

msa

An R Package for Multiple Sequence Alignment

Enrico Bonatista, Christoph Horejs-Kainrath, and Ulrich Bodenhofer

Institute of Bioinformatics, Johannes Kepler University Linz
Altenberger Str. 69, 4040 Linz, Austria
msa@bioinf.jku.at

Version 1.2.1, December 19, 2015

Scope and Purpose of this Document

This document provides a gentle introduction into the R package `msa`. Not all features of the R package are described in full detail. Such details can be obtained from the documentation enclosed in the R package. Further note the following: (1) this is not an introduction to multiple sequence alignment or algorithms for multiple sequence alignment; (2) this is not an introduction to R or any of the Bioconductor packages used in this document. If you lack the background for understanding this manual, you first have to read introductory literature on the subjects mentioned above.

Contents

1	Introduction	4
2	Installation	4
3	msa for the Impatient	5
4	Functions for Multiple Sequence Alignment in More Detail	11
4.1	ClustalW-Specific Parameters	13
4.2	ClustalOmega-Specific Parameters	13
4.3	MUSCLE-Specific Parameters	14
5	Printing Multiple Sequence Alignments	14
6	Processing Multiple Alignments	21
7	Pretty-Printing Multiple Sequence Alignments	24
7.1	Consensus Sequence and Sequence Logo	25
7.2	Color Shading Modes	26
7.3	Subsetting	27
7.4	Additional Customizations	27
7.5	Sweave or knitr Integration	28
7.6	Further Caveats	29
8	Known Issues	29
9	Future Extensions	30
10	How to Cite This Package	30
11	Change Log	31

1 Introduction

Multiple sequence alignment is one of the most fundamental tasks in bioinformatics. Algorithms like ClustalW [10], ClustalOmega [9], and MUSCLE [2, 3] are well known and widely used. However, all these algorithms are implemented as stand-alone command line programs without any integration into the R/Bioconductor ecosystem. Before the `msa` package, only the `muscle` package has been available in R, but no other multiple sequence alignment algorithm, although the `Biostrings` package has provided data types for representing multiple sequence alignments for quite some time. The `msa` package aims to close that gap by providing a unified R interface to the multiple sequence alignment algorithms ClustalW, ClustalOmega, and MUSCLE. The package requires no additional software packages and runs on all major platforms. Moreover, the `msa` package provides an R interface to the powerful \LaTeX package `TeXshade` [1] which allows for a highly customizable plots of multiple sequence alignments. Unless some very special features of `TeXshade` are required, users can pretty-print multiple sequence alignments without the need to know the details of \LaTeX or `TeXshade`.

2 Installation

The `msa` R package (current version: 1.2.1) is available via Bioconductor. The simplest way to install the package is the following:

```
source("http://www.bioconductor.org/biocLite.R")
biocLite("msa")
```

To test the installation of the `msa` package, enter

```
library(msa)
```

in your R session. If this command terminates without any error message or warning, you can be sure that the `msa` package has been installed successfully. If so, the `msa` package is ready for use now and you can start performing multiple sequence alignments.

To make use of all functionalities of `msaPrettyPrint()`, a $\text{\TeX}/\text{\LaTeX}$ system [4] must be installed. To make use of \LaTeX code created by `msaPrettyPrint()` or to use the output of `msaPrettyPrint()` in Sweave [5] or knitr [11] documents, the \LaTeX package `TeXshade` (file `texshade.sty`) [1] must be accessible to the \LaTeX system too. The file `texshade.sty` is shipped with the `msa` package. To determine where the file is located, enter the following command in your R session:

```
system.file("tex", "texshade.sty", package="msa")
```

Alternatively, `TeXshade` can be installed directly from the Comprehensive \TeX Archive Network (CTAN).¹

¹<https://www.ctan.org/pkg/texshade>

3 msa for the Impatient

In order to illustrate the basic workflow, this section presents a simple example with default settings and without going into the details of each step. Let us first load amino acid sequences from one of the example files that are supplied with the `msa` package:

```
mySequenceFile <- system.file("examples", "exampleAA.fasta", package="msa")
mySequences <- readAAStringSet(mySequenceFile)
mySequences

## A AAStringSet instance of length 9
## width seq names
## [1] 452 MSTAVLENPGLGRKLS...NSEIGILCSALQKIK PH4H_Homo_sapiens
## [2] 453 MAAVLENGVLSRKLS...SEVGILCNALQKIKS PH4H_Rattus_norve...
## [3] 453 MAAVLENGVLSRKLS...SEVGILCHALQKIKS PH4H_Mus_musculus
## [4] 297 MNDRADFVVPDITTRK...LNAGDRQGWADTEDV PH4H_Chromobacter...
## [5] 262 MKTTQYVARQPDDNGF...RLGLHAPLFPPKQAA PH4H_Pseudomonas...
## [6] 451 MSALVLESRALGRKLS...SSEVEILCSALQKIK PH4H_Bos_taurus
## [7] 313 MAIATPTSAAPTPAPA...LNAGTREGWADTADI PH4H_Ralstonia_so...
## [8] 294 MSGDGLSNGPPPGARP...AYATAGGRLAGAAAG PH4H_Caulobacter...
## [9] 275 MSVAEYARDCAAQGLR...VARRKDQKALDPATV PH4H_Rhizobium_loti
```

