MAFFT

Section: Mafft Manual (1)
Updated: 2007-06-09

Contents

NAME

mafft - Multiple alignment program for amino acid or nucleotide sequences

SYNOPSIS

mafft [**options**]	*input*		[> *output*]
linsi	*input*		[> *output*]
ginsi	*input*		[> *output*]
einsi	*input*		[> *output*]
fftnsi	*input*		[> *output*]
fftns	*input*		[> *output*]
nwns	*input*		[> *output*]
nwnsi	*input*		[> *output*]
mafft-profile	*group1*	*group2*	[> *output*]

input, *group1* and *group2* must be in FASTA format.

DESCRIPTION

MAFFT is a multiple sequence alignment program for unix-like operating systems. It offers a range of multiple alignment methods.

Accuracy-oriented methods:

*L-INS-i (probably most accurate; recommended for <200 sequences; iterative refinement method incorporating local pairwise alignment information):

mafft --localpair --maxiterate *1000 input* [> *output*]

linsi *input* [> *output*]

*G-INS-i (suitable for sequences of similar lengths; recommended for <200 sequences; iterative refinement method incorporating global pairwise alignment information):

mafft --globalpair --maxiterate *1000 input* [> *output*]

ginsi *input* [> *output*]

*E-INS-i (suitable for sequences containing large unalignable regions; recommended for <200 sequences):

mafft --ep *0* **--genafpair --maxiterate** *1000 input* [> *output*]

einsi *input* [> *output*]

For E-INS-i, the **--ep** *0* option is recommended to allow large gaps.

Speed-oriented methods:

*FFT-NS-i (iterative refinement method; two cycles only):

mafft --retree *2* **--maxiterate** *2 input* [> *output*]

fftnsi *input* [> *output*]

*FFT-NS-i (iterative refinement method; max. 1000 iterations):

mafft --retree *2* **--maxiterate** *1000 input* [> *output*]

*FFT-NS-2 (fast; progressive method):

mafft --retree *2* **--maxiterate** *0 input* [> *output*]

fftns *input* [> *output*]

*FFT-NS-1 (very fast; recommended for >2000 sequences; progressive method with a rough guide tree):

mafft --retree *1* **--maxiterate** *0 input* [> *output*]

*NW-NS-i (iterative refinement method without FFT approximation; two cycles only):

mafft --retree *2* **--maxiterate** *2* **--nofft** *input* [> *output*]

nwnsi *input* [> *output*]

*NW-NS-2 (fast; progressive method without the FFT approximation):

mafft --retree *2* **--maxiterate** *0* **--nofft** *input* [> *output*]

nwns *input* [> *output*]

*NW-NS-PartTree-1 (recommended for ~10,000 to ~50,000 sequences; progressive method with the PartTree algorithm):

mafft --retree *1* **--maxiterate** *0* **--nofft --parttree** *input* [> *output*]

Group-to-group alignments

mafft-profile *group1 group2* [> *output*]

or:

mafft --maxiterate *1000* **--seed** *group1* **--seed** *group2* /dev/null [> *output*]

OPTIONS

Algorithm

--auto

Automatically selects an appropriate strategy from L-INS-i, FFT-NS-i and FFT-NS-2, according to data size. Default: off (always FFT-NS-2)

--6merpair

Distance is calculated based on the number of shared 6mers. Default: on

--globalpair

All pairwise alignments are computed with the Needleman-Wunsch algorithm. More accurate but slower than --6merpair. Suitable for a set of globally alignable sequences. Applicable to up to ~200 sequences. A combination with --maxiterate 1000 is recommended (G-INS-i). Default: off (6mer distance is used)

--localpair

All pairwise alignments are computed with the Smith-Waterman algorithm. More accurate but slower than --6merpair. Suitable for a set of locally alignable sequences. Applicable to up to ~200 sequences. A combination with --maxiterate 1000 is recommended (L-INS-i). Default: off (6mer distance is used)

--genafpair

All pairwise alignments are computed with a local algorithm with the generalized affine gap cost (Altschul 1998). More accurate but slower than --6merpair. Suitable when large internal gaps are expected. Applicable to up to ~200 sequences. A combination with --maxiterate 1000 is recommended (E-INS-i). Default: off (6mer distance is used)

--fastapair

All pairwise alignments are computed with FASTA (Pearson and Lipman 1988). FASTA is required. Default: off (6mer distance is used)

--weighti *number*

Weighting factor for the consistency term calculated from pairwise alignments. Valid when either of --blobalpair, --localpair, --genafpair, --fastapair or --blastpair is selected. Default: 2.7

--retree *number*

Guide tree is built *number* times in the progressive stage. Valid with 6mer distance. Default: 2

--maxiterate *number*

number cycles of iterative refinement are performed. Default: 0

--fft

Use FFT approximation in group-to-group alignment. Default: on

--nofft

Do not use FFT approximation in group-to-group alignment. Default: off

--noscore

Alignment score is not checked in the iterative refinement stage. Default: off (score is checked)

--memsave

Use the Myers-Miller (1988) algorithm. Default: automatically turned on when the alignment length exceeds 10,000 (aa/nt).

