Package 'msa'

October 9, 2015

Type Package

Title Multiple Sequence Alignment

Version 1.0.2

Date 2015-09-29

Author Enrico Bonatesta, Christoph Horejs-Kainrath, Ulrich Bodenhofer

Maintainer Ulrich Bodenhofer <bodenhofer@bioinf.jku.at>

Description This package provides a unified R/Bioconductor interface to the multiple sequence alignment algorithms ClustalW, ClustalOmega, and Muscle. All three algorithms are integrated in the package, therefore, they do not depend on any external software tools and are available for all major platforms. The multiple sequence alignment algorithms are complemented by a function for pretty-printing multiple sequence alignments using the LaTeX package TeXshade.

URL http://www.bioinf.jku.at/software/msa/

License GPL (>= 2)

Copyright See file inst/COPYRIGHT

Depends R (>= 3.1.0), methods, Biostrings (>= 2.30.0)

Imports Rcpp (>= 0.11.1), BiocGenerics, IRanges (>= 1.20.0), S4Vectors, tools

Suggests Biobase, knitr

LinkingTo Rcpp

SystemRequirements

VignetteBuilder knitr

LazyLoad yes

Collate AllClasses.R AllGenerics.R params-methods.R version-methods.R helperFunctions.R inputChecks.R convertRows.R msaPrettyPrint.R print-methods.R show-methods.R msa.R msaMuscle.R msaClustalW.R msaClustalOmega.R

biocViews MultipleSequenceAlignment, Alignment, MultipleComparison, Sequencing

NeedsCompilation yes

R topics documented:

`msa-package` *Multiple Sequence Alignment*

Description

The **msa** package provides a unified R/Bioconductor interface to different multiple sequence alignment algorithms. Currently, 'ClustalW', 'ClustalOmega', and 'MUSCLE' are supported. All algorithms are usable without additional software packages and on all major platforms. The multiple sequence algorithms are complemented by an R interface to the powerful LaTeX package **texshade.sty** which allows for a highly customizable plots of multiple sequence alignments.

Details

Package:	msa
Type:	Package
Version:	1.0.2
Date:	2015-09-29
License:	GPL-2

Author(s)

Enrico Bonatesta, Christoph Horejs-Kainrath, and Ulrich Bodenhofer <msa@bioinf.jku.at>

References

http://www.bioinf.jku.at/software/msa

U. Bodenhofer, E. Bonatesta, C. Horejs-Kainrath, and S. Hochreiter (2015). msa: an R package for multiple sequence alignment. *Bioinformatics* (accepted). DOI: 10.1093/bioinformatics/btv494.

Thompson, J. D., Higgins, D. G., and Gibson, T. J. (1994) CLUSTAL W: improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position-specific gap penalties and weight matrix choice. *Nucleic Acids Res.* **22**(22):4673-4680. DOI: 10.1093/nar/22.22.4673.

Sievers, F., Wilm, A., Dineen, D., Gibson, T. J., Karplus, K., Li, W., Lopez, R., McWilliam, H., Remmert, M., Soeding, J., Thompson, J. D., and Higgins, D. G. (2011) Fast, scalable generation of high-quality protein multiple sequence alignments using Clustal Omega. *Mol. Syst. Biol.* **7**:539. DOI: 10.1038/msb.2011.75.

Edgar, R. C. (2004) MUSCLE: multiple sequence alignment with high accuracy and high throughput. *Nucleic Acids Res.* **32**(5):1792-1797. DOI: 10.1093/nar/gkh340.

Edgar, R. C. (2004) MUSCLE: a multiple sequence alignment method with reduced time and space complexity. *BMC Bioinformatics* **5**:113. DOI: 10.1186/1471-2105-5-113.

Beitz, E. (2000) TeXshade: shading and labeling of multiple sequence alignments using LaTeX2e *Bioinformatics* **16**(2):135-139. DOI: 10.1093/bioinformatics/16.2.135.

See Also

msa, msaClustalW, msaClustalOmega, msaMuscle, msaPrettyPrint

Examples

```
## read sequences
filepath <- system.file("examples", "exampleAA.fasta", package="msa")
mySeqs <- readAAStringSet(filepath)

## call unified interface msa() for default method (ClustalW) and
## default parameters
msa(mySeqs)
```

---

msa *Unified interface to multiple sequence alignment algorithms*

Description

The msa function provides a unified interface to the three multiple sequence alignment algorithms in this package: 'ClustalW', 'ClustalOmega', and 'MUSCLE'.