Now that we have loaded the sequences, we can run the `msa()` function which, by default, runs ClustalW with default parameters:

```
myFirstAlignment <- msa(mySequences)

## use default substitution matrix

myFirstAlignment

## CLUSTAL 2.1
##
## Call:
## msa(mySequences)
##
## MsaAAMultipleAlignment with 9 rows and 456 columns
## aln names
## [1] MAAVLENGVLSRKLSDF...SINSEVGILCNALQKIKS PH4H_Rattus_norve...
## [2] MAAVLENGVLSRKLSDF...SINSEVGILCHALQKIKS PH4H_Mus_musculus
## [3] MSTAVLENPGLGRKLSDF...SINSEIGILCSALQKIK- PH4H_Homo_sapiens
## [4] MSALVLESRALGRKLSDF...SISSEVEILCSALQKIK- PH4H_Bos_taurus
## [5] -----...GWADTEDV----- PH4H_Chromobacter...
## [6] -----...GWADTADI----- PH4H_Ralstonia_so...
## [7] -----...AYATAGGRLAGAAAG--- PH4H_Caulobacter...
```

```
## [8] -----...----- PH4H_Pseudomonas_...
## [9] -----...----- PH4H_Rhizobium_loti
## Con -----...??????IL??A??-- Consensus
```

