MSAA v1.0 - Manual

Carsten Kemena

February 21, 2013

1 Introduction

Multiple Sequence alignment analyzer (MSAA) is a very simple tool to analyze and modify your alignment. The special feature of this program is that it can handle several alignments in one go. This is especially helpful when you have many alignments in a directory and you want to convert them all at once or analyze them all the same way by just giving the directory.

2 Options

The MSAA program distinguished between three kinds of options:

- **General options:** These options regulate the input and output behavior of MSAA - the input/output files and formats.
- **Analysis options:** These options regulate the analyses done on the alignments.
- **Modification options:** These options regulate the modification of the given alignments.
- **Comparison options:** These options regulate the comparions of two or more alignments.

Basically all of this options can be combined freely. It is important to know that the analyses of the alignments are always done **after** the modification option. For example: If you specifying the extract options and the identity option, the identity will only be calculated for the sequences left after the extraction.

2.1 General options

There are several general options available:

Table 1: General input/output options which can be used with *msaa*.

option	default	effect
`-h (--help)`	-	Produces a simple help message
`-a (--alignments)`	-	Alignment file(s) or directory(s)
`-e (--extensions)`	-	File extension to consider when given a directory. If none given all files will be used.
`-o (--output)`	-	The file or directory to write the output to
`-f (--format)`	*FASTA*	The output format to use
`-s (--suffix)`	-	Replaces extension with a new one. In case no extension exists it will be added.
`-A (--analysis)`	-	File for analysis output
`-n (--num_threads)`	*1*	Number of threads to use
`-l (--logging)`	-	Log errors in this file
`--no-check`	-	No checking of the alignments will be done

Details

a,alignments: This option is required. It can be either one or more files or directories. In case of a directory each file inside this directory will be taken. The files to be used can be limited by using the extension option.

e,extension: This option can be used as a file filter when giving a directory. Several different endings can be given. All endings need the '.' (example: -e .fa .aln).

o,output: This can be either a file when only a single file is given or a directory when multiple files are given. If several alignments this has to be a directory or or the suffix option have to be given.

f,format: The format which should be used to write the alignment. Currently supported: fasta, clustalw, msf, phylip_i (Phylip interleaved), phylip_s (Phylip sequential).

s,suffix: This options allows to easily replace the file ending by a new one. The '.' isn't needed for this option (example: -s fa).

A,analysis: In case of using one of the analysis option the output can be written to a file using this option. If none is given output is written to standard output.

l,logging: A logfile is produced where errors/warnings will be written into.

no-check: Alignments will not be checked to be correct alignments (length/sequence characters). Might be useful to be turned off for very large alignments.

Examples

```
# Convert all alignments having a fileending ".msf" into clustalw format,
    write them into a new folder and change the fileending to aln
PROMPT: msaa -a <in_folder> -e .msf -o <out_folder> -f clustalw -s aln
```

2.2 Analysis

The analysis section provides some basic alignment analysis tools. The analysis is done after any modification option which made have been used. In the detailed mode results are given for each alignment.

Table 2: Alignment analysis options

option	default	effect
`-v (--average_only)`	-	Only the average is printed
`--no-average)`	-	Do not print an average
`--no-header)`	-	Do not print the header
`-i (--identity)`	-	Computes the average identity of the alignments
`-S (--score)`	-	Compute the Sum-of-Pairs score
`-L (--length)`	-	Computes the length of the alignment
`-m (--matrix)`	*BLOSUM62*	Score matrix to use!
`--gop`	*-11*	Gap opening costs
`--gep`	*-1*	Gap extension costs

Details

v,average_only: only print the average

no-average: do not print the average

no-header: do not print a header

i,identity Computes the average identity of the alignment using the following formula:

$$Id(A) = \frac{\#matches}{\#matches + \#mismatches}$$

Pairs containing a gap character are ignored. The average returned is based on two values. The first one is calculated as following: The % identity of each alignment is taken and the average is calculated. The second one is calculated by adding up the #matches and #mismatches of all alignments before the % identity is calculated.

S,score: Compute the Sum-of-Pairs score.

l,length Computes the length of the alignment.

m,matrix: This options defines the score matrix to use. For this the environment variable ”MSAA_DATA” has to point to a directory containing the actual matrices.

gop: Gap opening costs.

gep: Gap extension costs.

Examples

```
# calculating the identity of several alignments
PROMPT: msaa -a aln.msf aln.fa -i
NAME     id
aln.fa   41.0
aln.msf  68.6
AVG      54.8/54.9

# the same as abouve but not showing the values for each single alignment
PROMPT: msaa -a aln.msf aln.fa -e .msf -i -v
NAME     id
AVG      54.8/54.9

# Calculate identity and the sum-of-pairs score for two alignments but do
      not show the average
PROMPT: msaa -a aln.msf aln.fa -e .msf -i -S --no-average
NAME     id         S-o-P
aln.fa   41.0      1590.0
aln.msf  68.6      3357.0

# Calculate sum-of-pairs score with changed gap opening and gap extension
      consts for two alignments without showing the header
PROMPT: msaa -a aln.msf aln.fa -e .msf -S --no-header --gep -2 --gop -12
NAME     id         S-o-P
aln.fa   41.0      1590.0
aln.msf  68.6      3357.0
```

2.3 Alignment modification

Table 3: Alignment modification options

option	default	effect
`--upper`		Turns all sequences to uppercase
`--lower`		Turns all sequences to lowercase
`-E (--extract`		Sequences to extract
`-c (--column_trim)`	-	Deletes the columns which contain more gaps then the given percentage
`-D (--delete_gaps)`		Turns back the alignment into sequences by deleteing all gap characters.
`-t (--seq_trim)`	-	trim_option threshold. The trimming option can be one of the following min_cov, max_cov, min_sim, max_sim

Details

upper: Turns all nucleotides to uppercase

lower: Turns all nucleotides to lowercase

E,extract: The names of sequences to extract from the alignment. This is the first modification done.

c,column_trim: Deletes the columns which contain more gaps then the given percentage. If used in combination with option **-e** this modification will applied on the extracted sequences only.

D,delete_gaps: Deletes the gaps from the alignment and returns only the sequences.

2.4 Alignment comparison

This options allows to compare two alignments with each other. The test alignments are given using the **-a**: option. Several alignments can be compared to a single reference alignment. The output of this comparison is the percentage of of pairs/columns in the reference alignment which exist as well in the test alignment.

Details

r,ref_aln: The reference alignment.

G,gap_limit: Only columns with less then the given percentages of gaps are considered.

Table 4: Alignment comparison options

option	default	effect
`-r (--ref_aln)`		The reference alignment
`-G (--gap_limit)`	*100*	Columns with less then the given percentages of gaps are considered
`-C (--compare_mode)`	*pair*	Compare mode: pair/column
`-q (--ignore_mis_seqs)`		Ignores missing sequences in found in the reference but not in the test alignment

C,compare_mode: Allowed values are: ”pair” and ”column”. In the pair mode each pair of nucleotides in considered. Pairs containing a gap are ignored. In the column mode whole columns are checked if they exist the same way in the test alignment.

Examples

```
# Compare an alignment to a reference
PROMPT: msaa -a <test_alignment> -r <reference_alignment>

# Compare an alignment to a reference using the column score and using
    only columns with less than 30% gaps
PROMPT: msaa -a <test_alignment> -r <reference_alignment> -C column -G 30
```