

Short template switch events explain mutation clusters in the human genome

Ari Löytynoja¹ and Nick Goldman²

¹*Institute of Biotechnology, University of Helsinki, Helsinki, Finland;*

²*European Molecular Biology Laboratory, European Bioinformatics Institute (EMBL-EBI), Wellcome Genome Campus, Hinxton, UK*

Supplemental Methods

FPA – the four-point aligner

The algorithm described in Supplemental Algorithm S1 is implemented in the program FPA.

Compilation

```
$ git clone https://github.com/ariloytynoja/fpa.git
$ cd fpa/
$ make
```

Usage

FPA has two use modes, scan and visualisation. For filtering, awk is used.

First, mutation clusters are identified and the best explanation involving a template switch is computed. The results are output as a table that is redirected to a file.

```
$ ./fpa --scan --pair homo_sapiens.12.74743744.74973891.fas > homo_sapiens.12.74743744.74973891.csv
```

Second, candidate events are filtered using awk.

```
$ awk -F, '($8-$9>12 && $11>=0.95 && $13>=0.95 && $17==0 && $20>5) {print $0}' \
homo_sapiens.12.74743744.74973891.csv > one_hit.csv
```

More complex criteria can be used for filtering. The fields used here are:

```
8: sp2_ref (switch point 2)
9: sp3_ref (switch point 3)
11: iden_up (identity upstream)
13: iden_down (identity downstream)
17: masked (sequence masking: 0=none)
20: sum_mis (excess mismatches in forward alignments)
```

Third, interesting cases are visualised.

```
$ ./fpa --pair homo_sapiens.12.74743744.74973891.fas --print-file one_hit.csv
```

```
chr12:74744825-74744838
```

Switch process:

```
F1: L AATTACACAATTGTGTGATAATATGTGCTGGCCTGTGCC 1
```

```
F3:
```

```
4 TGTTCATGTCATATCATTACCCTTAACTATGTAATCT R
```

```
RF: AATTACACAATTGTGTGATAATATGTGCTGGCCTGTGCCCATTTAGCTACCTTCATGTCATGTCATATCATTACCCTTAACTATGTAATCT
```

```
RR: TTAATGTGTTAACTATTATAACACGACCGGAACAAGGGGTAAATCGATGGAAGTACAACGTACAGTATAGTAATGGGAATTGATACATTAGA
```

```
F2: 3 ACAAGGGGTAAATC 2
```

Template-switch alignment:

```
TATTGTGCTGGCCTGTGCC|CTAAATGGGGAACA|TGTTCATGTCATATCATT
TATTGTGCTGGCCTGTGCC|CTAAATGGGGAACA|TGTTCATGTCATATCATT
```

EPO alignment:

```
TATTGTGCTGGCCTGTGCCcAaaTgGggAa- -CATGTTGCATGTCATATCATT
TATTGTGCTGGCCTGTGCCCATTTAGCTACCTTCATGTTGCATGTCATATCATT
```

Reconstruction and analysis of *de novo* mutations

Besenbacher et al. (2016) provide the MNM clusters (286 in total) in a binary spreadsheet format. We converted the data to comma-separated format, a subset of which looks like this:

```
chrom,pos,# mutations , mutations , , , , , , range
chr3,61130501,3,chr3:61130501_G_GTC,chr3:61130502_G_T,chr3:61130503_A_C , , , , , 2
chr11,49726755,3,chr11:49726755_A_T,chr11:49726756_A_T,chr11:49726758_T_A , , , , , 3
chr4,38561590,3,chr4:38561590_A_T,chr4:38561591_C_T,chr4:38561593_CT_C , , , , , 3
chr4,127759756,3,chr4:127759756_G_T,chr4:127759757_C_G,chr4:127759759_A_T , , , , , 3
...
```

We wrote a custom Perl script that parses this format, places the mutations on the reference sequence and then finds the best explanation involving a template switch event:

```
#!/usr/bin/perl
use strict;
my $ref = "human_b36_male.fa";
my $ufl = 100;
my $dfl = 100;

open IN,$ARGV[0];
while(<IN>){
    chomp;
    next if (/chrom/);
    my @r=split /\./,$_;
    my $l = pop @r;
    next if ($l>100);
    my @g;
    foreach my $e(@r) {
        if ($e =~ /chr.+:/){
            push @g,$e;
        }
    }
    my ($c1,$p1) = ($g[0] =~ /chr(.+):(\d+)-/);
    my $cmd = "samtools faidx $ref $c1:."($p1-$ufl)."-."($p1+$dfl)."| grep -v '>'|tr -d '\n'";
    my $os = '$cmd';
    my $cs = $os;
    my $offset=0;
    foreach my $e(@g) {
        my ($c,$p,$ra,$va) = ($e =~ /chr(.+):(\d+)-(.+)-(.+)/);
        my $pos = $p-$p1+$ufl+$offset;
        substr($os,$pos,length($ra))=$va;
        $offset+=length($va)-length($ra);
    }

    print "\n","#"x50,"\n\n$c1:$p1\n\n";
    my $name = $c1."-".$p1;
    open(TMP,">tmp.fas");
    print TMP ">mut\n$os\n>ref\n$cs\n";
    open(REF,">$name.ref");
    print REF ">ref\n$cs\n";
    open(QRY,">$name.qry");
    print QRY ">mut\n$os\n";

    system "mafft tmp.fas > $name.pair 2> /dev/null";
    system "fpa --pair $name.pair --scan --verbose --long --align Mafft";
}
```

We applied this script to clusters that were less than 100 bp in size. Assuming that the data and script file are called `data.csv` and `script.pl`, we can run the analysis with command:

```
$ perl script.pl data.csv
```

The script works on Linux systems equipped with the necessary sequence analysis tools (samtools, mafft).