Obviously, the default printing function shortens the alignment for the sake of compact output. The `print()` function provided by the `msa` package provides some ways for customizing the output, such as, showing the entire alignment split over multiple blocks of sub-sequences:

```
print(myFirstAlignment, show="complete")

##
## MsaAAMultipleAlignment with 9 rows and 456 columns
##   aln (1..39)                               names
## [1] MAAVLENGVLSRKLSDFGQETSYIEDNSNQNGAISLIF PH4H_Rattus_norve...
## [2] MAAVLENGVLSRKLSDFGQETSYIEDNSNQNGAVSLIF PH4H_Mus_musculus
## [3] MSTAVLENPGLGRKLSDFGQETSYIEDNCNQNNGAISLIF PH4H_Homo_sapiens
## [4] MSALVLESRALGRKLSDFGQETSYIEGNSDQN-AVSLIF PH4H_Bos_taurus
## [5] ----- PH4H_Chromobacter...
## [6] ----- PH4H_Ralstonia_so...
## [7] ----- PH4H_Caulobacter_...
## [8] ----- PH4H_Pseudomonas_...
## [9] ----- PH4H_Rhizobium_loti
## Con ----- Consensus
##
##   aln (40..78)                               names
## [1] SLKEEVGALAKVLRRLFEEENDINLTHIESRPSRLNKDEYE PH4H_Rattus_norve...
## [2] SLKEEVGALAKVLRRLFEEENEINLTHIESRPSRLNKDEYE PH4H_Mus_musculus
## [3] SLKEEVGALAKVLRRLFEEENDVNLTHIESRPSRLKKDEYE PH4H_Homo_sapiens
## [4] SLKEEVGALARVLRRLFEEENDINLTHIESRPSRLRKDEYE PH4H_Bos_taurus
## [5] ----- PH4H_Chromobacter...
## [6] ----- PH4H_Ralstonia_so...
## [7] ----- PH4H_Caulobacter_...
## [8] ----- PH4H_Pseudomonas_...
## [9] ----- PH4H_Rhizobium_loti
## Con ----- Consensus
##
##   aln (79..117)                              names
## [1] FFTYLDKRTKPVLSGSIKSLRNDIGATVHELSDRDKKNT PH4H_Rattus_norve...
## [2] FFTYLDKRSKPVLSGSIKSLRNDIGATVHELSDRDKKNT PH4H_Mus_musculus
## [3] FFTHLDRSLPALTNIKILRHDIGATVHELSDRDKKNT PH4H_Homo_sapiens
## [4] FFTNLDQRSVPALANIKILRHDIGATVHELSDRDKKNT PH4H_Bos_taurus
## [5] ----- PH4H_Chromobacter...
## [6] ----- PH4H_Ralstonia_so...
## [7] ----- PH4H_Caulobacter_...
## [8] ----- PH4H_Pseudomonas_...
## [9] ----- PH4H_Rhizobium_loti
```

```

## Con ----- Consensus
##
##     aln (118..156)                               names
## [1] VPWFPRTIQELDRFANQILSYGAELDADHPGFKDPVYRA PH4H_Rattus_norve...
## [2] VPWFPRTIQELDRFANQILSYGAELDADHPGFKDPVYRA PH4H_Mus_musculus
## [3] VPWFPRTIQELDRFANQILSYGAELDADHPGFKDPVYRA PH4H_Homo_sapiens
## [4] VPWFPRTIQELDNFANQVLSYGAELDADHPGFKDPVYRA PH4H_Bos_taurus
## [5] -----MNDRADFVVPD-----ITTRKNVG PH4H_Chromobacter...
## [6] -----MAIATPTSAAPTAPAGFTGTLTDKLRQ PH4H_Ralstonia_so...
## [7] -----MSG-----DGLSNG PH4H_Caulobacter_...
## [8] -----MKTQY PH4H_Pseudomonas_...
## [9] -----MSVAEYAR-----DCAAQG PH4H_Rhizobium_loti
## Con -----?????????Y????D?????D????? Consensus
##
##     aln (157..195)                               names
## [1] RRKQFADIAYNYRHGQPIPRVEYTEEEKQTWGTVFRTLK PH4H_Rattus_norve...
## [2] RRKQFADIAYNYRHGQPIPRVEYTEEEKKTWGTVFRTLK PH4H_Mus_musculus
## [3] RRKQFADIAYNYRHGQPIPRVEYMEEEKKTWGTVFRTLK PH4H_Homo_sapiens
## [4] RRKQFADIAYNYRHGQPIPRVEYTEEEKKTWGTVFRTLK PH4H_Bos_taurus
## [5] LSHDAN-----DFTLPQPLDRYSAEDHATWATLYQRQC PH4H_Chromobacter...
## [6] FAEGLDGQTLRPDFTMEQPVHRYTAADHATWRTLYDRQE PH4H_Ralstonia_so...
## [7] PPPGAR-----PDWTIDQGWETYTQAEDVWITLYERQT PH4H_Caulobacter_...
## [8] VARQPD-----DNGFIHYPETEHQVWNTLITRQL PH4H_Pseudomonas_...
## [9] LRGDYS--VCRADFTVAQDYD--YSDEEQAVWRTLCDRQT PH4H_Rhizobium_loti
## Con ?R?Q????????????P?P???YTEEE??TW?TL??RQ? Consensus
##
##     aln (196..234)                               names
## [1] ALYKTHACYEHNHIFPLLEKYCGFREDNIPQLEDVVSQFL PH4H_Rattus_norve...
## [2] ALYKTHACYEHNHIFPLLEKYCGFREDNIPQLEDVVSQFL PH4H_Mus_musculus
## [3] SLYKTHACYEHNHIFPLLEKYCGFHEDNIPQLEDVVSQFL PH4H_Homo_sapiens
## [4] SLYKTHACYEHNHIFPLLEKYCGFREDNIPQLEEVVSQFL PH4H_Bos_taurus
## [5] KLLPGRACDEFMEGL----ERLEVDADRVPDFNKLNQKL PH4H_Chromobacter...
## [6] ALLPGRACDEFQGL----STLGMSREGVPSFDRLNETL PH4H_Ralstonia_so...
## [7] DMLHGRACDEFMRGL----DALDLHRSGIPDFARINEEL PH4H_Caulobacter_...
## [8] KVIEGRACQEYLDGI----EQLGLPHERIPQLDEINRVL PH4H_Pseudomonas_...
## [9] KLTRKLAHHSYLDGV----EKLGL-LDRIPDFEDVSTKL PH4H_Rhizobium_loti
## Con ?L????AC?E???G?----??LG???D?IPQLE?VSQ?L Consensus
##
##     aln (235..273)                               names
## [1] QTCTGFRLRPVAGLLSSRDFLGGLAFRVFHCTQYIRHGS PH4H_Rattus_norve...
## [2] QTCTGFRLRPVAGLLSSRDFLGGLAFRVFHCTQYIRHGS PH4H_Mus_musculus
## [3] QTCTGFRLRPVAGLLSSRDFLGGLAFRVFHCTQYIRHGS PH4H_Homo_sapiens
## [4] QSCTGFRLRPVAGLLSSRDFLGGLAFRVFHCTQYIRHGS PH4H_Bos_taurus
## [5] MAATGWKIVAVPGLIPDDVFFEHLANRRFPVTWVWLRPH PH4H_Chromobacter...
## [6] MRATGWQIVAVPGLVPDEVFFEHLANRRFPASWWMRRPD PH4H_Ralstonia_so...
## [7] KRLTGWTVAVPGLVPDDVFFDHLANRRFPAGQFIRKPH PH4H_Caulobacter_...