Usage

```
msa(inputSeqs, method=c("ClustalW", "ClustalOmega", "Muscle"),
    cluster="default", gapOpening="default",
    gapExtension="default", maxiters="default",
    substitutionMatrix="default", type="default",
    order=c("aligned", "input"), verbose=FALSE, help=FALSE,
    ...)
```

Arguments

inputSeqs	input sequences; this argument can be a character vector, an object of class XStringSet (includes the classes AAStringSet, DNAStringSet, and RNAStringSet), or a single character string with a file name. In the latter case, the file name is required to have the suffix '.fa' or '.fasta', and the file must be in FASTA format.
method	specifies the multiple sequence alignment to be used; currently, "ClustalW", "ClustalOmega", and "Muscle" are supported.
cluster	parameter related to sequence clustering; its interpretation and default value depends on the method; see msaClustalW, msaClustalOmega, or msaMuscle for algorithm-specific information.
gapOpening	gap opening penalty; the defaults are specific to the algorithm (see msaClustalW, and msaMuscle). Note that the sign of this parameter is ignored. The sign is automatically adjusted such that the called algorithm penalizes gaps instead of rewarding them.
gapExtension	gap extension penalty; the defaults are specific to the algorithm (see msaClustalW, and msaMuscle). Note that the sign of this parameter is ignored. The sign is automatically adjusted such that the called algorithm penalizes gaps instead of rewarding them.
maxiters	maximum number of iterations; its interpretation and default value depends on the method; see msaClustalW, msaClustalOmega, or msaMuscle for algorithm-specific information.
substitutionMatrix	substitution matrix for scoring matches and mismatches; format and defaults depend on the algorithm; see msaClustalW, msaClustalOmega, or msaMuscle for algorithm-specific information.
type	type of the input sequences inputSeqs; possible values are "dna", "rna", or "protein". In the original ClustalW implementation, this parameter is also called -type; "auto" is also possible in the original ClustalW, but, in this package, "auto" is deactivated. The type argument is mandatory if inputSeqs is a character vector or the file name of a FASTA file (see above). If inputSeqs is an object of class AAStringSet, DNAStringSet, or RNAStringSet, the type of sequences is determined by the class of inputSeqs and the type parameter is not necessary. If it is nevertheless specified and the type does not match the class of inputSeqs, the function stops with an error.
order	how the sequences should be ordered in the output object; if "aligned" is chosen, the sequences are ordered in the way the multiple sequence alignment algorithm orders them. If "input" is chosen, the sequences in the output object are ordered in the same way as the input sequences. For MUSCLE, the choice "input" is not available for sequence data that is read directly from a FASTA file. Even if sequences are supplied directly via R, the sequences must have unique names, otherwise the input order cannot be recovered. If the sequences do not have names or if the names are not unique, the msaMuscle function assignes generic unique names "Seq1"-Seqn to the sequences and issues a warning.
verbose	if TRUE, the algorithm displays detailed information and progress messages.

help — if TRUE, information about algorithm-specific parameters is displayed. In this case, no multiple sequence alignment is performed and the function quits after displaying the additional help information.

... — all other parameters are passed on to the multiple sequence algorithm, i.e. to one of the functions msaClustalW, msaClustalOmega, or msaMuscle. An overview of parameters that are available for the chosen method is shown when calling msa with help=TRUE. For more details, see also the documentation of chosen multiple sequence alignment algorithm.

Details

msa is a simple wrapper function that unifies the interfaces of the three functions msaClustalW, msaClustalOmega, and msaMuscle. Which function is called, is controlled by the method argument.

Note that the input sequences may be reordered by the multiple sequence alignment algorithms in order to group together similar sequences (see also description of argument order above). So, if the input order should be preserved or if the input order should be recovered later, we strongly recommend to always assign unique names to the input sequences. As noted in the description of the inputSeqs argument above, all functions, msa(), msaClustalW, msaClustalOmega, and msaMuscle, also allow for direct reading from FASTA files. This is mainly for the reason of memory efficiency if the sequence data set is very large. Otherwise, we want to encourage users to first read the sequences into the R workspace. If sequences are read from a FASTA file directly, the order of output sequences is completely under the control of the respective algorithm and does not allow for checking whether the sequences are named uniquely in the FASTA file. The preservation of the input order works also for sequence data read from a FASTA file, but only for ClustalW and ClustalOmega; MUSCLE does not support this (see also argument order above and msaMuscle).

Value

Depending on the type of sequences for which it was called, msa returns a MsaAAMultipleAlignment, MsaDNAMultipleAlignment, or MsaRNAMultipleAlignment object. If called with help=TRUE, msa returns an invisible NULL.

Author(s)

Enrico Bonatesta and Christoph Horejs-Kainrath <msa@bioinf.jku.at>

References

http://www.bioinf.jku.at/software/msa

U. Bodenhofer, E. Bonatesta, C. Horejs-Kainrath, and S. Hochreiter (2015). msa: an R package for multiple sequence alignment. *Bioinformatics* (accepted). DOI: 10.1093/bioinformatics/btv494.

http://www.clustal.org/download/clustalw_help.txt

http://www.clustal.org/omega/README

http://www.drive5.com/muscle/muscle.html

Thompson, J. D., Higgins, D. G., and Gibson, T. J. (1994) CLUSTAL W: improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position-specific gap penalties and weight matrix choice. *Nucleic Acids Res.* **22**(22):4673-4680. DOI: 10.1093/nar/22.22.4673.

