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’Informatique de Nantes Atlantique,
UMR CNRS 6241, UFR de Sciences et Techniques de Nantes

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 several *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
  - duplication
  - translocation
  - ... (more operations can be considered)
- **Find a most parsimonious rearrangement scenario**

<table>
<thead>
<tr>
<th>Comparison based on the structure of genomes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Compare the structure (genes order) of the two genomes</td>
</tr>
<tr>
<td>Compute a (dis)similarity measure between genomes</td>
</tr>
<tr>
<td>- number of breakpoints/adjacencies</td>
</tr>
<tr>
<td>- number of common intervals</td>
</tr>
<tr>
<td>- number of conserved intervals</td>
</tr>
<tr>
<td>- Sum Adjacency Disruption</td>
</tr>
<tr>
<td>- ... (more measures can be considered)</td>
</tr>
</tbody>
</table>
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
  - translocation...
- 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
  - duplication
  - translocation
  - ...  
- 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
  - ...

Angibaud Sébastien
Phd Thesis - Defense  
October 7th 2009
Genomes representation

Representation and notations

1. Unichromosomal genome: sequence of signed genes

Example

1. $G_0 = +1 + 2 - 3 - 7 + 4 + 5 + 7 - 8 + 10 - 9 + 4 - 6 - 4$
Genomes representation

Representation and notations

1. Unichromosomal genome: sequence of *signed genes*
2. Alphabet $\Sigma \leftrightarrow gene families$

Example

1. $G_0 = +1 + 2 - 3 -7 + 4 + 5 +7 - 8 + 10 - 9 + 4 - 6 - 4$
2. $\Sigma = \{1, 2, 3 \ldots 10\}$
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. $G_0 = +1 + 2 - 3 -7 + 4 + 5 + 7 - 8 + 10 - 9 + 4 - 6 - 4$
2. $\Sigma = \{1, 2, 3 \ldots 10\}$
## 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 $G_0$
4. Let $occ(G_0)$ be the maximum number of genes in a gene family

### Example

1. $G_0 = +1 + 2 - 3 - 7 +4 + 5 + 7 - 8 + 10 - 9 +4 - 6 -4$
2. $\Sigma = \{1, 2, 3 \ldots 10\}$
4. $occ(G_0) = 3$
## 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 $G_0$
4. Let $\text{occ}(G_0)$ be the maximum number of genes in a gene family
5. Let $\eta_{G_0}$ be the number of genes in $G_0$

### Example

1. $G_0 = +1 + 2 - 3 - 7 + 4 + 5 + 7 - 8 + 10 - 9 + 4 - 6 - 4$
2. $\Sigma = \{1, 2, 3 \ldots 10\}$
4. $\text{occ}(G_0) = 3$
5. $\eta_{G_0} = 13$
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 $G_0[p]$ and $G_0[p+1]$ iff $(G_0[p], G_0[p+1])$ or $(-G_0[p+1], -G_0[p])$ appears as a pair of consecutive genes in $G_1$.

\[
G_0 = +1 + 2 + 3 + 4 + 5 \\
G_1 = +3 + 4 - 5 - 2 - 1
\]
Breakpoint and adjacency

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

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

\[
G_0 = +1 + 2 + 3 + 4 + 5
\]
\[
G_1 = -4 - 3 - 5 +1 + 2
\]

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

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

\[
G_0 = +1 + 2 + 3 + 4 + 5 \\
G_1 = -4 - 3 - 5 + 1 + 2
\]
Breakpoint and adjacency

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

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

\[
G_0 = +1 + 2 \, \text{Adjacency} \, +3 + 4 \, \text{Adjacency} + 5
\]

\[
G_1 = -4 - 3 - 5 + 1 + 2
\]
Breakpoint and adjacency

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

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

\[
\begin{align*}
G_0 &= +0\,\uparrow+1\,\uparrow+2\,\downarrow+3\,\downarrow+4\,\uparrow+5\,\downarrow+6 \\
G_1 &= +0\,\uparrow+1\,\downarrow+2
\end{align*}
\]
Breakpoint and adjacency

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

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

Two measures:

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

\[
\begin{align*}
G_0 &= +0 \text{ Adjacency} + 1 \text{ Adjacency} + 2 \text{ Adjacency} + 3 \text{ Adjacency} + 4 \text{ Adjacency} + 5 \text{ Adjacency} + 6 \\
G_1 &= +0 \text{ Adjacency} - 4 \text{ Adjacency} - 3 \text{ Adjacency} - 5 \text{ Adjacency} + 1 \text{ Adjacency} + 2 \text{ Adjacency} + 6
\end{align*}
\]
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)

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

\[
G_0 = +1 + 2 + 3 + 4 + 5 \quad G_1 = +2 - 4 + 3 + 5 + 1
\]

\[
\Rightarrow s_0 = +3 + 4 + 5 \quad s_1 = - 4 + 3 + 5
\]

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)

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

$$\Rightarrow s_0 = +3 + 4 + 5 \quad s_1 = -4 + 3 + 5$$

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

\[ G_0 = +0 +1 +2 + 3 + 4 + 5 \]
\[ G_1 = -4 - 3 - 5 +0 -1+2 \]
Conserved interval

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

- common interval
- same extremities OR **reversed extremities**

\[
G_0 = +0 + 1 + 2 + 3 + 4 + 5
\]
\[
G_1 = -4 - 3 - 5 + 0 - 1 + 2
\]
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 \( \mathcal{M} \)

\[
G_0 = +0 +1 -2 -1 -3 +4
\]

\[
G_1 = +0 -1 +2 -1 -3 -1 +4
\]
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 $\mathcal{M}$

\[
\begin{align*}
G_0 &= +0 +1 -2 -1 -3 +4 \\
G_1 &= +0 -1 +2 -1 -3 -1 +4
\end{align*}
\]
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 $\mathcal{M}$

\[
G_0^E = +0 +1 -2 -3 +4
\]

\[
G_1^E = +0 +2 -1 -3 +4
\]
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 $\mathcal{M}$

\[
\begin{align*}
G_0^E &= +0^\uparrow +1 - 2^\downarrow - 3 + 4 \\
G_1^E &= +0 + 2 - 1 - 3 + 4 \\
Bkp(G_0^E, G_1^E) &= 2
\end{align*}
\]
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 \( \mathcal{M} \)

\[
\begin{align*}
G_0^E &= +0 \downarrow +1 - 2 \downarrow - 3 + 4 \\
G_1^E &= +0 + 2 - 1 - 3 + 4 \\
\text{Bkp}(G_0^E, G_1^E) &= 2
\end{align*}
\]