```

```

## [8] QATTGWRVARVPALIPFQTFPELLASQQFPVATFIRTPE PH4H_Pseudomonas_...
## [9] RKLTGWEIIAVPGLIPAAPFFDHLANRRFPVTNWLRTQ PH4H_Rhizobium_loti
## Con Q??TGWR??VPGL?P??FF??LA?R?FP?TQ?IR?? Consensus
##
##     aln (274..312)                                names
## [1] KPMYTPEPDICHELLGHVPLFSDRSFAQFSQEIG-LASL PH4H_Rattus_norve...
## [2] KPMYTPEPDICHELLGHVPLFSDRSFAQFSQEIG-LASL PH4H_Mus_musculus
## [3] KPMYTPEPDICHELLGHVPLFSDRSFAQFSQEIG-LASL PH4H_Homo_sapiens
## [4] KPMYTPEPDICHELLGHVPLFSDRSFAQFSQEIG-LASL PH4H_Bos_taurus
## [5] QLDYLQEPDVFHDLFGHVPLLINPVFADYLEAYGKGGVK PH4H_Chromobacter...
## [6] QLDYLQEPDGFHDIFGHVPLLINPVFADYMQAYQGGLK PH4H_Ralstonia_so...
## [7] ELDYLQEPDIFHDVFGHVPLMLTDPVFADYMQAYGEGRR PH4H_Caulobacter_...
## [8] ELDYLQEPDIFHEIFGHCPLLTNPWFAEFTHTYKGLGLK PH4H_Pseudomonas_...
## [9] ELDYIVEPDMFHDFFGHVPVLSQPVFADFMQMYGKKAGD PH4H_Rhizobium_loti
## Con ?LDY??EPDIFHELFGHVPLSDP?FA?F?Q?YG?LA?? Consensus
##
##     aln (313..351)                                names
## [1] GAPDEYIEKLATIIYWFTVEFGLCKEG-DSIKAYGAGLLS PH4H_Rattus_norve...
## [2] GAPDEYIEKLATIIYWFTVEFGLCKEG-DSIKAYGAGLLS PH4H_Mus_musculus
## [3] GAPDEYIEKLATIIYWFTVEFGLCKQG-DSIKAYGAGLLS PH4H_Homo_sapiens
## [4] GAPDEYIEKLATIIYWFTVEFGLCKQG-DSIKAYGAGLLS PH4H_Bos_taurus
## [5] AKALGALPMLARLYWYTVEFGLINTP-AGMRIYGAGILS PH4H_Chromobacter...
## [6] AARLGALDMLARLYWYTVEFGLIRTP-AGLRIYGAGIVS PH4H_Ralstonia_so...
## [7] ALGLGRLANLARLYWYTVEFGLMNTP-AGLRIYGAGIVS PH4H_Caulobacter_...
## [8] ASKE-ERVFLARLYWMTIEFGLVETD-QGKRIYGGGILS PH4H_Pseudomonas_...
## [9] IIALGDEMITRLYWYTAEYGLVQEAGQPLKAFGAGLMS PH4H_Rhizobium_loti
## Con ?A?????E?LARLYW?TVEFGL????-???KAYGAGLLS Consensus
##
##     aln (352..390)                                names
## [1] SFGELQYCLSD-KPKLLPLELEKTACQEYSVTEFQPLY Y PH4H_Rattus_norve...
## [2] SFGELQYCLSD-KPKLLPLELEKTACQEYTVTEFQPLY Y PH4H_Mus_musculus
## [3] SFGELQYCLSE-KPKLLPLELEKTAIQNYTVTEFQPLY Y PH4H_Homo_sapiens
## [4] SFGELQYCLSD-KPKLLPLELEKTAVQEYTITEFQPLY Y PH4H_Bos_taurus
## [5] SKSESIYCLDSASPNRVGFDLMRIMNTRYRIDTFQKTYF PH4H_Chromobacter...
## [6] SKSESVYALDSASPNRIGFDVHRIMRTRYRIDTFQKTYF PH4H_Ralstonia_so...
## [7] SRTESIFALDDPSNRI GFDLERVMRTLYRIDDFQQVYF PH4H_Caulobacter_...
## [8] SPKETVYSLSD-EPLHQAFNPLEAMRTPYRIDILQPLYF PH4H_Pseudomonas_...
## [9] SFTELQFAVEGKDAHHPFDLETVMRTGYEIDKFQRAYF PH4H_Rhizobium_loti
## Con SF?ELQYCLSD-?P??PF?LE??M?T?Y?ID?FQPLYF Consensus
##
##     aln (391..429)                                names
## [1] VAESFSDAKEKVRTFAATIPRPFSVRYDPYTQRVEVL DN PH4H_Rattus_norve...
## [2] VAESFNDAKEKVRTFAATIPRPFSVRYDPYTQRVEVL DN PH4H_Mus_musculus
## [3] VAESFNDAKEKVRNFAATIPRPFSVRYDPYTQRIEVL DN PH4H_Homo_sapiens
## [4] VAESFNDAKEKVRNFAATIPRPFSVHYDPYTQRIEVL DN PH4H_Bos_taurus
## [5] VIDSFKQLFDATA-PDFAPLYLQLADAQPWGAGDVAP DD PH4H_Chromobacter...

```



```
## [6] VIDSFELFDATR-PDFTPLYEALGTLPTFGAGDVVDGD PH4H_Ralstonia_so...
## [7] VIDSIQTLQEVTL-RDFGAIYERLASVSDIGVAEIVPGD PH4H_Caulobacter_...
## [8] VLPDLKRLFQLAQ-EDIMALVHEAMRLG-LHAPLFPPKQ PH4H_Pseudomonas_...
## [9] VLPSFDALRDAFQTADFEAIVARRKDQKALDPATV---- PH4H_Rhizobium_loti
## Con V??SF??L?E??R??D?T????????????P??????V?D? Consensus
##
##     aln (430..456)                names
## [1] TQQLKILADSINSEVGILCNALQKIKS PH4H_Rattus_norve...
## [2] TQQLKILADSINSEVGILCHALQKIKS PH4H_Mus_musculus
## [3] TQQLKILADSINSEIGILCSALQKIK- PH4H_Homo_sapiens
## [4] TQQLKILADSISSVEILCSALQKIK- PH4H_Bos_taurus
## [5] LVLNAGDRQGWADTEDV----- PH4H_Chromobacter...
## [6] AVLNAGTREGWADTADI----- PH4H_Ralstonia_so...
## [7] AVLTRGT-QAYATAGGRLAGAAAAG--- PH4H_Caulobacter_...
## [8] AA----- PH4H_Pseudomonas_...
## [9] ----- PH4H_Rhizobium_loti
## Con ?????????????????IL??A???--- Consensus
```

