ARLEM

User Manual and Tutorial

Mohamed I. Abouelhoda [*]

April 2, 2009

1 Introduction

A genomic region is classified as a minisatellite locus if it spans more than 500 bp and is composed of tandemly repeated DNA stretches. Each stretch, called *unit*, is a sequence of nucleotides whose length ranges between 7-100 bp. A minisatellite map represents a minisatellite region, where each unit is encoded by a character and handled as one entity; see Figure 1.

ARLEM is a software tool for performing pairwise alignment of a set of minisatellite maps. The alignment algorithm is the key methodology for comparing maps of different individuals, which has applications in forensic and population studies. In aligning minisatellite maps, one has to consider that regions of the map have arisen as a result of duplication events from the neighboring units; this is in addition to the traditional edit operations (match, mismatch, and insertion/deletions). The single copy duplication model, where only one unit can duplicate at a time, is the most popular, and its biological validation was asserted for the MSY1 minisatellites; see References[4, 6].

The scoring of minisatellite map alignment accounts for common aligned units as well as for individual duplication histories. For example, the score of the alignment in Figure 2 is composed of (1) replacement scores for the unit pairs (s_1, r_1), (s_2, r_2), (s_7, r_3) and (s_8, r_5), (2) costs of duplication of the units s_3 and s_6 originated from the unit s_2, duplication of s_4 from s_5, and duplication of r_4 from r_5, and (3) insertion of the unit s_5. That is, the comparison delivers a three-stages scenario: The aligned units refer to common ancestors, the duplications refer to

[*] Nile University, Center for Informatics Sciences, Giza, Egypt. Email: mabouelhoda@nileuniversity.edu.eg, mabouelhoda@yahoo.com

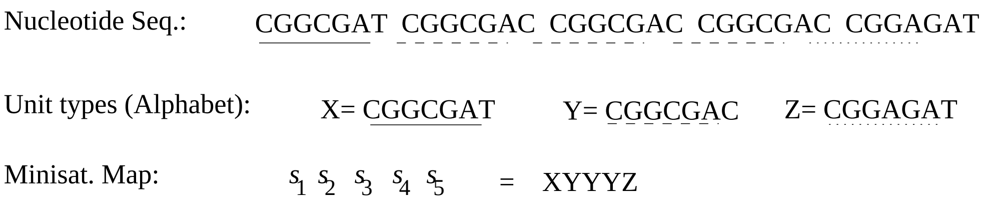

Figure 1: An example of a minisatellite locus: five units, their nucleotide sequences, and the respective map is shown.

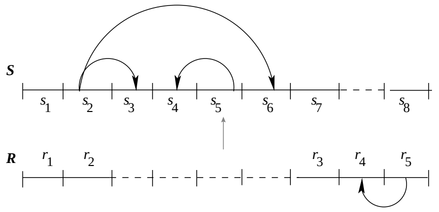

Figure 2: Alignment of two sequences $S = s_1, \ldots, s_8$ and $R = r_1, \ldots, r_5$. The matched copies are put above each other. The arcs represent duplication events.

differences in the individual duplication histories, and the indels refer to units (possibly emerged by a transposition; see Reference[3]) not homologous to the map units.

ARLEM implements an alignment algorithm that improves upon the previous algorithms in many respects: The alignment model is more general and our algorithm relaxes the constraint that the mutation distance $M(a, b)$ between two units is symmetric. The time complexity of our algorithm is alphabet-independent and we show that the run length encoding scheme can be incorporated, which yields an $O(n^2 + nn'^2 + n'^3)$ time and $O(n^2)$ space algorithm. This makes our algorithm the fastest map alignment algorithm in theory, and in practice as well, see [1, 2].

Our algorithm computes an optimal alignment by first computing and storing the cost of an optimal duplication history for each interval in each sequence separately. Then it computes the optimal alignment based on the precomputed costs.

