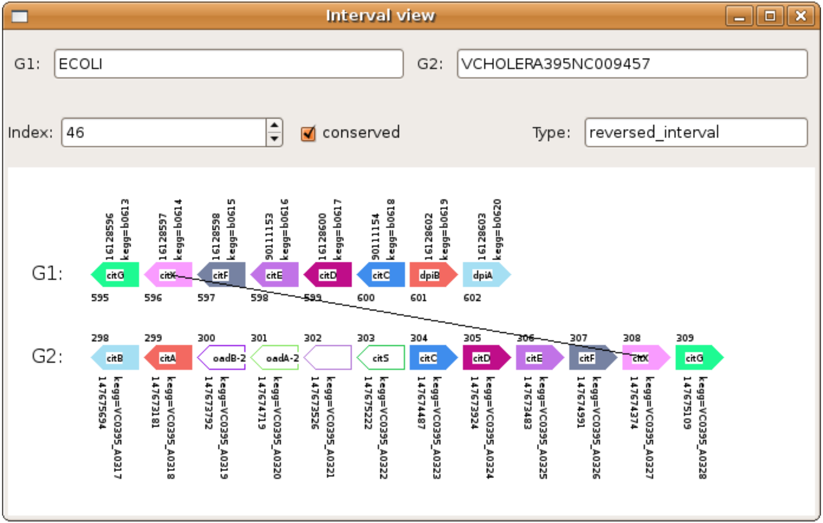
Index: 46

☑ conserved

Type: reversed_interval

G1:
16128596 kegg=b0613
16128597 kegg=b0614
16128598 kegg=b0615
90111153 kegg=b0616
16128600 kegg=b0617
90111154 kegg=b0618
16128602 kegg=b0619
16128603 kegg=b0620
citG
citX
citF
citE
citD
citC
dpiB
dpiA
595
596
597
598
599
600
601
602
G2:
298
299
300
301
302
303
304
305
306
307
308
309
citB
citA
oadB-2
oadA-2
citS
citC
citD
citE
citF
citX
citG
14675694 kegg=VC0395_A0317
14673181 kegg=VC0395_A0318
14673792 kegg=VC0395_A0319
14674719 kegg=VC0395_A0320
14673526 kegg=VC0395_A0321
14675222 kegg=VC0395_A0322
14674487 kegg=VC0395_A0323
14673924 kegg=VC0395_A0324
14673483 kegg=VC0395_A0325
14674991 kegg=VC0395_A0326
14674374 kegg=VC0395_A0327
14675109 kegg=VC0395_A0328

Visualization



Interval view

G1: ECOLI

G2: VCHOLERA395NC009457

Index: 46

☑ conserved

Type: reversed_interval

G1:
16128596 kegg=b0613
16128597 kegg=b0614
16128598 kegg=b0615
90111153 kegg=b0616
16128600 kegg=b0617
90111154 kegg=b0618
16128602 kegg=b0619
16128603 kegg=b0620
citG
citX
citF
citE
citD
citC
dpiB
dpiA
595
596
597
598
599
600
601
602
G2:
298
299
300
301
302
303
304
305
306
307
308
309
citB
citA
oadB-2
oadA-2
citS
citC
citD
citE
citF
citX
citG
147675694 kegg=VC0395_A0317
147673181 kegg=VC0395_A0318
147673792 kegg=VC0395_A0319
147674719 kegg=VC0395_A0320
147673526 kegg=VC0395_A0321
147675222 kegg=VC0395_A0322
147674487 kegg=VC0395_A0323
147673924 kegg=VC0395_A0324
147673483 kegg=VC0395_A0325
147674991 kegg=VC0395_A0326
147674374 kegg=VC0395_A0327
147675109 kegg=VC0395_A0328

Appendix

1 Appendix

- Pseudo boolean transformation for other problems
- ILCS$_X$ and IILCS$_X$
- Visualization tool
- Common intervals filtering
- First experimental results

Common intervals filtering

- Lots of common intervals
- Relevance of common intervals ?

⇒ Three filters to emphasize *the most interresting* common intervals

Common intervals filtering

- Lots of common intervals
- Relevance of common intervals ?

⇒ Three filters to emphasize *the most interresting* common intervals

Filters

1. **Maximal common intervals**:
   Select only common intervals that are not contained in another one

Common intervals filtering

- Lots of common intervals
- Relevance of common intervals ?