The `msa` package additionally offers the function `msaPrettyPrint()` which allows for pretty-printing multiple alignments using the \LaTeX package `TeXshade`. As an example, the following R code creates a PDF file `myFirstAlignment.pdf` which is shown in Figure 1:

```
msaPrettyPrint(myFirstAlignment, output="pdf", showNames="none",
               showLogo="none", askForOverwrite=FALSE, verbose=FALSE)
```

In the above call to `msaPrettyPrint()`, the printing of sequence names has been suppressed by `showNames="none"`. The settings `askForOverwrite=FALSE` and `verbose=FALSE` are necessary for building this vignette, but, in an interactive R session, they are not necessary.

Almost needless to say, the file names created by `msaPrettyPrint()` are customizable. By default, the name of the argument is taken as file name. More importantly, the actual output of `msaPrettyPrint()` is highly customizable, too. For more details, see the Section 7 and the help page of the function (`?msaPrettyPrint`).

The `msaPrettyPrint()` function is particularly useful for pretty-printing multiple sequence alignments in Sweave [5] or knitr [11] documents. More details are provided in Section 7. Here, we restrict to a teasing example:

```
msaPrettyPrint(myFirstAlignment, y=c(164, 213), output="asis",
               showNames="none", showLogo="none", askForOverwrite=FALSE)
```



```

IAYNYRHGQPIPRVEYTEEEKQTWGTVFRTLKALYKTHACYEHNHIFPLL 213
IAYNYRHGQPIPRVEYTEEERKTWGTVFRTLKALYKTHACYEHNHIFPLL 213
IAYNYRHGQPIPRVEYMEEEKKTWGTVFRTLKSLYKTHACYEYNHIFPLL 213
IAYNYRHGQPIPRVEYTEEEKKTWGTVFRTLKSLYKTHACYEHNHIFPLL 212
.....DFTLPQPLDRYSAEDHATWATLYQRQCKLLPGRACDEFMEGL... 67
QTLRPDFTMEQPVHRYTAADHATWRTLYDRQEALLPGRACDEFLLQGL... 83
....PDWTIDQGWETYTQAEHDVWITLYERQTDMLHGRACDEFMRGL... 58
.....DNGFIHYPETEHQVWNTLITRQLKVIIEGRACQEYLDGI... 50
.VCRADFTVAQDYD.YSDEEQAVWRTLCDRQTKLTRKLAHHSYLDGV... 65
          * *      !**** *! !*  ** * *      !* * *

```

X non conserved
X $\geq 50\%$ conserved

4 Functions for Multiple Sequence Alignment in More Detail

The example in Section 3 above simply called the function `msa()` without any additional arguments. We mentioned already that, in this case, ClustalW is called with default parameters. We can also explicitly request ClustalW or one of the two other algorithms ClustalOmega or MUSCLE:

```

myClustalWAlignment <- msa(mySequences, "ClustalW")

## use default substitution matrix

myClustalWAlignment

## CLUSTAL 2.1
##
## Call:
##   msa(mySequences, "ClustalW")
##
## MsaAAMultipleAlignment with 9 rows and 456 columns
##      aln                                     names
## [1] MAAVLENGVLSRKLSDFL...SINSEVGILCNALQKIKS PH4H_Rattus_norve...
## [2] MAAVLENGVLSRKLSDFL...SINSEVGILCHALQKIKS PH4H_Mus_musculus
## [3] MSTAVLENPGLGRKLSDFL...SINSEIGILCSALQKIK- PH4H_Homo_sapiens
## [4] MSALVLESRALGRKLSDFL...SISSEVEILCSALQKIK- PH4H_Bos_taurus
## [5] -----...GWADTEDV----- PH4H_Chromobacter...
## [6] -----...GWADTADI----- PH4H_Ralstonia_so...
## [7] -----...AYATAGGRLAGAAAG--- PH4H_Caulobacter_...
## [8] -----...----- PH4H_Pseudomonas_...
## [9] -----...----- PH4H_Rhizobium_loti
## Con -----...???????IL??A??- Consensus

myClustalOmegaAlignment <- msa(mySequences, "ClustalOmega")