2 ARLEM distribution

ARLEM is free of charge for academic and research proposes. For commercial use, please contact the author (Mohamed Abouelhoda) for license agreement.

3 Installation

ARLEM is written in C++. To build the program for your machine run the commands

```
> make clean
> make
```

Pre-compiled binaries for Linux 64bit versions are distributed with the source code.

4 Running ARLEM

Given an input file , `Mapsfile`, containing a set of maps and a cost file, `CostFile`, specifying the alignment costs (file formats are given in Section 5), the program ARLEM is called as follows:

```
 >arlem -f MapsFile -cfile CostFile  [options]
```

And here is a description of the options.

`-align`
: compute pairwise alignment for all maps.

`-onlyleft`
: Allow only left duplications, where regions in the map can appear by duplication events originated from the unit on the left boundary of this region, and do not allow duplications originated from the unit on the right. For example, duplication of the unit r_4 from r_5 in Figure 2 would not appear in the alignment if this option was set, while the duplication of the units s_3 to s_6 originated from s_2 is allowed.

`-onlyright`
: allow only right duplications, where regions in the map can appear by duplication events originated from the unit on the right boundary of this region, and do not allow duplications originated from the unit on the left. For example, the duplication of the units s_3 to s_6 originated from s_2 would not appear in the alignment if this option was set, while duplication of the unit r_4 from r_5 in Figure 2 is allowed.

`-insert`
: allow free insertions in the history and alignment. If this option was not for the example in Figure 2, then unit s_5 would not have appeared by this operation, rather by duplication from neighboring unit. This option is already set, if the `-rle` option is set.

`-alphadep`
: use alphabet dependent algorithm. This option is not recommended. It is just developed for comparison with old methods.

`-rle`
: use Run Length Encoding Scheme allowing insertions (The option `-insert` is already set). This option is useful if the input maps contain repetitions, which is usually the case with real minisatellite maps.

`-showalign`

display alignment. (This option sets by default ”-align” option) This option reports the operations of an optimal alignment and reports the duplication history of set of maps. The format of this output is in Section 6

`-showstat`

show statistics about input sequences. The output includes their number, and minimum/maximum/averag sequence length.

`-showcost`

show cost file. Redisplay of the input cost file.

`-evaluate a b c`

compare alignment results for 2 datasets, where

a is file containing alignments with left and right duplications (default).

b is file storing alignments without left/right duplications, i.e., with the options `-onlyleft/-only` set.

`-random a b c d e`

generate random map sequences, where

a is the number of map sequences,

b is the minimum sequence length,

c is the maximum sequence length,

d is the alphabet size,

e is the max number of unit repetitions.

5 Program input

5.1 Input map sequences

The format we use is a Fasta like format, where each sequence is represented by two lines: The first is a header line defined by the symbol ¿ and the second contains the sequence itself. Each type can be a word (for example 1a) and the types are separated by white spaces. The following is an example of three maps in Fasta-like format.

```
> ExSeq1
c c b c c c c c d d d d b a a
> ExSeq2
c c d d d d d b b c a c c b b a a
> ExSeq3
 c c d d b b d d b b c b c c b b a a a a
```

5.2 Cost file

The format for editing a cost is as follows: The first lines specify the number of types the symbols representing each type in the input sequence, the indel cost, and the duplication cost, respectively. The pairwise replacement costs are represented by the triangular matrix, where the first row is the cost of replacing the first type into the second, third types, etc., and the second row is the cost of replacing the second type into the third, fourth types, etc. The following is an example of a cost file. Where the cost of replacing type `a` with type `b` and vice versa is 20, while the cost of replacing type `c` with type `d` is 20.

```
# Type no. 4
# Types a b c d
# Indel 40
# Dup 1
# matrix
20 10 20
10 20
10     <--- cost for replacing type c with d
```

6 Program output

ARLEM output is a text file of three parts:

1. Reporting the duplication history of each map originated from the leftmost unit. This is the default calculation of ARLEM, even if no alignment option is specified.

