

Using progressiveMauve from the command-line

progressiveMauve also offers many command-line options to provide detailed control over alignment parameters. This section gives a reference of the parameters and some examples regarding their usage.

Command-line paramter reference

`--apply-backbone=<file>` Read an existing sequence alignment in XMFA format and apply backbone statistics to it

`--disable-backbone` Disable backbone detection

`--mums` Find MUMs only, do not attempt to determine locally collinear blocks (LCBs)

`--seed-weight=<number>` Use the specified seed weight for calculating initial anchors

`--output=<file>` Output file name. Prints to screen by default

`--backbone-output=<file>` Backbone output file name (optional).

`--match-input=<file>` Use specified match file instead of searching for matches

`--input-id-matrix=<file>` An identity matrix describing similarity among all pairs of input sequences/alignments

`--max-gapped-aligner-length=<number>` Maximum number of base pairs to attempt aligning with the gapped aligner

`--input-guide-tree=<file>` A phylogenetic guide tree in NEWICK format that describes the order in which sequences will be aligned

`--output-guide-tree=<file>` Write out the guide tree used for alignment to a file

`--version` Display software version information

`--debug` Run in debug mode (perform internal consistency checks--very slow)

`--scratch-path-1=<path>` Designate a path that can be used for temporary data storage. Two or more paths should be specified.

`--scratch-path-2=<path>` Designate a path that can be used for temporary data storage. Two or more paths should be specified.

`--collinear` Assume that input sequences are collinear--they have no rearrangements

`--scoring-scheme=<ancestral | sp_ancestral | sp>` Selects the anchoring score function. Default is extant sum-of-pairs (sp).

`--no-weight-scaling` Don't scale LCB weights by conservation distance and breakpoint distance

`--max-breakpoint-distance-scale=<number [0,1]>` Set the maximum weight scaling by breakpoint distance. Defaults to 0.9

`--conservation-distance-scale=<number [0,1]>` Scale conservation distances by this amount. Defaults to 1

`--skip-refinement` Do not perform iterative refinement

`--skip-gapped-alignment` Do not perform gapped alignment

`--bp-dist-estimate-min-score=<number>` Minimum LCB score for estimating pairwise breakpoint distance

`--mem-clean` Set this to true when debugging memory allocations

```
--gap-open=<number> Gap open penalty
--gap-extend=<number> Gap extend penalty
--substitution-matrix=<file> Nucleotide substitution matrix in NCBI format
--weight=<number> Minimum pairwise LCB score
--min-scaled-penalty=<number> Minimum breakpoint penalty after scaling the penalty by expected divergence
--hmm-p-go-homologous=<number> Probability of transitioning from the unrelated to the homologous state [0.0001]
--hmm-p-go-unrelated=<number> Probability of transitioning from the homologous to the unrelated state [0.000001]
--seed-family Use a family of spaced seeds to improve sensitivity
```

Examples of progressiveMauve usage

Example 1. Align three genomes from the input files `genome_1.gbk`, `genome_2.gbk`, and `genome_3.gbk`, saving the output to a file called `threeway.xmfa`

```
progressiveMauve --output=threeway.xmfa genome_1.gbk genome_2.gbk genome_3.gbk
```

Example 2. Align the same three genomes but also save the guide tree and produce a backbone file

```
progressiveMauve --output=threeway.xmfa --output-guide-tree=threeway.tree
--backbone-output=threeway.backbone genome_1.gbk genome_2.gbk genome_3.gbk
```

Example 3. Align the same three genomes, but do not detect forced alignment of unrelated sequence and do not create a backbone file

```
progressiveMauve --output=threeway_no_backbone.xmfa --disable-backbone genome_1.gbk
genome_2.gbk genome_3.gbk
```

Example 4. Detect forced alignment of unrelated sequence in the alignment produced in Example 3. Use custom Homology HMM transition parameters. Save a backbone file.

```
progressiveMauve --apply-backbone=threeway_no_backbone.xmfa --output=threeway.xmfa
--backbone-output=threeway.backbone --hmm-p-go-homologous=0.001
--hmm-p-go-unrelated=0.000005
```

Example 5. Compute ungapped local-multiple alignments among the input sequences and save them to a file called `threeway.mums`

```
progressiveMauve --mums --output=threeway.mums genome_1.gbk genome_2.gbk genome_3.gbk
```

Example 6. Compute an alignment of the same three genomes, using previously computed local-multiple alignments

```
progressiveMauve --match-input=threeway.mums --output=threeway.xmfa genome_1.gbk
genome_2.gbk genome_3.gbk
```

Example 7. Set a custom breakpoint penalty to cope with genomes where default penalty does not work. The default penalty can be extracted from the program's textual output, in this hypothetical example, the default penalty will be 100000.

```
progressiveMauve --output=threeway.xmfa --weight=50000 genome_1.gbk genome_2.gbk
genome_3.gbk
```

Example 8. Set a minimum scaled breakpoint penalty to cope with the case where most genomes are aligned correctly, but manual inspection reveals that a divergent genome has too many predicted rearrangements.

```
progressiveMauve --min-scaled-penalty=5000 --output=threeway.xmfa genome_1.gbk
```

```
genome_2.gbk genome_3.gbk
```

Example 9. Globally align a set of collinear virus genomes that reside in a single FastA file. Use seed families to improve anchoring sensitivity in regions below 70% sequence identity.

```
progressiveMauve --collinear --seed-family --disable-backbone --output=virus.xmfa  
all_virii.fasta
```