Sievers, F., Wilm, A., Dineen, D., Gibson, T. J., Karplus, K., Li, W., Lopez, R., McWilliam, H., Remmert, M., Soeding, J., Thompson, J. D., and Higgins, D. G. (2011) Fast, scalable generation of high-quality protein multiple sequence alignments using Clustal Omega. *Mol. Syst. Biol.* **7**:539. DOI: 10.1038/msb.2011.75.

Edgar, R. C. (2004) MUSCLE: multiple sequence alignment with high accuracy and high throughput. *Nucleic Acids Res.* **32**(5):1792-1797. DOI: 10.1093/nar/gkh340.

Edgar, R. C. (2004) MUSCLE: a multiple sequence alignment method with reduced time and space complexity. *BMC Bioinformatics* **5**:113. DOI: 10.1186/1471-2105-5-113.

See Also

msaClustalW, msaClustalOmega, msaMuscle, msaPrettyPrint, MsaAAMultipleAlignment, MsaDNAMultipleAlignment, MsaRNAMultipleAlignment, MsaMetaData

Examples

```
## read sequences
filepath <- system.file("examples", "exampleAA.fasta", package="msa")
mySeqs <- readAAStringSet(filepath)

## call unified interface msa() for default method (ClustalW) and
## default parameters
msa(mySeqs)

## call ClustalOmega through unified interface
msa(mySeqs, method="ClustalOmega")

## call MUSCLE through unified interface with some custom parameters
msa(mySeqs, method="Muscle", gapOpening=12, gapExtension=3, maxiters=16,
    cluster="upgmamax", SUEFF=0.4, brenner=FALSE,
    order="input", verbose=FALSE)
```

---

msaClustalOmega *Multiple Sequence Alignment with ClustalOmega*

Description

This function calls the multiple sequence alignment algorithm ClustalOmega.

Usage

```
msaClustalOmega(inputSeqs, cluster="default",
                gapOpening="default", gapExtension="default",
                maxiters="default",  substitutionMatrix="default",
                type="default", order=c("aligned", "input"),
                verbose=FALSE, help=FALSE, ...)
```

Arguments

inputSeqs	input sequences; see msa. In the original ClustalOmega implementation, this parameter is called infile.
cluster	The cluster size which should be used. The default is 100. In the original ClustalOmega implementation, this parameter is called cluster-size.
gapOpening,gapExtension	ClustalOmega currently does not allow to adjust gap penalties; these arguments are only for future extensions and consistency with the other algorithms and msa. However, setting these parameters to values other than "default" will result in a warning.
maxiters	maximum number of iterations; the default value is 0 (no limitation). In the original ClustalOmega implementation, this parameter is called iterations.
substitutionMatrix	substitution matrix for scoring matches and mismatches; can be a real matrix, a file name, or the name of a built-in substitution matrix. In the latter case, the choices "BLOSUM30", "BLOSUM40", "BLOSUM50", "BLOSUM65", "BLOSUM80", and "Gonnet" are supported. This parameter is a new feature - the original ClustalOmega implementation does not allow for using a custom substitution matrix.
type	type of the input sequences inputSeqs; see msa.
order	how the sequences should be ordered in the output object (see msa); in the original ClustalW implementation, this parameter is called output-order.
verbose	if TRUE, the algorithm displays detailed information and progress messages.
help	if TRUE, information about algorithm-specific parameters is displayed. In this case, no multiple sequence alignment is performed and the function quits after displaying the additional help information.
...	further parameters specific to ClustalOmega; An overview of parameters that are available in this interface is shown when calling msaClustalOmega with help=TRUE. For more details, see also the documentation of ClustalOmega.

Details

This is a function providing the ClustalOmega multiple alignment algorithm as an R function. It can be used for various types of sequence data (see inputSeqs argument above). Parameters that are common to all multiple sequences alignments provided by the **msa** package are explicitly provided by the function and named in the same for all algorithms. Most other parameters that are specific to ClustalOmega can be passed to ClustalOmega via additional arguments (see argument help above).

For a note on the order of output sequences and direct reading from FASTA files, see msa.

Value

Depending on the type of sequences for which it was called, msaClustalOmega returns a MsaAAMultipleAlignment, MsaDNAMultipleAlignment, or MsaRNAMultipleAlignment object. If called with help=TRUE, msaClustalOmega returns an invisible NULL.