```

```

## using Gonnet

myClustalOmegaAlignment

## ClustalOmega 1.2.0
##
## Call:
##   msa(mySequences, "ClustalOmega")
##
## MsaAAMultipleAlignment with 9 rows and 467 columns
##   aln                                     names
## [1] MSALVLESRALGRKLSDF...SISSEVEILCSALQKLK- PH4H_Bos_taurus
## [2] MSTAVLENPGLGRKLSDF...SINSEIGILCSALQKIK- PH4H_Homo_sapiens
## [3] MAAVLENGVLSRKLSD...SINSEVILCNALQKIKS PH4H_Rattus_norve...
## [4] MAAVLENGVLSRKLSD...SINSEVILCNALQKIKS PH4H_Mus_musculus
## [5] -----...----- PH4H_Pseudomonas_...
## [6] -----...----- PH4H_Rhizobium_loti
## [7] -----...LAGAAAG----- PH4H_Caulobacter_...
## [8] -----...V----- PH4H_Chromobacter...
## [9] -----...I----- PH4H_Ralstonia_so...
## Con -----...???????----- Consensus

myMuscleAlignment <- msa(mySequences, "Muscle")
myMuscleAlignment

## MUSCLE 3.8.31
##
## Call:
##   msa(mySequences, "Muscle")
##
## MsaAAMultipleAlignment with 9 rows and 460 columns
##   aln                                     names
## [1] MAAVLENGVLSRKLSD...SINSEVILCNALQKIKS PH4H_Rattus_norve...
## [2] MAAVLENGVLSRKLSD...SINSEVILCNALQKIKS PH4H_Mus_musculus
## [3] MSTAVLENPGLGRKLSDF...SINSEIGILCSALQKIK- PH4H_Homo_sapiens
## [4] MSALVLESRALGRKLSDF...SISSEVEILCSALQKLK- PH4H_Bos_taurus
## [5] -----...----- PH4H_Pseudomonas_...
## [6] -----...----- PH4H_Rhizobium_loti
## [7] -----...AYATAGGRLAGAAAG--- PH4H_Caulobacter_...
## [8] -----MNDRADF...QGWADTEDV----- PH4H_Chromobacter...
## [9] MAIATPTSAAPTAPAGF...EGWADTADI----- PH4H_Ralstonia_so...
## Con M????????????????DF...????????L??A???--- Consensus

```

Please note that the call `msa(mySequences, "ClustalW", ...)` is just a shortcut for the call `msaClustalW(mySequences, ...)`, analogously for `msaClustalOmega()` and `msaMuscle()`.

In other words, `msa()` is nothing else but a wrapper function that provides a unified interface to the three functions `msaClustalW()`, `msaClustalOmega()`, and `msaMuscle()`.

All three functions `msaClustalW()`, `msaClustalOmega()`, and `msaMuscle()` have the same parameters: The input sequences are passed as argument `inputSeqs`, and all functions have the following arguments: `cluster`, `gapOpening`, `gapExtension`, `maxiters`, `substitutionMatrix`, `order`, `type`, and `verbose`. The ways these parameters are interpreted, are largely analogous, although there are some differences, also in terms of default values. See the subsections below and the man page of the three functions for more details. All of the three functions `msaClustalW()`, `msaClustalOmega()`, and `msaMuscle()`, however, are not restricted to the parameters mentioned above. All three have a `'...'` argument through which several other algorithm-specific parameters can be passed on to the underlying library. The following subsections provide an overview of which parameters are supported by each of the three algorithms.

4.1 ClustalW-Specific Parameters

The original implementation of ClustalW offers a lot of parameters for customizing the way a multiple sequence alignment is computed. Through the `'...'` argument, `msaClustalW()` provides an interface to make use of most these parameters (see the documentation of ClustalW² for a comprehensive overview). Currently, the following restrictions and caveats apply:

- The parameters `infile`, `clustering`, `gapOpen`, `gapExt`, `numiters`, `matrix`, and `outorder` have been renamed to the standardized argument names `inputSeqs`, `cluster`, `gapOpening`, `gapExtension`, `maxiters`, `substitutionMatrix`, and `order` in order to provide a consistent interface for all three multiple sequence alignment algorithms.
- Boolean flags must be passed as logical values, e.g. `verbose=TRUE`.
- The parameter `quiet` has been replaced by `verbose` (with the exact opposite meaning).
- The following parameters are (currently) not supported: `bootstrap`, `check`, `fullhelp`, `interactive`, `maxseqLen`, `options`, and `tree`.
- For the parameter `output`, only the choice `"clustal"` is available.

4.2 ClustalOmega-Specific Parameters

In the same way as ClustalW, the original implementation of ClustalOmega also offers a lot of parameters for customizing the way a multiple sequence alignment is computed. Through the `'...'` argument, `msaClustalOmega()` provides an interface to make use of most these parameters (see the documentation of ClustalOmega³ for a comprehensive overview). Currently, the following restrictions and caveats apply:

- The parameters `infile`, `cluster-size`, `iterations`, and `output-order` have been renamed to the argument names `inputSeqs`, `cluster`, `maxiters`, and `order` in order to provide a consistent interface for all three multiple sequence alignment algorithms.