**Maximum matching model (M)**

[Tang & al, 03]

a maximum number of occurrences in \( \mathcal{M} \)

\[
\begin{align*}
G_0 &= +0 + 1 - 2 - 1 - 3 + 4 \\
G_1 &= +0 - 1 + 2 - 1 - 3 - 1 + 4
\end{align*}
\]
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 $\mathcal{M}$

$$
G^E_0 = +0^\uparrow +1 - 2^\downarrow - 3 + 4
$$
$$
G^E_1 = +0 + 2 - 1 - 3 + 4
$$
$$
\text{Bkp}(G^E_0, G^E_1) = 2
$$

**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 $\mathcal{M}$

- $G_0^E = +0^\uparrow +1 - 2^\downarrow - 3 + 4$
- $G_1^E = +0 + 2 - 1 - 3 + 4$
- $\text{Bkp}(G_0^E, G_1^E) = 2$

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

$$G_0^E = +0^\uparrow +1 - 2^\downarrow - 3 + 4$$
$$G_1^E = +0 + 2 - 1 - 3 + 4$$

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

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

$$G_0^M = +0^\uparrow +1'^\downarrow - 2^\downarrow - 1'' - 3 + 4$$
$$G_1^M = +0 - 1' + 2 - 1'' - 3 + 4$$

$\text{Bkp}(G_0^M, G_1^M) = 3$
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 \( \mathcal{M} \)

\[
\begin{align*}
G_0 &= +0 \, +1 \, - 2 \, -1 \, - 3 \, + 4 \\
G_1 &= +0 \, -1 \, + 2 \, -1 \, - 3 \, -1 \, + 4
\end{align*}
\]
Several possible matchings?

*maximum matching model* (\(\mathbf{M}\))

[Tang & al, 03]

a maximum number of occurrences in \(\mathcal{M}\)

\[
G_0 = +0 \; \boxed{+1} \; - \; 2 \; \boxed{-1} \; - \; 3 \; + \; 4
\]

\[
G_1 = +0 \; \boxed{-1} \; + \; 2 \; \boxed{-1} \; - \; 3 \; \boxed{-1} \; + \; 4
\]
Measure between genomes with duplicates

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 $G_1^X$
Measure between genomes with duplicates

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 $G_1^X$

<table>
<thead>
<tr>
<th>measure</th>
<th>problem</th>
</tr>
</thead>
<tbody>
<tr>
<td>common interval</td>
<td>$ICOM_X$</td>
</tr>
<tr>
<td>conserved interval</td>
<td>$ICONS_X$</td>
</tr>
<tr>
<td>breakpoint</td>
<td>$BD_X$</td>
</tr>
<tr>
<td>adjacency</td>
<td>$ADJ_X$</td>
</tr>
</tbody>
</table>
# Measure between genomes with duplicates

**Problem**

- **Input:**
  - Two genomes $G_0$ and $G_1$
  - A model $X \in \{E, M, I\}$

- **Question:** Are there $G_0^X$ and $G_1^X$ which satisfy the model $X$, and which imply no breakpoint?

<table>
<thead>
<tr>
<th>measure</th>
<th>problem</th>
</tr>
</thead>
<tbody>
<tr>
<td>common interval</td>
<td>$ICOM_X$</td>
</tr>
<tr>
<td>conserved interval</td>
<td>$ICONS_X$</td>
</tr>
<tr>
<td>breakpoint</td>
<td>$BD_X$</td>
</tr>
<tr>
<td>adjacency</td>
<td>$ADJ_X$</td>
</tr>
<tr>
<td></td>
<td>$ZBD_X$</td>
</tr>
</tbody>
</table>
Theoretical complexity results

Outline

1. Genomes comparison
2. Theoretical complexity results
3. Algorithms
4. MATCH&WATCH application
5. Conclusion
What do we know?

<table>
<thead>
<tr>
<th>exemplar model</th>
<th>maximum matching model</th>
<th>intermediate model</th>
</tr>
</thead>
<tbody>
<tr>
<td>$ICOM_X$</td>
<td>$ICONS_X$</td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>NP</strong>-Complete [Chauve et al.] (instance $(1, 2)$)</td>
<td></td>
</tr>
<tr>
<td>$BD_X$</td>
<td></td>
<td><strong>NP</strong>-Complete [Bryant] (instance $(1, 2)$)</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>NP</strong>-Complete [Blin et al.] *</td>
</tr>
<tr>
<td>$ZBD_X$</td>
<td><strong>NP</strong>-Complete [Chen et al.] (instance $(3, 3)$)</td>
<td>?</td>
</tr>
</tbody>
</table>

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

* only one family contains several occurrences
**α-approximation** and **PTAS**

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

**α-approximation and PTAS**

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

- A polynomial algorithm $B$ is a *Polynomial-Time Approximation Scheme* (PTAS) iff $\forall \epsilon > 0$
  - If $P$ is a problem of minimization, then $B(I) \leq (1 + \epsilon) \cdot \text{optimal}(I)$
  - If $P$ is a problem of maximization, then $B(I) \geq \frac{1}{1+\epsilon} \cdot \text{optimal}(I)$
**Definition**

### α-approximation and PTAS

- Let \( P \) be an optimization problem
- Let \( I \) be an instance of \( P \)
- A polynomial algorithm \( A \) is an \( \alpha \)-approximation iff
  - If \( P \) is a problem of minimization, then \( A(I) \leq \alpha \cdot \text{optimal}(I) \)
  - If \( P \) is a problem of maximization, then \( A(I) \geq \frac{1}{\alpha} \cdot \text{optimal}(I) \)

