

## Mauve Output File Formats

When a genome alignment is created, Mauve creates several output files containing data related to the alignment. Two of these files, the .mauve and .alignment files actually contain the alignment in two different formats. The other files contain auxiliary information such as the genome phylogenetic guide tree that was used for alignment, an identity matrix for the genomes, the location of *backbone* -- regions conserved among all genomes, and the locations of *islands* -- regions where one or a subset of the genomes has a unique sequence element.

The following sections describe the information contained by each of these files and their associated file formats.

### The .alignment file and the XMFA file format

The .alignment file contains the complete genome alignment generated by Mauve in the eXtended Multi-FastA (XMFA) file format. This standard file format is also used by other genome alignment systems that align sequences with rearrangements. The XMFA file format supports the storage of several collinear sub-alignments, each separated with an = sign, that constitute a single genome alignment. Each sub-alignment consists of one FastA format sequence entry per genome where the entry's defline gives the strand (orientation) and location in the genome of the sequence in the alignment.

The general structure of the file format as described by its author ([Michael Brudno](#)) is as follows:

```
>seq_num:start1-end1 ± comments (sequence name, etc.)
AC-TG-NAC--TG
AC-TG-NACTGTG
...

> seq_num:startN-endN ± comments (sequence name, etc.)
AC-TG-NAC--TG
AC-TG-NACTGTG
...
= comments, and optional field-value pairs, i.e. score=12345

> seq_num:start1-end1 ± comments (sequence name, etc.)
AC-TG-NAC--TG
AC-TG-NACTGTG
...

> seq_num:startN-endN ± comments (sequence name, etc.)
AC-TG-NAC--TG
AC-TG-NACTGTG
...
= comments, and optional field-value pairs, i.e. score=12345
```

### Non-standard XMFA formatting used by the Mauve GUI

The Java-based Mauve alignment viewer requires some non-standard formatting of the XMFA file in order to display it. Most importantly, the Mauve viewer requires that every nucleotide of the input genome sequences be recorded exactly once in the XMFA file. Thus, nucleotides in one genome can not to multiple sites in a second genome. For segments of a genome that did not align and were outside LCBs, these segments must be given as ungapped singleton entries in the XMFA:

```
> seq_num:startN-endN ± comments
ACTCAGGTTATCG...
=
```

A second non-standard formatting requirement is that the first LCB entry in the XMFA list all input genome sequences, even if they did not align in that LCB. Genomes that have no sequence in that LCB are given using 0-0 as the coordinate range. For example, in an alignment of four genomes where only two were aligned in the first LCB, the initial LCB might look like:

```
> 1:0-0 +
```

```
>2:1-377 +
ACGA---TAAAATTCCC...

>3:1-422 -
ACTACCCTACAATTGGC...

>4:0-0 +
=
```

## The Mauve alignment file format

The .mauve or .mln file also contains a representation of the genome alignment. Instead of including every aligned nucleotide as in the XMFA format, the Mauve alignment format stores the coordinates of large exactly matching regions to save space. For similar genomes the Mauve alignment format saves a significant amount of disk space over the XMFA format.

The mauve alignment format begins with a single line stating the revision of the file format, followed by several lines describing the sequences that were aligned. Using the alignment of three *Salmonella* genomes as an example:

```
FormatVersion      4
SequenceCount      3
Sequence0File      D:S_typhi.fas
Sequence0Length    4809037
Sequence1File      D:S_typhi2.fas
Sequence1Length    4791961
Sequence2File      D:S_typhimurium.fas
Sequence2Length    4857432
IntervalCount      69
```

Currently Mauve uses version 4 of its alignment file format. The next line contains the token SequenceCount which is used to specify the number of sequences aligned. Two lines are then given for each sequence, the first specifying the location of the original sequence file and the second specifying the sequence length in nucleotides. The final line contains the token IntervalCount which specifies the number of locally collinear blocks that were found among the aligned genomes.

The remainder of the file contains a number of Interval definitions, each of which specifies an LCB-- one collinear region of aligned sequence. Together, these LCBs make up a complete genome alignment with rearrangements. Here is an example interval definition:

```
Interval 3
153 292618 -2687311      294793
GappedAlignment
7 292771 -2687304      294946
GCCTGCG
GCCTGCG
CCATGTC
53 292778 -2687251      294953
GappedAlignment
1 292831 -2687250      295006
A
A
G
127 292832 -2687123      295007
```

Each LCB begins with an Interval token specifying the relative position of the LCB within the first genome aligned (the reference genome). Subsequent lines specify the actual alignment. When constructing an alignment, Mauve chooses a set of multi-MUMs (exactly matching regions present in each genome aligned) to anchor its alignment with. Each Interval definition records the position of these multi-MUM anchors in addition to alignments of the regions between anchors that were calculated using Clustal-W. In the example above, the line 153 292618 -2687311 294793 records a multi-MUM of length 153, with a left-end at 292,618 in *S\_typhi*, on the opposite strand at position 2,687,311 in *S\_typhi2*, and at 294,793 in *S\_typhimurium*.