2. Reporting all the score of all pairwise map alignments. The first sequence is numbered zero, the second one, and so on.

3. Reporting, in addition to the alignment score, the alignment oprations. This requires to set the `showalign` option. Figure 3 shows an example of ARLEM textual output. The respective graphical representation of the alignment can also be produced by calling another script see Section 8.1.

7 Example

Assume we have the following two maps as input stored in file called `ExampleMaps`.

```
> ExSeq1
c c b c c c c c d d d d b a a
> ExSeq2
 c c d d b b d d b b c b c c b b a a a a
```

Assume the following cost file, `ExampleCost`, is also given.

```
# Type no. 4
# Types a b c d
# Indel align 40
# Indel hist 40
# Dup 1
# matrix
20 10 20
10 20
10
```

To produce alignment based on the run length encoding scheme and report alignment operations, we call ARLEM as follows:

```
>arlem -f ExampleMaps -cfile ExampleCost  -rle -align -showalign
```

8 Post-processing

8.1 Generating graphical representation

We provide a Perl script (`pp_map_result.pl`) that reads the ARLEM alignment output and generates LaTex-based images as displayed in Figure 4. This script is in the directory `postprocessing/pl` distributed with ARLEM. The script runs as follows.

```
perl pp_map_result.pl ArlemOutput InputSequences PrefixOut
```

where ArlemOutput is the output file of ARLEM (obtained by directing the standard output into a file using the "¿" directive.), and PrefixOut is a prefix to the report files. This script works correctly provided that Latex is well installed on your system. The output is .dvi, .ps files.

8.2 Computing phylogeny

We provide a Perl script (`pp_w.matrix.pl`) that reads the ARLEM alignment output and computes a phylogenitc tree. The tree is reported in Newick format and plotted using njplot into a PDF file. The script is called as follows.

```
perl  pw_matrix.pl MSATcompareOutfile SequenceFile
```

To run this script correctlty, you should be sure that the programs `bionj` [5] and `njplot` whose executables for Linux 64bit version are distributed with this distribution are running on your machines. For installing these programs and compiling them on your machine, see the following subsection.

This script produces three output files:

1. ".mat" file containing the pairwise distances in a mtrix fromat suitable for the program `bionj`.
2. ".nwk" file containing the resulting phylogentic tree computed with the program `bionj`.
3. ".pdf" file containing a plot of the tree produced by the program `njplot`.

The destination of these three files will be displayed, which is the directory containing ARLEM output file.

8.3 Installing `bionj` and `njplot`

`bionj` [5] is a program whose code exists in a single ".c" file. You can complile it using the traditional "gcc" compiler.

Here we use `njplot` to produce PDF output displaying the tree. To compile the program for this task do the following two steps

- Download the PDFlib-Lite library, and install it on your machine.
- compile `njplot` as follows:

  ```
  g++ -DNO_GUI -Wimplicit-function-declaration -g njplot-vib.c
  LIBPDF_PATH/libpdf.a -o njplot.x
  ```

  where `LIBPDF_PATH` is the path containing the PDFlib library. The executable here is called `njplot.x`. Note that this version of `njplot` has no graphical user interface.

9 Acknowledgment

This work was funded by Center of Informatics Sciences, Nile University, and ITIDA, Egypt.

References

[1] M.I. Abouelhoda, R. Giegerich, B. Behzadi, and J. M. Steyaert. Alignment of minisatellite maps: A minimum spanning tree-based approach. In *Proc. of the 6th APBC*, pages 261–272, Kyoto, Japan, 2008.

[2] M.I. Abouelhoda, R. Giegerich, B. Behzadi, and J. M. Steyaert. Alignment of minisatellite maps based on run length encoding scheme. *Journal of Bioinformatics and Computational Biology*, In press.