- A polynomial algorithm \( B \) is a *Polynomial-Time Approximation Scheme* (PTAS) iff \( \forall \epsilon > 0 \)
  - If \( P \) is a problem of minimization, then \( B(I) \leq (1 + \epsilon) \cdot \text{optimal}(I) \)
  - If \( P \) is a problem of maximization, then \( B(I) \geq \frac{1}{1+\epsilon} \cdot \text{optimal}(I) \)

### APX-Hard Class

- If a problem \( P \) is APX-Hard then \( P \) does not admit a PTAS
### New results

<table>
<thead>
<tr>
<th>exemplar model</th>
<th>maximum matching model</th>
<th>intermediate model</th>
</tr>
</thead>
<tbody>
<tr>
<td>$ICOM_X$</td>
<td><strong>NP</strong>-Complete [Chauve et al.] (instance $(1, 2)$)</td>
<td><strong>APX</strong>-Hard (instance $(1, 2)$) *</td>
</tr>
<tr>
<td>$ICONS_X$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$BD_X$</td>
<td><strong>NP</strong>-Complete [Bryant] (instance $(1, 2)$)</td>
<td><strong>NP</strong>-Complete [Blin et al.]</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>APX</strong>-Hard (instance $(1, 2)$) *</td>
</tr>
<tr>
<td>$ZBD_X$</td>
<td><strong>NP</strong>-Complete [Chen et al.] (instance $(3, 3)$)</td>
<td>polynomial *</td>
</tr>
<tr>
<td></td>
<td>(instance $(2, k)$) *</td>
<td></td>
</tr>
<tr>
<td></td>
<td>[Blin et al.] (instance $(2, 2)$)</td>
<td></td>
</tr>
<tr>
<td>$ADJ_X$</td>
<td>$ADJ_E \simeq BD_E$ *</td>
<td>$ADJ_M \simeq BD_M$ *</td>
</tr>
</tbody>
</table>

* S. Angibaud, G. Fertin, I. Rusu, A. Thévenin et et S. Vialette
On the Approximability of Comparing Genomes with Duplicates
## New results

<table>
<thead>
<tr>
<th>exemplar model</th>
<th>maximum matching model</th>
<th>intermediate model</th>
</tr>
</thead>
<tbody>
<tr>
<td>$ICOM_X$</td>
<td><strong>NP</strong>-Complete [Chauve et al.] (instance $(1, 2)$)</td>
<td></td>
</tr>
<tr>
<td>$ICONS_X$</td>
<td><strong>APX</strong>-Hard (instance $(1, 2)$)</td>
<td></td>
</tr>
<tr>
<td>$BD_X$</td>
<td><strong>NP</strong>-Complete [Bryant] (instance $(1, 2)$)</td>
<td><strong>NP</strong>-Complete [Blin et al.]</td>
</tr>
<tr>
<td></td>
<td><strong>APX</strong>-Hard (instance $(1, 2)$)</td>
<td></td>
</tr>
<tr>
<td>$ZBD_X$</td>
<td><strong>NP</strong>-Complete [Chen et al.] (instance $(3, 3)$)</td>
<td>polynomial</td>
</tr>
<tr>
<td></td>
<td>(instance $(2, k)$)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>[Blin et al.] (instance $(2, 2)$)</td>
<td></td>
</tr>
<tr>
<td>$ADJ_X$</td>
<td>$ADJ_E \simeq BD_E$</td>
<td>$ADJ_M \simeq BD_M$</td>
</tr>
</tbody>
</table>

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

$A \neq B$: An optimal solution for $A$ is not necessarily an optimal solution for $B$
## New results

<table>
<thead>
<tr>
<th>Exemplar model</th>
<th>Maximum matching model</th>
<th>Intermediate model</th>
</tr>
</thead>
<tbody>
<tr>
<td>ICOM&lt;sub&gt;x&lt;/sub&gt;</td>
<td><strong>NP</strong>-Complete [Chauve et al.] (instance (1, 2))</td>
<td><strong>NP</strong>-Complete [Chauve et al.] (instance (1, 2))</td>
</tr>
<tr>
<td>ICONS&lt;sub&gt;x&lt;/sub&gt;</td>
<td><strong>APX</strong>-Hard (instance (1, 2))</td>
<td><strong>APX</strong>-Hard (instance (1, 2))</td>
</tr>
<tr>
<td>BD&lt;sub&gt;x&lt;/sub&gt;</td>
<td><strong>NP</strong>-Complete [Bryant] (instance (1, 2))</td>
<td><strong>NP</strong>-Complete [Bryant] (instance (1, 2))</td>
</tr>
<tr>
<td>ZBD&lt;sub&gt;x&lt;/sub&gt;</td>
<td><strong>NP</strong>-Complete [Chen et al.] (instance (3, 3))</td>
<td>polynomial</td>
</tr>
<tr>
<td>[instance (2, k)]</td>
<td>[Blin et al.] (instance (2, 2))</td>
<td></td>
</tr>
<tr>
<td>ADJ&lt;sub&gt;x&lt;/sub&gt;</td>
<td>( \text{ADJ}_E \simeq \text{BD}_E )</td>
<td>( \text{ADJ}_M \simeq \text{BD}_M )</td>
</tr>
</tbody>
</table>

\( A \simeq B \): An optimal solution for \( A \) is an optimal solution for \( B \)

\( A \neq B \): An optimal solution for \( A \) is not necessarily an optimal solution for \( B \)
### New results