⇒ Three filters to emphasize *the most interresting* common intervals

Filters

1. **Maximal common intervals**:
   Select only common intervals that are not contained in another one
2. **Annotated common intervals**:
   Select maximal common intervals that contain some annotations in the *Ecocyc database*

Common intervals filtering

- Lots of common intervals
- Relevance of common intervals ?

⇒ Three filters to emphasize *the most interresting* common intervals

Filters

1. **Maximal common intervals**:
   Select only common intervals that are not contained in another one
2. **Annotated common intervals**:
   Select maximal common intervals that contain some annotations in the *Ecocyc database*
3. **Relevant common intervals**:
   Select annotated common intervals with good *p-value* (obtained by `GO-TermFinder`)

Common intervals filtering

- Lots of common intervals
- Relevance of common intervals ?

⇒ Three filters to emphasize *the most interresting* common intervals

Filters

1. **Maximal common intervals**:
   Select only common intervals that are not contained in another one
2. **Annotated common intervals**:
   Select maximal common intervals that contain some annotations in the *Ecocyc database*
3. **Relevant common intervals**:
   Select annotated common intervals with good *p-value* (obtained by `GO−TermFinder`)

Appendix

1. Appendix
   - Pseudo boolean transformation for other problems
   - ILCS$_X$ and IILCS$_X$
   - Visualization tool
   - Common intervals filtering
   - First experimental results

Experimental results

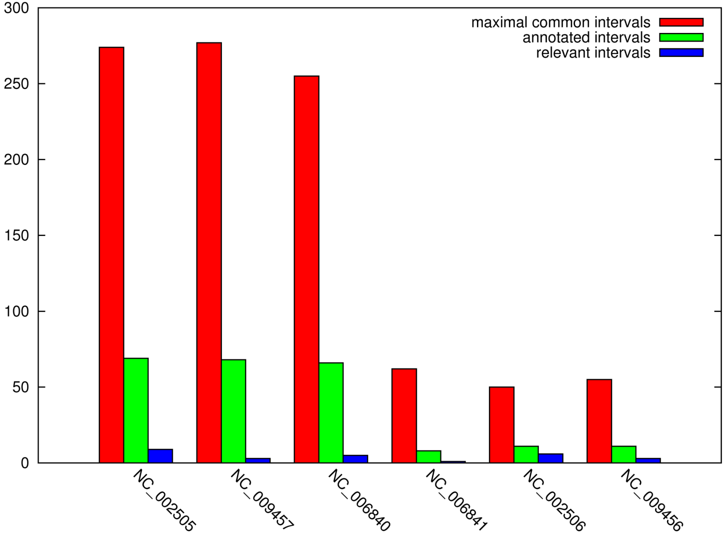
Input : six chromosomes of γ-Proteobacteria

NCBI identifiant	Name
NC_000913	*Escherichia coli* K12
NC_002505	*Vibrio cholerae* 01 biovar eltor str. N16961 chromosome I
NC_002506	*Vibrio cholerae* 01 biovar eltor str. N16961 chromosome II
NC_009456	*Vibrio cholerae* 0395 chromosome I
NC_009457	*Vibrio cholerae* 0395 chromosome II
NC_006840	*Vibrio fischeri* ES114 chromosome I
NC_006841	*Vibrio fischeri* ES114 chromosome II

Results: common intervals

	genome size			computational time		common intervals	
genome G_2	*E. coli*	G_2	method	Inparanoid (s)	matching (s)	number	maximal
***NC*002505**	**4243**	**2742**	***IILCS***	1144	15	**7418**	**274**
***NC*002506**	**4243**	**1093**	***PSB***	638	41	**246**	**50**
***NC*009456**	**4243**	**1133**	***PSB***	651	46	**264**	**55**
***NC*009457**	**4243**	**2742**	***IILCS***	1199	18	**7204**	**278**
***NC*006840**	**4243**	**2586**	***IILCS***	1012	1	**3865**	**255**
***NC*006841**	**4243**	**1175**	***IILCS***	715	1	**203**	**62**

Experimental results


maximal common intervals
annotated intervals
relevant intervals
NC_002505
NC_009457
NC_006840
NC_006841
NC_002506
NC_009456