Author(s)

Enrico Bonatesta and Christoph Horejs-Kainrath <msa@bioinf.jku.at>

References

http://www.bioinf.jku.at/software/msa

U. Bodenhofer, E. Bonatesta, C. Horejs-Kainrath, and S. Hochreiter (2015). msa: an R package for multiple sequence alignment. *Bioinformatics* (accepted). DOI: 10.1093/bioinformatics/btv494.

http://www.clustal.org/omega/README

Sievers, F., Wilm, A., Dineen, D., Gibson, T. J., Karplus, K., Li, W., Lopez, R., McWilliam, H., Remmert, M., Soeding, J., Thompson, J. D., and Higgins, D. G. (2011) Fast, scalable generation of high-quality protein multiple sequence alignments using Clustal Omega. *Mol. Syst. Biol.* **7**:539. DOI: 10.1038/msb.2011.75.

See Also

msa, MsaAAMultipleAlignment, MsaDNAMultipleAlignment, MsaRNAMultipleAlignment, MsaMetaData

Examples

```
## read sequences
filepath <- system.file("examples", "exampleAA.fasta", package="msa")
mySeqs <- readAAStringSet(filepath)

## call msaClustalOmega with default values
msaClustalOmega(mySeqs)

## call msaClustalOmega with custom parameters
msaClustalOmega(mySeqs, auto=FALSE, cluster=120, dealign=FALSE,
                useKimura=FALSE, order="input", verbose=FALSE)
```

---

msaClustalW *Multiple Sequence Alignment with ClustalW*

Description

This function calls the multiple sequence alignment algorithm ClustalW.

Usage

```
msaClustalW(inputSeqs, cluster="default", gapOpening="default",
            gapExtension="default", maxiters="default",
            substitutionMatrix="default", type="default",
            order=c("aligned", "input"), verbose=FALSE,
            help=FALSE, ...)
```

Arguments

inputSeqs	input sequences; see msa. In the original ClustalW implementation, this parameter is called infile.
cluster	The clustering method which should be used. Possible values are "nj" (default) and "upgma". In the original ClustalW implementation, this parameter is called clustering.
gapOpening	gap opening penalty; the default value for nucleotide sequences is 15.0, the default value for amino acid sequences is 10.0.
gapExtension	gap extension penalty; the default value for nucleotide sequences is 6.66, the default value for amino acid sequences is 0.2.
maxiters	maximum number of iterations; the default value is 16. In the original ClustalW implementation, this parameter is called numiters.
substitutionMatrix	substitution matrix for scoring matches and mismatches; can be a real matrix, a file name, or the name of a built-in substitution matrix. In the latter case, the choices "blosum", "pam", "gonnet", and "id" are supported for amino acid sequences. For aligning nucleotide sequences, the parameter dnamatrix must be used instead. The valid choices for this parameter are "iub" and "clustalw". In the original ClustalW implementation, this parameter is called matrix.
type	type of the input sequences inputSeqs; see msa.
order	how the sequences should be ordered in the output object (see msa); in the original ClustalW implementation, this parameter is called outorder.
verbose	if TRUE, the algorithm displays detailed information and progress messages.
help	if TRUE, information about algorithm-specific parameters is displayed. In this case, no multiple sequence alignment is performed and the function quits after displaying the additional help information.
...	further parameters specific to ClustalW; An overview of parameters that are available in this interface is shown when calling msaClustalW with help=TRUE. For more details, see also the documentation of ClustalW.

Details

This is a function providing the ClustalW multiple alignment algorithm as an R function. It can be used for various types of sequence data (see inputSeqs argument above). Parameters that are common to all multiple sequences alignments provided by the **msa** package are explicitly provided by the function and named in the same for all algorithms. Most other parameters that are specific to ClustalW can be passed to ClustalW via additional arguments (see argument help above).

For a note on the order of output sequences and direct reading from FASTA files, see msa.

Value

Depending on the type of sequences for which it was called, msaClustalW returns a MsaAAMultipleAlignment, MsaDNAMultipleAlignment, or MsaRNAMultipleAlignment object. If called with help=TRUE, msaClustalW returns an invisible NULL.

Author(s)

Enrico Bonatesta and Christoph Horejs-Kainrath <msa@bioinf.jku.at>

References

http://www.bioinf.jku.at/software/msa

U. Bodenhofer, E. Bonatesta, C. Horejs-Kainrath, and S. Hochreiter (2015). msa: an R package for multiple sequence alignment. *Bioinformatics* (accepted). DOI: 10.1093/bioinformatics/btv494.

http://www.clustal.org/download/clustalw_help.txt

Thompson, J. D., Higgins, D. G., and Gibson, T. J. (1994) CLUSTAL W: improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position-specific gap penalties and weight matrix choice. *Nucleic Acids Res.* **22**(22):4673-4680. DOI: 10.1093/nar/22.22.4673.

