The Minisatellite Transformation Problem: The Run-Length-Encoding Approach and Further Enhancements

Behshad Behzadi & Jean-Marc Steyaert, Ecole Polytechnique
Mohamed Abouelhoda, Cairo University
Robert Giegerich, Bielefeld University
Biology…

- Minisatellites consist of tandem arrays of short repeat units found in genome of most higher eukaryotes.

- High degree of polymorphism at minisatellites has applications from forensic studies to the investigation of the origins of modern human groups.
...Biology...

- These repeats are called variants.

- MVR-PCR is designed to find the variants.

- As an example, MSY1 is the minisatellite on the human Y-chromosomes. There are five different repeats (variants) in MSY1.
Different Repeat Types (Variants) of MSY1

Map Types:

<table>
<thead>
<tr>
<th>Type 1:</th>
<th>C</th>
<th>A</th>
<th>C</th>
<th>T</th>
<th>ATACAT</th>
<th>G</th>
<th>ATGTATATTATA</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type 1a:</td>
<td>C</td>
<td>A</td>
<td>A</td>
<td>C</td>
<td>ATACAT</td>
<td>G</td>
<td>ATGTATATTATA</td>
</tr>
<tr>
<td>Type 2:</td>
<td>C</td>
<td>T</td>
<td>A</td>
<td>A</td>
<td>T</td>
<td>ATACAT</td>
<td>G</td>
</tr>
<tr>
<td>Type 3:</td>
<td>C</td>
<td>A</td>
<td>T</td>
<td>A</td>
<td>T</td>
<td>ATACAT</td>
<td>C</td>
</tr>
<tr>
<td>Type 3a:</td>
<td>C</td>
<td>A</td>
<td>A</td>
<td>C</td>
<td>ATACAT</td>
<td>C</td>
<td>ATGTATATTATA</td>
</tr>
<tr>
<td>Type 4:</td>
<td>C</td>
<td>T</td>
<td>A</td>
<td>A</td>
<td>T</td>
<td>ATACAT</td>
<td>C</td>
</tr>
<tr>
<td>Type 4a:</td>
<td>C</td>
<td>T</td>
<td>A</td>
<td>A</td>
<td>C</td>
<td>ATACAT</td>
<td>C</td>
</tr>
</tbody>
</table>

Distance between types:

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>1a</th>
<th>2</th>
<th>3</th>
<th>3a</th>
<th>4</th>
<th>4a</th>
<th>null</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td>1a</td>
<td>1</td>
<td>0</td>
<td>2</td>
<td>2</td>
<td>1</td>
<td>3</td>
<td>2</td>
<td>4</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>2</td>
<td>0</td>
<td>2</td>
<td>3</td>
<td>1</td>
<td>2</td>
<td>4</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>4</td>
</tr>
<tr>
<td>3a</td>
<td>2</td>
<td>1</td>
<td>3</td>
<td>1</td>
<td>0</td>
<td>2</td>
<td>1</td>
<td>4</td>
</tr>
<tr>
<td>4</td>
<td>2</td>
<td>3</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>0</td>
<td>1</td>
<td>4</td>
</tr>
<tr>
<td>4a</td>
<td>3</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>4</td>
</tr>
<tr>
<td>null</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
</tr>
</tbody>
</table>
Minisatellite Maps: The MSY1 Dataset

DNA Sequence: ... CGGCGAT CGGCGAC CGGCGAC CGGCGAC CGGAGAT ...

Unit types (Alphabet): X = CGGCGAT  Y = CGGCGAC  Z = CGGAGAT

Minisatellite Map: XYYYYZ

• Example Maps from the MSY1 Dataset:

<table>
<thead>
<tr>
<th>Code</th>
<th>Pop.</th>
<th>Hg</th>
<th>MVR map</th>
</tr>
</thead>
<tbody>
<tr>
<td>m1</td>
<td>English</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>m19</td>
<td>English</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>m110</td>
<td>Indian</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>m47</td>
<td>Pygray</td>
<td>6</td>
<td></td>
</tr>
<tr>
<td>m82</td>
<td>San</td>
<td>7</td>
<td></td>
</tr>
<tr>
<td>m121</td>
<td>Finn</td>
<td>16</td>
<td></td>
</tr>
<tr>
<td>m707</td>
<td>Maya</td>
<td>18</td>
<td></td>
</tr>
<tr>
<td>m65</td>
<td>Japanese</td>
<td>20</td>
<td></td>
</tr>
<tr>
<td>m6</td>
<td>English</td>
<td>21</td>
<td></td>
</tr>
<tr>
<td>m125</td>
<td>Berber</td>
<td>21</td>
<td></td>
</tr>
<tr>
<td>m715</td>
<td>Bantu</td>
<td>21</td>
<td></td>
</tr>
</tbody>
</table>