<table>
<thead>
<tr>
<th>exemplar model</th>
<th>maximum matching model</th>
<th>intermediate model</th>
</tr>
</thead>
<tbody>
<tr>
<td>$ICOM_X$</td>
<td><strong>NP</strong>-Complete [Chauve et al.] (instance $(1, 2)$)</td>
<td><strong>APX</strong>-Hard (instance $(1, 2)$)</td>
</tr>
<tr>
<td>$ICONS_X$</td>
<td><strong>NP</strong>-Complete [Bryant] (instance $(1, 2)$)</td>
<td><strong>NP</strong>-Complete [Blin et al.]</td>
</tr>
<tr>
<td>$BD_X$</td>
<td><strong>NP</strong>-Complete [Bryant] (instance $(1, 2)$)</td>
<td><strong>NP</strong>-Complete [Blin et al.]</td>
</tr>
<tr>
<td>$ZBD_X$</td>
<td><strong>NP</strong>-Complete [Chen et al.] (instance $(3, 3)$) (instance $(2, k)$) [Blin et al.] (instance $(2, 2)$)</td>
<td>polynomial</td>
</tr>
<tr>
<td>$ADJ_X$</td>
<td>$ADJ_E \simeq BD_E$</td>
<td>$ADJ_M \simeq BD_M$</td>
</tr>
</tbody>
</table>

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

#### New results

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

$\Rightarrow$ Bad news: $BD_E$ and $BD_I$ do not admit any $\alpha$-approximation, unless $P = NP$
# New results

<table>
<thead>
<tr>
<th>exemplar model</th>
<th>maximum matching model</th>
<th>intermediate model</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\text{ICOM}_X$</td>
<td>$\text{NP}$-Complete [Chauve et al.] (instance $(1, 2)$)</td>
<td></td>
</tr>
<tr>
<td>$\text{ICONS}_X$</td>
<td>$\text{APX}$-Hard (instance $(1, 2)$)</td>
<td></td>
</tr>
<tr>
<td>$\text{BD}_X$</td>
<td>$\text{NP}$-Complete [Bryant] (instance $(1, 2)$)</td>
<td>$\text{NP}$-Complete [Blin et al.]</td>
</tr>
<tr>
<td>$\text{ZBD}_X$</td>
<td>$\text{NP}$-Complete [Chen et al.] (instance $(3, 3)$)</td>
<td>polynomial</td>
</tr>
<tr>
<td></td>
<td>(instance $(2, k)$)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>[Blin et al.] (instance $(2, 2)$)</td>
<td></td>
</tr>
<tr>
<td>$\text{ADJ}_X$</td>
<td>$\text{ADJ}_E \simeq \text{BD}_E$</td>
<td>$\text{ADJ}_M \simeq \text{BD}_M$</td>
</tr>
</tbody>
</table>

$\Rightarrow$ Good news: $\text{BD}_M$ could admit an $\alpha$-approximation
Outline

1. Genomes comparison
2. Theoretical complexity results
3. Algorithms
   - Exact approach
     - Pseudo boolean problem
     - Pseudo-boolean transformation for $ICOM_E$
     - Experimental results
   - Heuristics and hybrid method
     - 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**:
  - \( x + 2 \cdot y \geq 2 \)
  - \( z + y \leq 1 \)
- **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**:
  - $x + 2 \cdot y \geq 2$
  - $z + y \leq 1$
- **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**:
  - $x + 2 \cdot y \geq 2$
  - $z + y \leq 1$
- **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**:
  - $x + 2 \cdot y \geq 2$
  - $z + y \leq 1$
- **Objective function**: maximize $x + 2 \cdot y - z$

⇒ Powerful solvers for this type of problem
Transformation for $ICOM_E$: variables

Variables $x$ and $l$: 

![Diagram showing variables and networks for $ICOM_E$]
Transformation for $ICOM_E$: variables

**Variables $x$ and $I$:**

$x^a_0$ true $\iff$ gene $G_0[a]$ and $G_1[b]$ are matched
Transformation for $ICOM_E$: variables

Variables $x$ and $I$:

$$l_{k,l,m,n} \text{ true } \iff [k, l] \text{ in } G_0 \text{ is a common interval of } (G_0, G_1), \text{ and } [m, n] \text{ in } G_1 \text{ is a permutation of } [k, l]$$
Transformation for $ICOM_E$: constraints

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

C1: $\forall f \in \mathcal{F}_{G_0} \cup \mathcal{F}_{G_1}$,
\[
\sum_{1 \leq a \leq \eta_{G_0}} x^a_{G_0[a]=f} \quad \sum_{1 \leq b \leq \eta_{G_1}} x^b_{G_1[b]=f} = 1
\]
Validity of variables $I_{k,l,m,n}$

$\mathbf{g}_1 \quad \mathbf{g}_2 \quad \mathbf{g}_1 \mathbf{g}_3 \quad \mathbf{g}_4 \quad \mathbf{G}_0$

$\mathbf{g}_2 \quad \mathbf{g}_1 \quad \mathbf{g}_4 \quad \mathbf{g}_1 \quad \mathbf{g}_3 \quad \mathbf{G}_1$

$I_{k,l,m,n} + x_2^3 \leq 1$
Transformation for $ICOM_E$

Objective function:

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

Variables:

$\mathcal{I} = \{l_k,l,m,n : 1 \leq k \leq \ell \leq \eta_{G_0} \wedge 1 \leq m \leq n \leq \eta_{G_1}\}$

$\mathcal{X} = \{x_a^b : 1 \leq a \leq \eta_{G_0} \wedge 1 \leq b \leq \eta_{G_1} \wedge G_0[a] = G_1[b]\}$

Constraints:

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

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

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

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

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

\((C.06)\) $\forall l_k,l,m,n \in \mathcal{I}, \ 4 l_k,l,m,n - \sum_{m \leq r \leq n} x_r^k - \sum_{m \leq s \leq n} x_s^\ell - \sum_{k \leq p \leq \ell} x_m^p - \sum_{k \leq q \leq \ell} x_n^q \leq 0$

Objective function:

Maximize $\sum_{k,l,m,n} l_k,l,m,n$
Transformation for $ICOM_E$

Variables:

$\mathcal{I} = \{I_k,l,m,n : 1 \leq k \leq \ell \leq \eta_{G_0} \land 1 \leq m \leq n \leq \eta_{G_1}\}$