See Also

msa, MsaAAMultipleAlignment, MsaDNAMultipleAlignment, MsaRNAMultipleAlignment, MsaMetaData

Examples

```
## read sequences
filepath <- system.file("examples", "exampleAA.fasta", package="msa")
mySeqs <- readAAStringSet(filepath)

## call msaClustalW with default values
msaClustalW(mySeqs)

## call msaClustalW with custom parameters
msaClustalW(mySeqs, gapOpening=1, gapExtension=1, maxiters=16,
            cluster="upgma", kimura=FALSE, order="input", maxdiv=23)
```

MsaMetaData-class *Class* MsaMetaData

Description

S4 class for storing metadata about multiple sequence alignment results

Objects

Objects of this virtual class are not be created and used directly. This is an auxiliary class used by the classes MsaAAMultipleAlignment, MsaDNAMultipleAlignment, and MsaRNAMultipleAlignment

Slots

The following slots are defined for `MsaMetaData` objects:

`version`: slot in which information is stored with which algorithm the multiple alignment has been computed along with its version number.

`params`: list in which the parameters are stored with which the multiple alignment algorithm has been executed.

`call`: the matched call with which the object was created

Methods

`version(object)`: accessor to the `version` slot

`params(x)`: accessor to the `params` slot

Author(s)

Enrico Bonatesta and Christoph Horejs-Kainrath <msa@bioinf.jku.at>

References

http://www.bioinf.jku.at/software/msa

U. Bodenhofer, E. Bonatesta, C. Horejs-Kainrath, and S. Hochreiter (2015). msa: an R package for multiple sequence alignment. *Bioinformatics* (accepted). DOI: 10.1093/bioinformatics/btv494.

See Also

`msa`, `msaClustalW`, `msaClustalOmega`, `msaMuscle`, `MsaAAMultipleAlignment`, `MsaDNAMultipleAlignment`, `MsaRNAMultipleAlignment`

Examples

```
## read sequences
filepath <- system.file("examples", "exampleAA.fasta", package="msa")
mySeqs <- readAAStringSet(filepath)

## simple call with default values
myAlignment <- msaClustalOmega(mySeqs)

## show the algorithm version with which the results were created
version(myAlignment)

## show the results
show(myAlignment)

## print the results
print(myAlignment, show="alignment")
print(myAlignment, show=c("alignment", "version"))
print(myAlignment, show="standardParams")
print(myAlignment, show="algParams")
print(myAlignment, show=c("call", "version"))
```

```
## show the params
params(myAlignment)
```

MsaMultipleAnlignmentClasses

Classes MsaAAMultipleAlignment, MsaDNAMultipleAlignment, *and* MsaRNAMultipleAlignment

---

Description

S4 classes for storing multiple alignments of amino acid, DNA, and RNA sequences along with algorithm metadata

Objects

Objects of these classes are returned by the multiple sequence alignment algorithms msaClustalW, msaClustalOmega, msaMuscle, and the wrapper function msa, all of which are provided by the **msa** package.

Details

The class MsaAAMultipleAlignment extends the AAMultipleAlignment class, the class MsaDNAMultipleAlignment extends the DNAMultipleAlignment class, and the class MsaRNAMultipleAlignment extends the RNAMultipleAlignment class. All three classes extend their parent classes by the slots contained in the MsaMetaData, i.e. all three classes are class unions of the aforementioned parent classes and the class MsaMetaData.

Methods

`print(x, show=c("alignment", "version", "call"))`: prints information about the object x; the show argument allows for determining what should be printed. The show must be a character vector and may contain any combination of the following five strings: if show contains "alignment", the multiple sequence alignment is printed using the corresponding method from the **Biostrings** package. If show contains "version", the version slot is shown. If show contains "call", the call slot is shown. If show contains "standardParams", the settings of the parameters that are common to all three multiple sequence alignment algorithms are shown. If show contains "algParams", the algorithm-specific parameters are shown. The order in which the strings are placed in the show argument does not have an effect on the order in which data are printed. The default is `show=c("alignment", "version", "call")`, i.e. by default, the multiple sequence alignment is shown along with version and call information.

`show(object)`: displays the alignment along with metadata; synonymous to calling print with default show argument `c("alignment", "version", "call")`

`version(object)`: displays the algorithm with which the multiple alignment has been computed along with its version number (see also MsaMetaData).

`params(x)`: accessor to the params slot (see also MsaMetaData)

Author(s)

Enrico Bonatesta and Christoph Horejs-Kainrath <msa@bioinf.jku.at>

References

http://www.bioinf.jku.at/software/msa

U. Bodenhofer, E. Bonatesta, C. Horejs-Kainrath, and S. Hochreiter (2015). msa: an R package for multiple sequence alignment. *Bioinformatics* (accepted). DOI: 10.1093/bioinformatics/btv494.