The next 5 lines in the example above give an alignment of inexact matching sequence generated by ClustalW. The token ClustalResult indicates that the following lines belong to such an alignment. The next line gives the total length of the (possibly

gapped) alignment and the left-end of the Clustal alignment in each genome. Finally, the next three lines (one per sequence aligned) record the actual alignment.

Each Interval records one or more of the multi-MUM and Clustal alignment entries which, when strung together, can specify a complete alignment over the region spanned by the LCB.

## The islands file

The .islands file contains a tab-delimited text listing of genomic islands found in the alignment. Each island represents a region of the alignment where one or more genomes have a sequence element that one or more others lack. In the current Mauve implementation, an island is defined by the genome coordinates of one sequence where another genome contains a gap of length  $n$  or longer in that part of the alignment. The length of gaps that constitute islands can be set with the Minimum Island Size field of the Align sequences dialog box. For example, if  $n$  was defined as 5 for the following alignment:

```
Genome 0: ACACGTTTCGCTTCGAAA
Genome 1: ACAC-----TTCGAA-
Genome 2: ATACGATCGCTTCGTAA
```

We would say that genomes 0 and 2 have an island at positions 5 through 10. Each line of the .islands file records a single island in the form: GenomeA # <tab> leftA <tab> rightA <tab> GenomeB # <tab> leftB <tab> rightB. So in the .islands file, our example islands would be recorded as:

```
0 4 11 1 4 5
1 4 5 2 4 11
```

The first line records that in Genome #0 nucleotides 4 through 11 align with nucleotides 4 through 5 in Genome #1. Similarly, the second line records that nucleotides 4 and 5 of Genome #1 align with nucleotides 4 through 11 of Genome #2. In both cases the island length is 6 and can be calculated as  $\text{absolute}(\text{rightA} - \text{leftA}) - (\text{rightB} - \text{leftB})$ . Note that negative left and right values indicate the inverse orientation (the opposite strand).

## The original Mauve backbone file

The .backbone file records regions of the alignment where sequence is conserved among *all* of the genomes being aligned. The current Mauve implementation defines a conserved region as an area of the alignment at least  $x$  nucleotides long that contains no gaps as long or longer than  $y$  nucleotides. When using the Align sequences window to perform an alignment, the values of  $x$  and  $y$  are fixed to the minimum island size. These values can be set explicitly using the command-line mauveAligner application.

Each line of the .backbone file records a single conserved segment. Left and right end coordinates of the conserved segment are given for each genome sequence. For example, the line:

```
22256      22371    20147    20299    22255    22370
```

Would indicate that nucleotides 22,256 through 22,371 from the first genome are conserved in the second and third genomes from 20,145 through 20,299, and 22,255 through 22,370 respectively. Two entries exist per genome, and each entry is tab-delimited. Negative valued coordinates indicate an inverted region (on the opposite strand).

## The Progressive Mauve backbone file

Progressive Mauve utilizes a revised backbone file format which reflects its ability to align regions conserved among subsets of the genomes under study. A short example of the backbone file format is:

```
seq_0_leftend  seq_0_rightend  seq_1_leftend  seq_1_rightend  seq_2_leftend  seq_2_rightend
1              15378           1              15377           1              15377
16728         19795           15378          18446           15378          18445
0              0               18447          18668           18446          18667
19796         20566           18669          19439           18668          19438
```

The first line is a header line, indicating the information contained by each column. Each subsequent line corresponds to a segment of DNA conserved among two or more genomes. Thus, the first line indicates that the segment between coordinates 1 and 15378 in the first genome is homologous to the segment between coordinates 1 and 15377 of the second and third genomes. Similarly, the second line indicates that the segment [16728-19795] in the first genome is homologous to [15378-18446] in the second genome

and [15378-18445] in the third genome.

An island exists in the first genome between [15379-16727], and its existence is given in the backbone file as a lack of any line containing that segment. The seq\_0\_rightend column skips from 15378 on line 1 to 16728 on line two. By default, the rows of the backbone file are sorted on the seq\_0\_leftend column (absolute value). To infer islands in seq 0, we can thus simply compare the rightend of one line to the leftend from the subsequent line. The data can be trivially processed to observe islands in other genomes using a spreadsheet program like OpenOffice Calc or MS Excel. Simply sort the rows on the absolute value of the column for a sequence (e.g. seq\_2\_leftend) and then compare right-end to left-end on the subsequent line.

An island (or subset backbone) also exists in the second and third genomes. The existence of the subset backbone is given on the third line, where seq\_0\_leftend and seq\_0\_rightend both have zero values to indicate that the first genome lacks any detectable homology to the segments [18447-18668] in the second genome and [18446-18667] in the third genome.

## The guide tree file

The guide tree is the standard Newick tree file format. A decent description of the Newick tree file format can be read here: <http://evolution.genetics.washington.edu/phylip/newicktree.html>

## The identity matrix file

This is tab-delimited text where rows and columns are genomes in the order input to the aligner. Identity scores range between 0 and 1, where 0 indicates that no identical homologous nucleotides were found, and 1 indicates that every homologous nucleotide was identical.