$\mathcal{X} = \{x^a_b : 1 \leq a \leq \eta_{G_0} \land 1 \leq b \leq \eta_{G_1} \land G_0[a] = G_1[b]\}$

Constraints:

(C.01) $\forall f \in \mathcal{F}_{G_0} \cup \mathcal{F}_{G_1}$, \[ \sum_{1 \leq a \leq \eta_{G_0}, G_0[a]=f} x^a_a = 1 \]
\[ \sum_{1 \leq b \leq \eta_{G_1}, G_1[b]=f} x^b_b = 1 \]

(C.02) $\forall I_k,l,m,n \in \mathcal{I}$, $\forall k < p \leq \ell$, $\forall 1 \leq r < m$, $G_0[p] = G_1[r]$, $l_k,l,m,n + x^P_r \leq 1$

(C.03) $\forall I_k,l,m,n \in \mathcal{I}$, $\forall k < p \leq \ell$, $\forall n < r \leq \eta_{G_1}$, $G_0[p] = G_1[r]$, $l_k,l,m,n + x^P_r \leq 1$

(C.04) $\forall I_k,l,m,n \in \mathcal{I}$, $\forall m < r \leq n$, $\forall 1 \leq p < k$, $G_0[p] = G_1[r]$, $l_k,l,m,n + x^P_r \leq 1$

(C.05) $\forall I_k,l,m,n \in \mathcal{I}$, $\forall m < r \leq n$, $\forall \ell < p \leq \eta_{G_0}$, $G_0[p] = G_1[r]$, $l_k,l,m,n + x^P_r \leq 1$

(C.06) $\forall I_k,l,m,n \in \mathcal{I}$, \[ 4l_k,l,m,n - \sum_{m \leq r \leq n, G_0[k]=G_1[r]} x^k_r - \sum_{m \leq s \leq n, G_0[\ell]=G_1[s]} x^\ell_s - \sum_{k \leq p \leq \ell, G_0[p]=G_1[m]} x^P_m - \sum_{k \leq q \leq \ell, G_0[q]=G_1[n]} x^q_n \leq 0 \]

Objective function:

Maximize $\sum_{k,l,m,n} l_k,l,m,n$
Transformation for $ICOM_E$

Variables:

$\mathcal{I} = \{l_k,l,m,n : 1 \leq k \leq \ell \leq \eta_{G_0} \land 1 \leq m \leq n \leq \eta_{G_1}\}$

$\mathcal{X} = \{x^a_b : 1 \leq a \leq \eta_{G_0} \land 1 \leq b \leq \eta_{G_1} \land G_0[a] = G_1[b]\}$

Constraints:

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

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

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

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

(C.05) $\forall l_k,l,m,n \in \mathcal{I}, \forall m < r < n, \forall \ell < p \leq \eta_{G_0}, \quad G_0[p] = G_1[r], \quad l_k,l,m,n + x^p_r \leq 1$

(C.06) $\forall l_k,l,m,n \in \mathcal{I}, \quad 4l_k,l,m,n - \sum_{m \leq r \leq n} x^k_r - \sum_{m \leq s \leq n} x^\ell_s - \sum_{k \leq p \leq \ell} x^p_m - \sum_{k \leq q \leq \ell} x^q_n \leq 0$

Objective function:

Maximize $\sum_{k,l,m,n} l_k,l,m,n$
Transformation for $ICOM_E$

Variables:

$$\mathcal{I} = \{ l, m, n : 1 \leq k \leq \ell \leq \eta_{G_0} \land 1 \leq m \leq n \leq \eta_{G_1} \}$$

$$\mathcal{X} = \{ x_a^b : 1 \leq a \leq \eta_{G_0} \land 1 \leq b \leq \eta_{G_1} \land G_0[a] = G_1[b] \}$$

Constraints:

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

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

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

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

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

(C.06) $\forall l, m, n \in \mathcal{I}, \quad 4 \sum_{m \leq r \leq n} x_k^r - \sum_{m \leq s \leq n} x_{s}^r - \sum_{k \leq p \leq \ell} x_m^p - \sum_{k \leq q \leq \ell} x_n^q \leq 0$

Objective function:

Maximize $\sum_{k, l, m, n} l, m, n$
Pseudo boolean transformation

Other problems ?

- **other models**: modify constraints C1
- **conserved intervals**: restriction on variables $l_k, l, m, n$
- **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


**BD$X$ and ADJ$X$**
Efficient Tools for Computing the Number of Breakpoints and the Number of Adjacencies between two Genomes with Duplicate Genes

### Experimental results

#### Dataset

Twelve genomes of \(\gamma\)-Proteobacteria [Lerat et al. 2003]

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

average: 3104
Experimental results

Dataset

- Twelve genomes of $\gamma$-Proteobacteria [Lerat et al. 2003]
- 66 possible pairs of genomes

Number of results:

<table>
<thead>
<tr>
<th></th>
<th>Exemplar</th>
<th>maximum matching</th>
<th>intermediate</th>
</tr>
</thead>
<tbody>
<tr>
<td>$ADJ_X$</td>
<td>61/66</td>
<td>66/66</td>
<td>63/66</td>
</tr>
<tr>
<td>$ICOM_X$</td>
<td>21/66</td>
<td>40/66</td>
<td>21/66</td>
</tr>
</tbody>
</table>
Experimental results

Dataset

- Twelve genomes of $\gamma$-Proteobacteria [Lerat et al. 2003]
- 66 possible pairs of genomes

Number of results:

<table>
<thead>
<tr>
<th>model</th>
<th>Exemplar</th>
<th>maximum matching</th>
<th>intermediate</th>
</tr>
</thead>
<tbody>
<tr>
<td>$ADJ_X$</td>
<td>61/66</td>
<td>66/66</td>
<td>63/66</td>
</tr>
<tr>
<td>$ICOM_X$</td>
<td>21/66</td>
<td>40/66</td>
<td>21/66</td>
</tr>
</tbody>
</table>