²http://www.clustal.org/download/clustalw_help.txt

³<http://www.clustal.org/omega/README>

- ClustalOmega does not allow for setting custom gap penalties. Therefore, setting the parameters `gapOpening` and `gapExtension` currently has no effect and will lead to a warning. These arguments are only defined for future extensions and consistency with the other algorithms available in `msa`.
- ClustalOmega only allows for choosing substitution matrices from a pre-defined set of names, namely "BLOSUM30", "BLOSUM40", "BLOSUM50", "BLOSUM65", "BLOSUM80", and "Gonnet". This is a new feature — the original ClustalOmega implementation does not allow for using any custom substitution matrix. However, since these are all amino acid substitution matrices, ClustalOmega is still hardly useful for multiple alignments of nucleotide sequences.
- Boolean flags must be passed as logical values, e.g. `verbose=TRUE`.
- The following parameters are (currently) not supported: `maxSeqLength` and `help`.
- For the parameter `outFmt`, only the choice "clustal" is available.

4.3 MUSCLE-Specific Parameters

Finally, also MUSCLE offers a lot of parameters for customizing the way a multiple sequence alignment is computed. Through the `'...'` argument, `msaMuscle()` provides an interface to make use of most these parameters (see the documentation of MUSCLE⁴ for a comprehensive overview). Currently, the following restrictions and caveats apply:

- The parameters `in`, `gapOpen`, `gapExtend`, `matrix`, and `seqtype` have been renamed to `inputSeqs`, `gapOpening`, `gapExtension`, `substitutionMatrix` and `type` in order to provide a consistent interface for all three multiple sequence alignment algorithms.
- Boolean flags must be passed as logical values, e.g. `verbose=TRUE`.
- The parameter `quiet` has been replaced by `verbose` (with the exact opposite meaning).
- The following parameters are currently not supported: `clw`, `clwstrict`, `fastaout`, `group`, `html`, `in1`, `in2`, `log`, `loga`, `msaout`, `msf`, `out`, `phyi`, `phyiout`, `phys`, `physout`, `refine`, `refinew`, `scorefile`, `spscore`, `stable`, `termgaps4`, `termgapsfull`, `termgapshalf`, `termgapshalflonger`, `tree1`, `tree2`, `usetree`, `weight1`, and `weight2`.

5 Printing Multiple Sequence Alignments

As already shown above, multiple sequence alignments can be shown in plain text format on the R console using the `print()` function (which is implicitly called if just the object name is entered on the R console). This function allows for multiple customizations, such as, enabling/disabling to display a consensus sequence, printing the entire alignment or only a subset, enabling/disabling to display sequence names, and adjusting the width allocated for sequence names. For more information, the reader is referred to the help page of the `print` function:

⁴<http://www.drive5.com/muscle/muscle.html>

```
help("print,MsaDNAMultipleAlignment-method")
```