Type 1: ○  Type 2: ●  Type 3: ●  Type 4: ○  Null (or type 0): ⊗ (undetermined variant)
The unequal crossover is a possible mechanism for tandem duplication:
Evolutionary Operations

- Insertion
- Deletion
- Mutation
  - Amplification  \((p\text{-plication})\)
  - Contraction  \((p\text{-contraction})\)
Examples of operations

- Insertion of $d$
  \[ abbc \rightarrow abbd\textit{c} \]

- Deletion of $c$
  \[ abb\textit{cb} \rightarrow abbb \]

- Mutation of $c$ into $d$
  \[ caab \rightarrow daab \]

- 4-plication of $c$
  \[ ab\textit{cb} \rightarrow ab\textit{cccccb} \]

- 2-contraction of $b$
  \[ ab\textit{bc} \rightarrow abc \]
Cost Functions

$I(x)$  
insertion of symbol $x$

$D(x)$  
deletion of symbol $x$

$M(x, y)$  
mutation of symbol $x$ to $y$

$A_p(x)$  
$p$-lication of symbol $x$

$C_p(x)$  
$p$-contraction of symbol $x$
Hypotheses

- All the costs are positive.
- The cost of duplications (and contractions) is less than all other operations.
- Triangle inequality holds:
  \[ M(x,y) + M(y,z) \leq M(x,z) ; M(x,x) = 0 \]
Transformation distance between $s$ and $t$

- Applying a sequence of operations on $s$ transforming it into $t$.
- The cost of a transformation is the sum of costs of its operations.
- $TD = \text{Minimum cost for a possible transformation of } s \text{ into } t$.
- Any transformation which gives this minimum is called an optimal transformation.
Previous Works

- Bérard & Rivals (RECOMB’02)
- Behzadi & Steyaert (CPM’03, JDA’04)
- Behzadi & Steyaert (WABI'04)
Generation vs. Reduction

- The symbols of $s$ which generate a non-empty substring of $t$ are called \textbf{generating symbols}.
- Other symbols of $s$ are \textbf{vanishing symbols}. (These symbols are eliminated during the transformation by a deletion or contraction.)
- The transformation of symbol $x$ into non-empty string $s$ is called \textbf{generation}.
- The transformation of a non-empty string $s$ into a unique symbol $x$ is called \textbf{reduction}. 
The optimal generation of a non-empty string $s$ from a symbol $x$ can be achieved by a non-
di
ti
The schema for an optimal transformation

There exists an optimal transformation of $s$ into $t$ in which all the *contractions* are done *before* all amplifications.
Run-Length Encoding and Run Generation

- The RLE encoding of $aaaabbbcccaabbbccc$ is $a^4b^4c^3a^1b^4c^2$.
- The lengths of the encoded strings with length $n$ and $m$ is denoted by $m'$ and $n'$.
- There exists an optimal generation of a non-empty string $t$ from a single symbol $x$ in which for every run of size $k > 1$ in $t$ the $k-1$ right symbols of the run are generated by duplications of the leftmost symbol of the run.
Preprocessing --> Core algorithm

- Compute the generation cost of all substrings of the target string $t$ from any symbol $x$ of the alphabet: $G(t)[x,i,j]$.
- Compute the optimal generation/reduction costs over the substrings by recurrence using dynamic programming.
- The running time is given by:
  $$O((m'^3 + n'^3)|\text{Alpha}| + mn'^2 + nm'^3 + mn)$$
A different look at Duplication History

Observed

- Right duplication: $s_3$, $s_6$
- Left duplication: $s_3s_4$, $s_6$
- Right duplication: $s_1s_3s_4s_5$, $s_6$
- Left duplication: $s_1s_3s_4s_5$, $s_6$
- Right duplication: $s_1s_2s_3s_4s_5$, $s_6$
- Right duplication: $s_1s_2s_3s_4s_5$, $s_6s_7$
- Right duplication: $s_1s_2s_3s_4s_5$, $s_6s_7s_8$
Alignment of Minisatellite Maps (1)

Complications: comparing maps is more than copy number
1) Types are not identical
2) Types duplicate according to a duplication model
3) Parts of the map may be foreign, appeared by transposition

Example of an alignment:

The two maps $S$ and $R$

Alignment of $S$ and $R$
Alignment of Minisatellite Maps (2)