⇒ Efficient approach for $ADJ_X$
Experimental results

Dataset

- Twelve genomes of \(\gamma\)-Proteobacteria [Lerat et al. 2003]
- 66 possible pairs of genomes

Number of results:

<table>
<thead>
<tr>
<th></th>
<th>model</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Exemplar</td>
<td>maximum matching</td>
<td>intermediate</td>
</tr>
<tr>
<td>(ADJ_X)</td>
<td>61/66</td>
<td>66/66</td>
<td>63/66</td>
</tr>
<tr>
<td>(ICOM_X)</td>
<td>21/66</td>
<td>40/66</td>
<td>21/66</td>
</tr>
</tbody>
</table>

⇒ 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 $ICOM_E$
     - Experimental results
   - Heuristics and hybrid method
     - IILCS$_X$
     - 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\textsubscript{M} heuristic

- Based on IILCS\textsubscript{M} heuristic [Tichy, 82]
- **Idea:** Match genes of a Longest Common Substring (LCS)

1. Compute the Longest Common Substring \( S \)

Example

\[
\begin{align*}
+1 & +2 +3 & +4 & +5 & +6 & +7 \\
+6 & -7 & +4 & +5 & +1 & +6 & -3 & -2 & -1
\end{align*}
\]
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

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

$$+6 \quad - 7 \quad + 4 \quad + 5 \quad + 1 \quad + 6 \quad -3 \quad -2 \quad -1$$
**IILCSₘ heuristic**

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

**IILCSₘ 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 IILCS$_{M}$ 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 + 6 -3 -2 -1
IILCS\(_M\) heuristic

- Based on IILCS\(_M\) 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\textsubscript{M} heuristic

- Based on ILCS\textsubscript{M} heuristic [Tichy, 82]
- **Idea:** Match genes of a Longest Common Substring (LCS)

### IILCS\textsubscript{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

\[
\begin{align*}
  +1 & \quad +2 & \quad +3 & \quad +4 & \quad +5 & \quad +6 & \quad +7 \\
  +6 & \quad -7 & \quad +4 & \quad +5 & \quad +6 & \quad -3 & \quad -2 & \quad -1
\end{align*}
\]
**IILCS_{M} heuristic**

- Based on ILCS_{M} heuristic [Tichy, 82]
- **Idea:** Match genes of a Longest Common Substring (LCS)

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 $S$
2. Match all the genes of $S$ accordingly
3. **Remove genes that cannot be matched**
4. Iterate the process until saturation

#### Example

\[ +1 \quad +2 \quad +3 \quad +4 \quad +5 \quad +6 \quad +7 \]

\[ -7 \quad +4 \quad +5 \quad +6 \quad -3 \quad -2 \quad -1 \]
**IILCS\(_M\) heuristic**

- Based on IILCS\(_M\) 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**

\[
\begin{array}{cccc}
+1 & +2 & +3 & +4 & +5 & +6 & +7 \\
-7 & +4 & +5 & +6 & -3 & -2 & -1 \\
\end{array}
\]
**IILCS}_M \text{ heuristic**

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

<table>
<thead>
<tr>
<th>Step</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Compute the Longest Common Substring $S$</td>
</tr>
<tr>
<td>2</td>
<td>Match all the genes of $S$ accordingly</td>
</tr>
<tr>
<td>3</td>
<td>Remove genes that cannot be matched</td>
</tr>
<tr>
<td>4</td>
<td>Iterate the process until saturation</td>
</tr>
<tr>
<td>5</td>
<td>Compute the measure</td>
</tr>
</tbody>
</table>

**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 $|S| \geq k$
   Then
   - Match all the genes of $S$
   - Remove genes that cannot be matched
   - Return to 1
3. Else Apply the exact method: transformation into a pseudo-boolean linear problem
## Experimental results

### Dataset
- Twelve genomes of \(\gamma\)-Proteobacteria [Lerat et al. 2003]
- 66 possible pairs of genomes

<table>
<thead>
<tr>
<th></th>
<th>Exemplar</th>
<th>maximum matching</th>
<th>intermediate</th>
</tr>
</thead>
<tbody>
<tr>
<td>(ADJ_X)</td>
<td>61/66</td>
<td>66/66</td>
<td>63/66</td>
</tr>
<tr>
<td>(ICOM_X)</td>
<td>21/66</td>
<td>40/66</td>
<td>21/66</td>
</tr>
</tbody>
</table>
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*  
Protocol

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

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.

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.

Places:
- \(G_1, G_2\)

**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

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

**Step 7) Visualization**

Places:
- \(G''_1, G''_2\)
Homologies computation

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

**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 $G'_2$

**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

- 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])
...
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:

- $([3, 5], [3, 5])$
- $([2, 5], [3, 6])$
- ...

Diagram:

- Nodes and edges illustrating the matching and interval computation process.
Seven steps

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

**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.

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

**Step 7) Visualization**
<table>
<thead>
<tr>
<th>Experience view</th>
<th>Genomes view</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>G1:</strong> ECOLI</td>
<td></td>
</tr>
<tr>
<td><strong>G2:</strong> VCHOLERAINC0002505</td>
<td></td>
</tr>
<tr>
<td>measure: common intervals</td>
<td></td>
</tr>
<tr>
<td>method: ILCS heuristic</td>
<td></td>
</tr>
<tr>
<td>model: maximum matching</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Gene view</th>
<th>Interval view</th>
</tr>
</thead>
<tbody>
<tr>
<td>id: 16128042</td>
<td>Index: 46</td>
</tr>
<tr>
<td>genome: ECOLI</td>
<td>conserved</td>
</tr>
<tr>
<td>Pathways</td>
<td>type: reversed_interval</td>
</tr>
<tr>
<td>ECOLI</td>
<td>eco00670</td>
</tr>
<tr>
<td>VCHOLERAINC0002505</td>
<td>eco00790</td>
</tr>
<tr>
<td>[47]:16128042</td>
<td>[434]:15640467</td>
</tr>
</tbody>
</table>
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 $ICOM_X$
- $IILCS_X$ heuristic and Hybrid method
  - Promising results on a real dataset for each problem
Perspectives