See Also

msa, msaClustalW, msaClustalOmega, msaMuscle, MsaMetaData

Examples

```
## read sequences
filepath <- system.file("examples", "exampleAA.fasta", package="msa")
mySeqs <- readAAStringSet(filepath)

## simple call with default values
myAlignment <- msaClustalOmega(mySeqs)

## show the algorithm version with which the results were created
version(myAlignment)

## show the results
show(myAlignment)

## print the results
print(myAlignment, show="alignment")
print(myAlignment, show=c("alignment", "version"))
print(myAlignment, show="standardParams")
print(myAlignment, show="algParams")
print(myAlignment, show=c("call", "version"))

## show the params
params(myAlignment)
```

---

msaMuscle *Multiple Sequence Alignment with MUSCLE*

Description

This function calls the multiple sequence alignment algorithm MUSCLE.

Usage

```
msaMuscle(inputSeqs, cluster="default", gapOpening="default",
          gapExtension="default", maxiters="default",
          substitutionMatrix="default",
          type="default", order=c("aligned", "input"),
          verbose=FALSE, help=FALSE, ...)
```

Arguments

inputSeqs	input sequences; see msa. In the original MUSCLE implementation, this parameter is called -in.
cluster	The clustering method which should be used. Possible values are "upgma", "upgmamax", "upgmamin", "upgmb", and "neighborjoining".
gapOpening	gap opening penalty; the default is 400 for DNA sequences and 420 for RNA sequences. The default for amino acid sequences depends on the profile score settings: for the setting le=TRUE, the default is 2.9, for sp=TRUE, the default is 1,439, and for sv=TRUE, the default is 300.
gapExtension	gap extension penalty; the default is 0.
maxiters	maximum number of iterations; the default is 16. In the original MUSCLE implementation, it is also possible to set maxiters to 0 which leads to an (out of memory) error. Therefore, maxiters=0 is not allowed in msaMuscle.
substitutionMatrix	substitution matrix for scoring matches and mismatches; can be a real matrix or a file name If the file interface is used, matrices have to be in NCBI-format. The original MUSCLE implementation also accepts matrices in WU_BLAST (AB_BLAST) format, but, due to copyright restrictions, this format is not supported by msaMuscle.
type	type of the input sequences inputSeqs; see msa.
order	how the sequences should be ordered in the output object (see msa for more details); the original MUSCLE implementation does not allow for preserving the order of input sequences. The msaMuscle function realizes this functionality by reverse matching of sequence names. Therefore, the sequences need to have unique names. If the sequences do not have names or if the names are not unique, the msaMuscle function assignes generic unique names "Seq1"-Seqn to the sequences and issues a warning. The choice "input" is not available at all for sequence data that is read directly from a FASTA file.
verbose	if TRUE, the algorithm displays detailed information and progress messages.
help	if TRUE, information about algorithm-specific parameters is displayed. In this case, no multiple sequence alignment is performed and the function quits after displaying the additional help information.
...	further parameters specific to MUSCLE; An overview of parameters that are available in this interface is shown when calling msaMuscle with help=TRUE. For more details, see also the documentation of MUSCLE.

Details

This is a function providing the MUSCLE multiple alignment algorithm as an R function. It can be used for various types of sequence data (see `inputSeqs` argument above). Parameters that are common to all multiple sequences alignments provided by the **msa** package are explicitly provided by the function and named in the same for all algorithms. Most other parameters that are specific to MUSCLE can be passed to MUSCLE via additional arguments (see argument `help` above).

For a note on the order of output sequences and direct reading from FASTA files, see `msa`.

Value

Depending on the type of sequences for which it was called, `msaMuscle` returns a `MsaAAMultipleAlignment`, `MsaDNAMultipleAlignment`, or `MsaRNAMultipleAlignment` object. If called with `help=TRUE`, `msaMuscle` returns an invisible `NULL`.

Author(s)

Enrico Bonatesta and Christoph Horejs-Kainrath <msa@bioinf.jku.at>

References

http://www.bioinf.jku.at/software/msa

U. Bodenhofer, E. Bonatesta, C. Horejs-Kainrath, and S. Hochreiter (2015). msa: an R package for multiple sequence alignment. *Bioinformatics* (accepted). DOI: 10.1093/bioinformatics/btv494.

http://www.drive5.com/muscle/muscle.html

Edgar, R. C. (2004) MUSCLE: multiple sequence alignment with high accuracy and high throughput. *Nucleic Acids Res.* **32**(5):1792-1797. DOI: 10.1093/nar/gkh340.

Edgar, R. C. (2004) MUSCLE: a multiple sequence alignment method with reduced time and space complexity. *BMC Bioinformatics* **5**:113. DOI: 10.1186/1471-2105-5-113.

See Also

`msa`, `MsaAAMultipleAlignment`, `MsaDNAMultipleAlignment`, `MsaRNAMultipleAlignment`, `MsaMetaData`

Examples

```
## read sequences
filepath <- system.file("examples", "exampleAA.fasta", package="msa")
mySeqs <- readAAStringSet(filepath)

## call msaMuscle with default values
msaMuscle(mySeqs)

## call msaMuscle with custom parameters
msaMuscle(mySeqs, gapOpening=12, gapExtension=3, maxiters=16,
          cluster="upgmamax", SUEFF=0.4, brenner=FALSE,
          order="input", verbose=FALSE)
```

`msaPrettyPrint` *Pretty-Printing of Multiple Sequence Alignments*

Description

The `msaPrettyPrint` function provides an R interface to the powerful LaTeX package **texshade.sty** which allows for a highly customizable plots of multiple sequence alignments.