--parttree

Use a fast tree-building method (PartTree, Katoh and Toh 2007) with the 6mer distance. Recommended for a large number (> ~10,000) of sequences are input. Default: off

--dpparttree

The PartTree algorithm is used with distances based on DP. Slightly more accurate and slower than --parttree. Recommended for a large number (> ~10,000) of sequences are input. Default: off

--fastaparttree

The PartTree algorithm is used with distances based on FASTA. Slightly more accurate and slower than --parttree. Recommended for a large number (> ~10,000) of sequences are input. FASTA is required. Default: off

--partsize *number*

The number of partitions in the PartTree algorithm. Default: 50

--groupsize *number*

Do not make alignment larger than *number* sequences. Valid only with the --*parttree options. Default: the number of input sequences

Parameter

--op *number*

Gap opening penalty at group-to-group alignment. Default: 1.53

--ep *number*

Offset value, which works like gap extension penalty, for group-to-group alignment. Deafult: 0.123

--lop *number*

Gap opening penalty at local pairwise alignment. Valid when the --localpair or --genafpair option is selected. Default: -2.00

--lep *number*

Offset value at local pairwise alignment. Valid when the --localpair or --genafpair option is selected. Default: 0.1

--lexp *number*

Gap extension penalty at local pairwise alignment. Valid when the --localpair or --genafpair option is selected. Default: -0.1

--LOP *number*

Gap opening penalty to skip the alignment. Valid when the --genafpair option is selected. Default: -6.00

--LEXP *number*

Gap extension penalty to skip the alignment. Valid when the --genafpair option is selected. Default: 0.00

--bl *number*

BLOSUM *number* matrix (Henikoff and Henikoff 1992) is used. *number*=30, 45, 62 or 80. Default: 62

--jtt *number*

JTT PAM *number* (Jones et al. 1992) matrix is used. *number*>0. Default: BLOSUM62

--tm *number*

Transmembrane PAM *number* (Jones et al. 1994) matrix is used. *number*>0. Default: BLOSUM62

--aamatrix *matrixfile*

Use a user-defined AA scoring matrix. The format of *matrixfile* is the same to that of BLAST. Ignored when nucleotide sequences are input. Default: BLOSUM62

--fmodel

Incorporate the AA/nuc composition information into the scoring matrix. Deafult: off

Output

--clustalout

Output format: clustal format. Default: off (fasta format)

--inputorder

Output order: same as input. Default: on

--reorder

Output order: aligned. Default: off (inputorder)

--treeout

Guide tree is output to the *input*.tree file. Default: off

--quiet

Do not report progress. Default: off

Input

--nuc

Assume the sequences are nucleotide. Deafult: auto

--amino

Assume the sequences are amino acid. Deafult: auto

--seed *alignment1* [--seed *alignment2* --seed *alignment3* ...]

Seed alignments given in *alignment_n* (fasta format) are aligned with sequences in *input*. The alignment within every seed is preserved.

FILES

Mafft stores the input sequences and other files in a temporary directory, which by default is located in */tmp*.

ENVIRONMENT

MAFFT_BINARIES

Indicates the location of the binary files used by mafft. By default, they are searched in */usr/local/lib/mafft*, but on Debian systems, they are searched in */usr/lib/mafft*.

FASTA_4_MAFFT

This variable can be set to indicate to mafft the location to the fasta34 program if it is not in the PATH.

SEE ALSO

mafft-homologs(1)

REFERENCES

In English

*Katoh and Toh (Bioinformatics 23:372-374, 2007) PartTree: an algorithm to build an approximate tree from a large number of unaligned sequences (describes the PartTree algorithm).

*Katoh, Kuma, Toh and Miyata (Nucleic Acids Res. 33:511-518, 2005) MAFFT version 5: improvement in accuracy of multiple sequence alignment (describes [ancestral versions of] the G-INS-i, L-INS-i and E-INS-i strategies)

*Katoh, Misawa, Kuma and Miyata (Nucleic Acids Res. 30:3059-3066, 2002) MAFFT: a novel method for rapid multiple sequence alignment based on fast Fourier transform (describes the FFT-NS-1, FFT-NS-2 and FFT-NS-i strategies)

In Japanese

*Katoh and Misawa (Seibutsubutsuri 46:312-317, 2006) Multiple Sequence Alignments: the Next Generation

*Katoh and Kuma (Kagaku to Seibutsu 44:102-108, 2006) Jissen-teki Multiple Alignment

AUTHORS

Kazutaka Katoh <katoh_at_bioreg.kyushu-u.ac.jp>

Wrote Mafft.

Charles Plessy <charles-debian-nospam_at_plessy.org>

Wrote this manpage in DocBook XML for the Debian distribution, using Mafft's homepage as a template.

COPYRIGHT

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1.Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

2.Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

3.The name of the author may not be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