Work on MATCH&WATCH

- First experimentation on six chromosomes of $\gamma$-Proteobacteria
- Analyze in details the common intervals obtained
- Add functionalities according to biologists
Perspectives

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

- Multi-chromosomal genome comparison

- Multiple genome comparison

- New algorithms
  - $\alpha$-approximation for $BD_E$ and $BD_I$ when $\text{occ}(G_0) = 1$?
  - $\alpha$-approximation or PTAS for $ICOM_X$ on balanced genomes?
Perspectives

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

- Multi-chromosomal genome comparison

- Multiple genome comparison

- New algorithms
  - $\alpha$-approximation for $BD_E$ and $BD_I$ when $\text{occ}(G_0) = 1$?
  - $\alpha$-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://www.mun.ca/biology/scarr/FISH_chromosomes_300dpi.jpg)
- [http://agaudi.files.wordpress.com/2008/09/dna_overview_es.png](http://agaudi.files.wordpress.com/2008/09/dna_overview_es.png)
- [http://joachimj.club.fr/imagesmada2004bis/PlanchePhylogeniedesprimates.jpg](http://joachimj.club.fr/imagesmada2004bis/PlanchePhylogeniedesprimates.jpg)
- [http://www.g-language.org/g3/](http://www.g-language.org/g3/)
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://www.mun.ca/biology/scarr/FISH_chromosomes_300dpi.jpg)
- [http://agaudi.files.wordpress.com/2008/09/dna_overview_es.png](http://agaudi.files.wordpress.com/2008/09/dna_overview_es.png)
- [http://joachimj.club.fr/imagesmada2004bis/PlanchePhylogeniedesprimates.jpg](http://joachimj.club.fr/imagesmada2004bis/PlanchePhylogeniedesprimates.jpg)
- [http://www.g-language.org/g3/](http://www.g-language.org/g3/)

Thank you

Merci
Appendix

1. Pseudo boolean transformation for other problems
   ILCS_\chi and IIILCS_\chi
   Visualization tool
   Common intervals filtering
   First experimental results
Pseudo boolean transformation for other problems

ILCS$_X$ and II LCS$_X$

Visualization tool

Common intervals filtering

First experimental results
Transformation for $ICOM_E$: objective function

Objective:

$$\text{maximize } \sum_{k,l,m,n} l_{k,l,m,n}$$
Transformation for $ICOM_E$: objective function

Objective:

\[
\text{maximize } \sum_{k,l,m,n} l_{k,l,m,n}
\]

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 $l_{k,l,m,n}$
Transformation for $ICOM_E$: objective function

**Objective:**

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

**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)
  \[ \forall f \in F_{G_0} \cup F_{G_1}, \sum_{1 \leq a \leq \eta_{G_0}} x^a_b = 1 \]
  \[ \sum_{1 \leq b \leq \eta_{G_1}} x^a_b = 1 \]
  \[ G_0[a] = f \quad G_1[b] = f \]

- **C1’**: (Maximal matching model)
  \[ \forall f \in F_{G_0} \cup F_{G_1}, \sum_{1 \leq a \leq \eta_{G_0}} x^a_b = \min\{\text{occ}(f, G_0), \text{occ}(f, G_1)\} \]
  \[ \sum_{1 \leq b \leq \eta_{G_1}} x^a_b \]
  \[ G_0[a] = f \quad G_1[b] = f \]

- **C1”**: (Intermediate matching model)
  \[ \forall f \in F_{G_0} \cup F_{G_1}, \sum_{1 \leq a \leq \eta_{G_0}} x^a_b \geq 1 \]
  \[ \sum_{1 \leq b \leq \eta_{G_1}} x^a_b \]
  \[ G_0[a] = f \quad G_1[b] = f \]
Other models

\[ \forall a = 1, 2, \ldots, \eta_{G_0}, \sum_{1 \leq b \leq \eta_{G_1}} x^a_b \leq 1 \]

\[ G_0[a] = G_1[b] \]

\[ \forall b = 1, 2, \ldots, \eta_{G_1}, \sum_{1 \leq a \leq \eta_{G_0}} x^a_b \leq 1 \]

\[ G_0[a] = G_1[b] \]
Other problems?

Other measures

- $\text{CONS}_X$:
  Generate only variables $l_k, l, m, n$ such that
  
  \[
  ( ( G_0[k] = G_1[m] \land G_0[\ell] = G_1[n] ) \lor \\
  ( G_0[k] = -G_1[n] \land G_0[\ell] = -G_1[m] ) )
  \]
Other problems?

Other measures

- $ICONS_X$: Generate only variables $l_k, l_m, n$ such that
  \[ ( ( G_0[k] = G_1[m] \land G_0[l] = G_1[n] ) \lor \\ ( G_0[k] = -G_1[n] \land G_0[l] = -G_1[m] ) ) \]

- $BD_X$ 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

Idea:
- Match genes of the LCS until saturation
- Compute the Longest Common Substring
- Match all the genes of $S$ accordingly
- Iterate the process until saturation
- Remove all the genes that have not been matched
- Compute the number of common intervals
**ILCS$_M$ heuristic**

**Idea:** Match genes of the LCS until saturation

**LCS:** Longest Common Substring [Tichy, 84]
**ILCS\textsubscript{M} heuristic**

**LCS:** Longest Common Substring [Tichy, 84]

<table>
<thead>
<tr>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
</tr>
</thead>
<tbody>
<tr>
<td>6</td>
<td>7</td>
<td>4</td>
<td>5</td>
<td>1</td>
<td>6</td>
<td>3</td>
</tr>
</tbody>
</table>

**ILCS\textsubscript{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]

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>6</td>
<td>7</td>
<td>4</td>
<td>5</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>3</td>
<td>2</td>
<td>1</td>
<td></td>
</tr>
</tbody>
</table>