Usage

```
msaPrettyPrint(x, y, output=c("pdf", "tex", "dvi", "asis"),
               subset=NULL, file=NULL, alFile=NULL,
               askForOverwrite=TRUE,  psFonts=FALSE, code=NA,
               paperWidth=11, paperHeight=8.5, margins=c(0.1, 0.3),
               shadingMode=c("identical", "similar", "functional"),
               shadingModeArg=NA,
               shadingColors=c("blues", "reds", "greens", "grays",
                               "black"),
               showConsensus=c("bottom", "top", "none"),
               consensusColors=c("ColdHot", "HotCold", "BlueRed",
                                 "RedBlue", "GreenRed",
                                 "RedGreen", "Gray"),
               consensusThreshold=50,
               showLogo=c("top", "bottom", "none"),
               logoColors=c("chemical", "rasmol", "hydropathy",
                            "structure", "standard area",
                            "accessible area"),
               showLogoScale=c("none", "leftright", "left",
                               "right"),
               showNames=c("left", "right", "none"),
               showNumbering=c("right", "left", "none"),
               showLegend=TRUE, furtherCode=NA, verbose=FALSE)
```

Arguments

`x`	an object of class `MultipleAlignment`, which includes the classes `MsaAAMultipleAlignment`, `MsaDNAMultipleAlignment`, and `MsaRNAMultipleAlignment`.
`y`	argument for restricting the output to a subset of columns; can be a numeric vector of length 2 with a lower and an upper bound or an object of class `IRanges`. If missing, the entire multiple alignment is printed.
`output`	type of output to be generated (see details below)
`subset`	can be used to specify a subset of sequences in the multiple alignment `x` if not all sequences should be printed.
`file`	name of output file; if no name is given, the name of the output file defaults to name of the object provided as argument `x` along with the proper suffix which depends on the type of output specified with the `output` argument. Note that this

might lead to invalid file names if not the name of an object, but an R expression is passed as argument `x`.

`alFile` name of alignment file to be created; `msaPrettyPrint` first writes the multiple alignment `x` to an alignment (`.aln`) file. The name of this file can be determined with the `alFile` argument. If no name is given, the name of the output file defaults to name of the object provided as argument `x` along with the suffix `.aln`. Note that this might lead to invalid file names if not the name of an object, but an R expression is passed as argument `x`.

`askForOverwrite`
if `TRUE` (default), `msaPrettyPrint` asks whether existing files should be overwritten or not. If `askForOverwrite` is set to `FALSE`, files are overwritten without further notice.

`psFonts` if `TRUE`, `msaPrettyPrint` produces LaTeX code that includes the LaTeX package **times.sty**; if `FALSE`, `msaPrettyPrint` produces LaTeX code based on the standard LaTeX fonts (default). Ignored for `output="asis"`.

`code` this argument can be used to specify the entire LaTeX code in the `texshade` environment. This overrides all arguments that customize the appearance of the output. Instead, all customizations must be done as LaTeX commands provided by the package **texshade.sty** directly. This option should only be used by expert users and for special applciations in which the possibilities of the customizations of the `msaPrettyPrint` function turn out to be insufficient.

`paperWidth,paperHeight`
paper format to be used in the resulting document; defaults to 11in x 8.5in (US letter in landscape orientation). Ignored for `output="asis"`.

`margins` a numeric vector of length 2 with the horizontal and vertical margins, respectively; the default is 0.1in for the horizontal and 0.3in for the vertical margin.

`shadingMode` shading mode; currently the shading modes `"identical"`, `"similar"`, and `"functional"` are supported (see documentation of **texshade.sty** for details).

`shadingModeArg` for shading modes `"identical"` and `"similar"`, `shadingModeArg` must be a single numeric threshold between 0 and 100 or two thresholds between 0 and 100 in increasing order. For shading mode `"functional"`, valid `shadingModeArg` arguments are `"charge"`, `"hydropathy"`, `"structure"`, `"chemical"`, `"rasmol"`, `"standard area"`, and `"accessible area"` (see documentation of **texshade.sty** for details).

`shadingColors` color scheme for shading; valid `"shadingColors"` arguments are `"blues"`, `"reds"`, `"greens"`, `"grays"`, and `"black"` (see documentation of **texshade.sty** for details).

`showConsensus` where to show the consensus sequence; possible values are `"bottom"`, `"top"`, and `"none"` (the latter option suppresses printing of the consensus sequence).