[3] C. Alkan, J.A. Bailey, E.E. Eichler, and et al. An algorithmic analysis of the role of unequal crossover in alpha-satellite DNA evolution. *Genome Informatics*, 13:93–102, 2002.

[4] S. Bérard and E. Rivals. Comparison of minisatellites. *Computational Biology*, 10((3-4)):357–372, 2003.

[5] O. Gascuel. BIONJ: an improved version of the nj algorithm based on a simple model of sequence data. *Molecular Biology and Evolution*, 14:685695, 1997.

[6] M.A. Jobling, N. Bouzekri, and P.G. Taylor. Hypervariable digital DNA codes for human paternal lineages: MVR-PCR at the Y-specific minisatellite, MSY1 (DYF155S1). *Human Molecular Genetics*, 7(4):643–653, 1998.

```
Aligning Seq1: 0  Seq2: 1

#----------- The operations ------------
Match             : (15, 20) score: 57
Left dup. in R    : [17..19] score: 57
#                   17 -> 18, d(17,18)=0
#                   18 -> 19, d(18,19)=0
Match             : (14, 17) score: 55
Left dup. in R    : [9..16] score: 55
#                   9 -> 10, d(9,10)=0
#                   10 -> 12, d(10,12)=0
#                   10 -> 11, d(10,11)=10
#                   12 -> 15, d(12,15)=0
#                   12 -> 13, d(12,13)=10
#                   13 -> 14, d(13,14)=0
#                   15 -> 16, d(15,16)=0
Match             : (13, 9) score: 28
Match             : (12, 8) score: 28
Left dup. in S    : [9..11] score: 28
#                   9 -> 10, d(9,10)=0
#                   10 -> 11, d(10,11)=0
Match             : (9, 7) score: 26
Left dup. in S    : [4..8] score: 26
#                   4 -> 5, d(4,5)=0
#                   5 -> 6, d(5,6)=0
#                   6 -> 7, d(6,7)=0
#                   7 -> 8, d(7,8)=0
Match             : (4, 6) score: 22
Match             : (3, 5) score: 12
Left dup. in R    : [2..4] score: 12
#                   2 -> 3, d(2,3)=10
#                   3 -> 4, d(3,4)=0
Match             : (2, 2) score: 0
Match             : (1, 1) score: 0
Match             : (0, 0) score: 0

Note the unit at position 0 is the artificial unit "$"
Score of aligning Seq:0, Seq:1 =57
```

Figure 3: Textual output of the program ARLEM for the two maps given above in the example file ExampleMaps. Here, the alignment operations are also reported. The operations are listed from right to left w.r.t. the given maps. The line “Left dup. in R : [2..4]” means that the uits R[2..4] appeared as a result of duplication events originated from the unit R[2]. The score of aligning the two sequences is also reported.

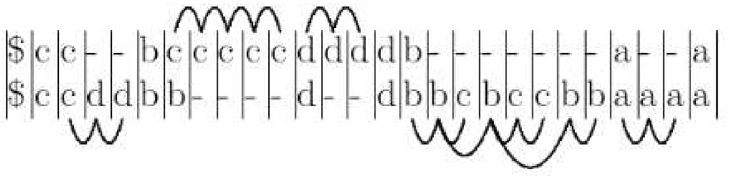
$ c c - - b c c c c c d d d d b - - - - - - - a - - a
$ c c d d b b - - - - d - - d b b c b c c b b a a a a

Figure 4: Graphical representation of the alignment of the two maps in Figure 3. Archs correspond to duplication events. A region of the map asscoiated with duplication events is the one such that each unit in it is sscoiated with at least one arch. To correctly read the duplication history, start from either the leftmost unit if it matches a unit in the other sequence and follow the archs. Otherwise start from the rightmost unit. For example, the interval containing the units cdd in the lower sequence is associated with duplication events where the unit c produced the unit d on its right, which in turn produced the other d right to it.