#### 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\textsubscript{M} heuristic

**LCS:** Longest Common Substring [Tichy, 84]

\[
\begin{array}{cccccc}
1 & 2 & 3 & 4 & 5 & 6 & 7 \\
6 & 7 & 4 & 5 & 1 & 6 & 3 & 2 & 1
\end{array}
\]

**ILCS\textsubscript{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
ILCSₘ heuristic

### LCS: Longest Common Substring [Tichy, 84]

<table>
<thead>
<tr>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
</tr>
</thead>
<tbody>
<tr>
<td>6</td>
<td>7</td>
<td>4</td>
<td>5</td>
<td>1</td>
<td>6</td>
<td>3</td>
</tr>
</tbody>
</table>

### ILCSₘ 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\textsubscript{M} heuristic

**LCS:** Longest Common Substring [Tichy, 84]

1 2 3 4 5 6 7

6 7 4 5 3 2 1

**ILCS\textsubscript{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

**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]

\[1\ 2\ 3\ 4\ 5\ 6\ 7\ 6\ 7\ 4\ 5\ 3\ 2\ 1\]

⇒ 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**

**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

**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**

**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\textsubscript{M} heuristic**

**IILCS\textsubscript{M} heuristic**

**LCS:** Longest Common Substring [Tichy, 84]

1 2 3 4 5 6 7

6 7 4 5 6 3 2 1

**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]

<table>
<thead>
<tr>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
</tr>
</thead>
<tbody>
<tr>
<td>6</td>
<td>7</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td></td>
<td>3</td>
</tr>
</tbody>
</table>

#### 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\textsubscript{M} heuristic**

**LCS**: Longest Common Substring [Tichy, 84]

```
1 2 3 4 5 6 7
7 4 5 6 3 2 1
```

**IILCS\textsubscript{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**

**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]

1 2 3 4 5 6 7

7 4 5 6 3 2 1

⇒ 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 \( II LCS_X \)
- Visualization tool
- Common intervals filtering
- First experimental results
Appendix

Visualization tool

**Experience view**

<table>
<thead>
<tr>
<th>Experience Window</th>
</tr>
</thead>
<tbody>
<tr>
<td>G1: ECOLI</td>
</tr>
<tr>
<td>G2: VCHOLERA1NC002505</td>
</tr>
<tr>
<td>measure: common intervals</td>
</tr>
<tr>
<td>method: ILCS heuristic</td>
</tr>
<tr>
<td>model: maximum matching</td>
</tr>
</tbody>
</table>

**Gene view**

| id: 16128042 | position: 47 |
| genome: ECOLI |
| Homologies: |
| ECOLI | VCHOLERA1NC002505 |
| [47]:16128042 | [434]:15640467 |
| Pathways: |
| eco00670 |
| eco00790 |

**Interval view**

| G1: ECOLI | G2: VCHOLERA395NC0009457 |
| Index: 46 | conserved | type: reversed_interval |

---

Angibaud Sébastien  
Defence of Phd Thesis  
October 7th 2009
Visualization tool

G1: ECOLI  G2: VCHOLERA395NC009457

Index: 46  conserved  Type: reversed_interval
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?
  \[\Rightarrow\] Three filters to emphasize \textit{the most interesting} common intervals
Common intervals filtering

- Lots of common intervals
- Relevance of common intervals?

⇒ Three filters to emphasize the most interesting common intervals

Filters

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

Common intervals filtering

- Lots of common intervals
- Relevance of common intervals?

⇒ Three filters to emphasize the most interesting 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 interesting* 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 interesting 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
- $\text{ILCS}_X$ and $\text{II LCS}_X$
- Visualization tool
- Common intervals filtering
- First experimental results
## Experimental results

### Input: six chromosomes of \( \gamma \)-Proteobacteria

<table>
<thead>
<tr>
<th>NCBI identifiant</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>NC_000913</td>
<td><em>Escherichia coli</em> K12</td>
</tr>
<tr>
<td>NC_002505</td>
<td><em>Vibrio cholerae</em> 01 biovar eltor str. N16961 chromosome I</td>
</tr>
<tr>
<td>NC_002506</td>
<td><em>Vibrio cholerae</em> 01 biovar eltor str. N16961 chromosome II</td>
</tr>
<tr>
<td>NC_009456</td>
<td><em>Vibrio cholerae</em> 0395 chromosome I</td>
</tr>
<tr>
<td>NC_009457</td>
<td><em>Vibrio cholerae</em> 0395 chromosome II</td>
</tr>
<tr>
<td>NC_006840</td>
<td><em>Vibrio fischeri</em> ES114 chromosome I</td>
</tr>
<tr>
<td>NC_006841</td>
<td><em>Vibrio fischeri</em> ES114 chromosome II</td>
</tr>
</tbody>
</table>
Results: common intervals

<table>
<thead>
<tr>
<th>genome G₂</th>
<th>genome size</th>
<th>computational time</th>
<th>common intervals</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Inparanoid (s)</td>
<td>matching (s)</td>
</tr>
<tr>
<td><strong>NC002505</strong></td>
<td>4243</td>
<td>2742</td>
<td><strong>IILCS</strong></td>
</tr>
<tr>
<td><strong>NC002506</strong></td>
<td>4243</td>
<td>1093</td>
<td><strong>PSB</strong></td>
</tr>
<tr>
<td><strong>NC009456</strong></td>
<td>4243</td>
<td>1133</td>
<td><strong>PSB</strong></td>
</tr>
<tr>
<td><strong>NC009457</strong></td>
<td>4243</td>
<td>2742</td>
<td><strong>IILCS</strong></td>
</tr>
<tr>
<td><strong>NC006840</strong></td>
<td>4243</td>
<td>2586</td>
<td><strong>IILCS</strong></td>
</tr>
<tr>
<td><strong>NC006841</strong></td>
<td>4243</td>
<td>1175</td>
<td><strong>IILCS</strong></td>
</tr>
</tbody>
</table>
Experimental results

- maximal common intervals
- annotated intervals
- relevant intervals