`consensusColors`
color scheme for printing the consensus sequence; the following choices are possible: `"ColdHot"`, `"HotCold"`, `"BlueRed"`, `"RedBlue"`, `"GreenRed"`, `"RedGreen"`, and `"Gray"` (see documentation of **texshade.sty** for details).

`consensusThreshold`
a single numeric between 0 and 100 (see documentation of **texshade.sty** for details)

showLogo	where to show a sequence logo; possible values are "top", "bottom", or "none" (the latter option suppresses printing of the consensus sequence). If a sequence logo and a consensus sequence should be shown together, they can only be located at opposite sides.
logoColors	color scheme for printing the sequence logo; the following choices are possible: "chemical", "rasmol", "hydropathy", "structure", "standard area", and "accessible area" (see documentation of **texshade.sty** for details).
showLogoScale	where to plot the vertical axis of the sequence logo; possible values are "left", "right", "leftright", and "none" (the latter option suppresses that the axis is displayed).
showNames	where to print sequence names; possible values are "left", "right", and "none" (the latter option suppresses that names are displayed).
showNumbering	where to print sequence numbers; possible values are "left", "right", and "none" (the latter option suppresses that numbers are displayed). If sequence names and numbers should be shown together, they can only be located at opposite sides.
showLegend	if TRUE (default), a legend is printed at the end of the alignment.
furtherCode	additional LaTeX code to be included in the texshade environment; all text passed as furtherCode is placed between the commands created by msaPrettyPrint and the end of the texshade environment. Note the difference to the code argument: while the code argument replaces all LaTeX code in the texshade environment, the code passed as furtherCode argument is added to the LaTeX code in the texshade environment.
verbose	if TRUE (default), progress messages are printed and also the output of running (PDF)LaTeX (if applicable) is printed to the R session.

Details

The msaPrettyPrint function writes a multiple alignment to an alignment (.aln) file and creates LaTeX code for pretty-printing the multiple alignment on the basis of the LaTeX package **texshade.sty**. If output="asis", msaPrettyPrint prints a LaTeX fragment consisting of the texshade environment to the console. The parameters described above can be used to customize the way the multiple alignment is formatted. If output="tex", a complete LaTeX file including preamble is created. For output="dvi" and output="pdf", the same kind of LaTeX file is created, but processed using (PDF)LaTeX to produce a final DVI or PDF file, respectively. The file argument be used to determine the file name of the final output file (except for the output="asis" which does not create an output file).

The choice output="asis" is particularly useful for Sweave or knitr documents. If msaPrettyPrint is called with output="asis" in a code chunk with results="tex" (Sweave) or results="asis" (knitr), then the resulting LaTeX fragment consisting of the texshade environment is directly included in the LaTeX document that is created from the Sweave/knitr document.

As noted above, if they are not specified explicitly, output file names are determined automatically. It is important to point out that all file names need to be LaTeX-compliant, i.e. no special characters and spaces (!) are allowed. If a file name would be invalid, msaPrettyPrint makes a default choice.

Note that texi2dvi and texi2pdf always save the resulting DVI/PDF files to the current working directory, even if the LaTeX source file is in a different directory. That is also the reason why the temporary file is created in the current working directory in the example below.

Value

msaPrettyPrint returns an invisible character vector consisting of the LaTeX fragment with the texshade environment.

Author(s)

Ulrich Bodenhofer, Enrico Bonatesta, and Christoph Horejs-Kainrath <msa@bioinf.jku.at>

References

http://www.bioinf.jku.at/software/msa

U. Bodenhofer, E. Bonatesta, C. Horejs-Kainrath, and S. Hochreiter (2015). msa: an R package for multiple sequence alignment. *Bioinformatics* (accepted). DOI: 10.1093/bioinformatics/btv494.

https://www.ctan.org/pkg/texshade

Beitz, E. (2000) TeXshade: shading and labeling of multiple sequence alignments using LaTeX2e *Bioinformatics* **16**(2):135-139. DOI: 10.1093/bioinformatics/16.2.135.

Examples

```
## read sequences
filepath <- system.file("examples", "exampleAA.fasta", package="msa")
mySeqs <- readAAStringSet(filepath)

## call unified interface msa() for default method (ClustalW) and
## default parameters
myAlignment <- msa(mySeqs)

## show resulting LaTeX code with default settings
msaPrettyPrint(myAlignment, output="asis", askForOverwrite=FALSE)

## create PDF file according to some custom settings
tmpFile <- tempfile(pattern="msa", tmpdir=".", fileext=".pdf")
tmpFile
msaPrettyPrint(myAlignment, file=tmpFile, output="pdf",
               showNames="left", showNumbering="none", showLogo="top",
               showConsensus="bottom", logoColors="rasmol",
               verbose=FALSE, askForOverwrite=FALSE)

## Not run:
library(Biobase)
openPDF(tmpFile)
## End(Not run)
```

Index