- Matches refer to common history
- Duplication events refer to individual duplication history
- Insertions/Deletions refer to foreign units
Improved Model of Comparison
Left and Right Simultaneous Dups

Example:

Assume: \( d(a,b)=d(d,c)=d(c,d) < d(a,c)=d(b,d) < d(a,d) \)

Bérard et al., Model

Our NEW Model

There is no rule to allow simultaneous left/right duplications in \( S \) and \( R \)

It has less score. Because there is a rule to allow simultaneous left/right duplications in \( S \) and \( R \)
Algorithm Layout

Observations:

-- Duplications compose intervals in S/R
-- The duplications within an alignment originate either from the leftmost or from the rightmost unit of the interval containing the duplications
-- Optimal alignment must contain optimal duplication history of these intervals

Therefore:

1. Pre-compute and store score of optimal history for all sub-intervals of S and R originated from leftmost/rightmost unit
2. Use Dynamic programming alignment algorithm considering that intervals of S/R appeared as duplications (optimal scores are look-up
Finding an Optimal Duplication History

Duplication history can be represented by an **ordered directed tree ORDT**: Nodes are the units

Edges are directed and weighted by distance between the unit

Each sub-tree can be written as contiguous units \([s_i..s_j]\)

Optimal duplication history:= an optimal ORDT

An optimal ORDT can be found in \(O(n^3)\) time and \(O(n^2)\) space by partitioning contiguous non-overlapping intervals:

\( \text{P1-P5: } O(n^2 \log \log n) \) time
Experimental Running Times

Duplication history:

| | Without RLE | | With RLE |
|---|---|---|---|---|---|---|
| 5 | 147 | 65 | | 5 | 0.46 | 0.46 |
| 10 | 262 | 65 | | 10 | 0.59 | 0.55 |
| 20 | 472 | 61 | | 20 | 0.95 | 0.59 |
| 30 | 703 | 65 | | 30 | 1.11 | 0.56 |
| 50 | 1165 | 65 | | 50 | 1.5 | 0.48 |
| 60 | 1428 | 67 | | 60 | 1.7 | 0.6 |

Alignment algorithm:

<table>
<thead>
<tr>
<th>Data</th>
<th>Algn. No.</th>
<th>MS_ALIGN</th>
<th>MSATcompare</th>
<th>MSATcompareRLE</th>
</tr>
</thead>
<tbody>
<tr>
<td>rand 50</td>
<td>1225</td>
<td>5.58</td>
<td>2.3</td>
<td>0.23</td>
</tr>
<tr>
<td>rand 100</td>
<td>4950</td>
<td>24.2</td>
<td>10.2</td>
<td>0.98</td>
</tr>
<tr>
<td>rand 150</td>
<td>11175</td>
<td>49.8</td>
<td>21.4</td>
<td>2.1</td>
</tr>
<tr>
<td>rand 250</td>
<td>3112</td>
<td>161.5</td>
<td>70</td>
<td>5.9</td>
</tr>
<tr>
<td>rand 350</td>
<td>61075</td>
<td>317</td>
<td>140</td>
<td>12</td>
</tr>
<tr>
<td>MSY1 345</td>
<td>59340</td>
<td>87</td>
<td>25</td>
<td>4.8</td>
</tr>
</tbody>
</table>

- MS_ALIGN is the algorithm of Bérard et al.
- MSATcompare is ours
Detection of Duplication Bias in MSY1 Dataset

E1: run algorithm allowing left- and right- duplications
EL: allow only left duplications
ER: allow only right duplications

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Total Algn.</th>
<th>$r = 1 \times d_H$</th>
<th>$r = 2 \times d_H$</th>
<th>$r = 5 \times d_H$</th>
<th>$r = 10 \times d_H$</th>
<th>$r = \infty$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>L</td>
<td>R</td>
<td>L</td>
<td>R</td>
<td>L</td>
</tr>
<tr>
<td>with nulls</td>
<td>59340</td>
<td>186</td>
<td>0</td>
<td>616</td>
<td>16</td>
<td>3005</td>
</tr>
<tr>
<td>with max. 3 nulls</td>
<td>53956</td>
<td>148</td>
<td>0</td>
<td>398</td>
<td>0</td>
<td>2403</td>
</tr>
<tr>
<td>with no nulls</td>
<td>30876</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>869</td>
</tr>
</tbody>
</table>

L: number of alignments in EL with cost higher than that in E1
R: number of alignments in ER with cost higher than that in E1

There is an important bias: R keeps small while E increases quickly as $r=M/DUP$ increases; this suggests that the units are most often generated from right to left

This raises questions about the underlying duplication mechanisms and the process of evolution.