## The permutation matrix file

This is a tab-delimited text file that records the order and orientation that each LCB occurs in the aligned genomes. The permutation matrix file can be used to infer phylogenetic rearrangement history using tools such as BADGER, GRAPPA, MGR, and others. This file is generated by adding the command-line option --permutation-matrix-output=<filename> when running mauveAligner. An example of a file with three genomes and seven LCBs follows:

```
0  1  2  3  4  5  6
1  2  3 -6 -5 -4  0
0  1  2  3 -6 -5 -4
```

Each genome is recorded on a single line, with lines ordered according to the order of input genomes. The LCB arrangements are recorded on each line, with the first genome used as a reference genome to assign numeric identifiers to LCBs. A minus sign (-) indicates that a block is inverted relative to the reference genome.

## The LCB boundary file

This is another tab-delimited file that complements the permutation matrix file with information about the LCB boundaries. This file can be used, for example, to derive the lengths of blocks and by extension, the lengths of genome rearrangements predicted by a rearrangement history reconstruction algorithm. An example of the file format for three genomes follows:

```
#seq0_leftend  seq0_rightend  seq1_leftend  seq1_rightend  seq2_leftend  seq2_rightend
1              11936          250           14147          1             11951
13856         54361         16067         56756         -3398531      -3439222
57092         120151        3854873       3917932       -41024        -106027
...
```

The file starts with a header line defining the content of each column. Each subsequent row defines the left- and right-side LCB boundaries for one LCB. Thus, the first row indicates that nucleotides 1-11936 from genome 0 correspond to nucleotides 250-14147 from genome 1 and nucleotides 1-11951 from genome 2.

## The SNP file

This tab-delimited file can be created from alignments using Mauve version 2.3.0 and later. For every polymorphic site in an alignment, the SNP file records the nucleotides present in each genome at that site, along with the sequence coordinates of the site

in each genome. An example on three genomes is as follows:

SNP pattern	sequence_1	sequence_2	sequence_3
AAT	5276590 5246627 394		
TTC	5276784 5246821 588		
AAC	5277418 5247455 1222		
MAA	5278225 5248262 2030		
AAC	5282804 5252841 6609		

The first column lists the SNP pattern with sequences ordered the same as when input for alignment. The top of the XMFA file can be used as a reference for sequence filenames. Each subsequent column contains coordinates for each of the genomes in turn. Thus each line lists a SNP pattern and the location of that SNP in each genome. Note that IUPAC nucleotide ambiguity codes are considered as possible SNPs -- hence the M in the fourth example SNP above.

## The orthologs file

The orthologs file is a tab-delimited file that can be created from progressiveMauve alignments using Mauve version 2.3.0 and later. The ortholog file lists groups of annotated and unannotated genes that are predicted to be *positionally* orthologous by whole-genome multiple alignment. Each row in the file lists a group of orthologous genes, along with the index of the genome from which the gene derives, the name of the gene (if given in the annotation file) and its sequence coordinates in the global coordinate system of that genome. Entries within a line are tab-delimited and colon-delimited. An example for 4 genomes follows:

```
0:Z03:2818-3750 1:c04:3512-4444 2::2801-3733 3:ECSE_03:2800-3732
0:Z04:3751-5037 1:c05:4445-5731 2::3734-5020 3:ECSE_04:3733-5019
0:Z05:5251-5547 1:c07:5945-6241 1:c08:6021-6269 2::5234-5530 3:ECSE_05:5233-5529
0:Z06:5700-6476 1:c10:6301-7077 3:ECSE_06:5682-6458
```

The first line lists a group of four orthologous genes, with one gene coming from each genome. In the first entry 0:Z03:2818-3750, the leading 0 refers to the genome's index, with indices assigned in the order the genomes were input for alignment. Thus genome 0 is the first genome, 1 is the second, and so on. The next part, Z03, refers to the locus\_tag identifier for the annotated gene. The third colon-delimited part refers to the coordinate range of the annotated gene. The remainder of the line lists out genes in the other three genomes found to be positionally orthologous. In the case of genome 2 we have an entry 2::2801-3733. In this case there was no gene annotated in the region, but a region was found to be positionally orthologous, and so the coordinates of that region are listed without a locus\_tag.

The third line in the example highlights a situation where multiple annotated genes in one genome are found to be orthologous to a single gene in other genomes. In this case, two overlapping genes were annotated in genome 1, and each of those genes individually was predicted to be positionally orthologous to the corresponding genes in other genomes. Since they overlap and are orthologous to the same genes in other genomes, they are considered a group of positional orthologs. Thus, any group of positional orthologs may contain multiple genes from a single genome.

The fourth line in the example illustrates the situation where one of the genomes does not have a positional ortholog of the genes. In this case, genome 2 lacks a region predicted to be positionally orthologous.