We only provide some examples here:

```
print(myFirstAlignment)
```

```
## CLUSTAL 2.1
##
## Call:
##   msa(mySequences)
##
## MsaAAMultipleAlignment with 9 rows and 456 columns
##   aln                               names
## [1] MAAVLENGVLSRKLSDF...SINSEVGILCNALQKIKS PH4H_Rattus_norve...
## [2] MAAVLENGVLSRKLSDF...SINSEVGILCHALQKIKS PH4H_Mus_musculus
## [3] MSTAVLENPGLGRKLSDF...SINSEIGILCSALQKIK- PH4H_Homo_sapiens
## [4] MSALVLESRALGRKLSDF...SISSEVEILCSALQKLK- PH4H_Bos_taurus
## [5] -----...GWADTEDV----- PH4H_Chromobacter...
## [6] -----...GWADTADI----- PH4H_Ralstonia_so...
## [7] -----...AYATAGGRLAGAAAG--- PH4H_Caulobacter_...
## [8] -----...----- PH4H_Pseudomonas_...
## [9] -----...----- PH4H_Rhizobium_loti
## Con -----...???????IL??A???--- Consensus
```

```
print(myFirstAlignment, show="complete")
```

```
##
## MsaAAMultipleAlignment with 9 rows and 456 columns
##   aln (1..39)                               names
## [1] MAAVLENGVLSRKLSDFGQETSYIEDNSNQNGAISLIF PH4H_Rattus_norve...
## [2] MAAVLENGVLSRKLSDFGQETSYIEDNSNQNGAVSLIF PH4H_Mus_musculus
## [3] MSTAVLENPGLGRKLSDFGQETSYIEDNCNQNNGAISLIF PH4H_Homo_sapiens
## [4] MSALVLESRALGRKLSDFGQETSYIEGNSDQN-AVSLIF PH4H_Bos_taurus
## [5] ----- PH4H_Chromobacter...
## [6] ----- PH4H_Ralstonia_so...
## [7] ----- PH4H_Caulobacter_...
## [8] ----- PH4H_Pseudomonas_...
## [9] ----- PH4H_Rhizobium_loti
## Con ----- Consensus
##
##   aln (40..78)                               names
## [1] SLKEEVGALAKVLRRLFEEENDINLTHIESRPSRLNKDEYE PH4H_Rattus_norve...
## [2] SLKEEVGALAKVLRRLFEEENEINLTHIESRPSRLNKDEYE PH4H_Mus_musculus
## [3] SLKEEVGALAKVLRRLFEEENDVNLTHIESRPSRLKKDEYE PH4H_Homo_sapiens
## [4] SLKEEVGALARVLRRLFEEENDINLTHIESRPSRLRKDEYE PH4H_Bos_taurus
```

```

## [5] ----- PH4H_Chromobacter...
## [6] ----- PH4H_Ralstonia_so...
## [7] ----- PH4H_Caulobacter_...
## [8] ----- PH4H_Pseudomonas_...
## [9] ----- PH4H_Rhizobium_loti
## Con ----- Consensus
##
##      aln (79..117)                                names
## [1] FFTYLDKRTKPVLSGSIKSLRNDIGATVHELSDKEKNT PH4H_Rattus_norve...
## [2] FFTYLDKRSKPVLGSIKSLRNDIGATVHELSDKEKNT PH4H_Mus_musculus
## [3] FFTHLDRSLPALTNIKILRHDIGATVHELSDKKKDT PH4H_Homo_sapiens
## [4] FFTNLDQRSVPALANI KILRHDIGATVHELSDKKKDT PH4H_Bos_taurus
## [5] ----- PH4H_Chromobacter...
## [6] ----- PH4H_Ralstonia_so...
## [7] ----- PH4H_Caulobacter_...
## [8] ----- PH4H_Pseudomonas_...
## [9] ----- PH4H_Rhizobium_loti
## Con ----- Consensus
##
##      aln (118..156)                                names
## [1] VPWFPRTIQELDRFANQILSYGAELDADHPGFKDPVYRA PH4H_Rattus_norve...
## [2] VPWFPRTIQELDRFANQILSYGAELDADHPGFKDPVYRA PH4H_Mus_musculus
## [3] VPWFPRTIQELDRFANQILSYGAELDADHPGFKDPVYRA PH4H_Homo_sapiens
## [4] VPWFPRTIQELDNFANQVLSYGAELDADHPGFKDPVYRA PH4H_Bos_taurus
## [5] -----MNDRADFVVPD-----ITTRKNVG PH4H_Chromobacter...
## [6] -----MAIATPTSAAPTPAPAGFTGTLTDKLEEQ PH4H_Ralstonia_so...
## [7] -----MSG-----DGLSNG PH4H_Caulobacter_...
## [8] -----MKTQY PH4H_Pseudomonas_...
## [9] -----MSVAEYAR-----DCAAQG PH4H_Rhizobium_loti
## Con -----?????????Y?????D?????????D????? Consensus
##
##      aln (157..195)                                names
## [1] RRKQFADIAYNYRHGQPIPRVEYTEEEKQWGTWVFRSLK PH4H_Rattus_norve...
## [2] RRKQFADIAYNYRHGQPIPRVEYTEEEKQWGTWVFRSLK PH4H_Mus_musculus
## [3] RRKQFADIAYNYRHGQPIPRVEYMEEEKQWGTWVFKSLK PH4H_Homo_sapiens
## [4] RRKQFADIAYNYRHGQPIPRVEYTEEEKQWGTWVFRSLK PH4H_Bos_taurus
## [5] LSHDAN-----DFTLPQPLDRYSAEDHATWATLYQRQC PH4H_Chromobacter...
## [6] FAEGLDGQTLRPDFTMEQPVHRYTAADHATWRTLYDRQE PH4H_Ralstonia_so...
## [7] PPPGAR-----PDWTIDQGWETYTQAEDVWITLYERQT PH4H_Caulobacter_...
## [8] VARQPD-----DNGFIHYPETEHQVWNTLITRQL PH4H_Pseudomonas_...
## [9] LRGDYS--VCRADFTVAQDYD--YSDEEQAVWRTLCDRQT PH4H_Rhizobium_loti
## Con ?R?Q?????????????P?P???YTEEE??TW?TL??RQ? Consensus
##
##      aln (196..234)                                names
## [1] ALYKTHACYEHNHIFPLLEKYCGFREDNIPQLEDVVSQFL PH4H_Rattus_norve...
## [2] ALYKTHACYEHNHIFPLLEKYCGFREDNIPQLEDVVSQFL PH4H_Mus_musculus

```



```

## [3] SLYKTHACYEYNHIFPILLEKYCGFHEDNIPQLEDVSQFL PH4H_Homo_sapiens
## [4] SLYKTHACYEYNHIFPILLEKYCGFREDNIPQLEEVSQFL PH4H_Bos_taurus
## [5] KLLPGRACDEFMEGL----ERLEVDADRVPDFNKLNQKL PH4H_Chromobacter...
## [6] ALLPGRACDEFQLQGL----STLGMSREGVPSFDRLNETL PH4H_Ralstonia_so...
## [7] DMLHGRACDEFMRGL----DALDLHRSGIPDFARINEEL PH4H_Caulobacter_...
## [8] KVIEGRACQEYLDGI----EQLGLPHERIPQLDEINRVL PH4H_Pseudomonas_...
## [9] KLTRKLAHHSYLDGV----EKLGL-LDRIPDFEDVSTKL PH4H_Rhizobium_loti
## Con ?L????AC?E???G?----??LG???D?IPQLE?VSQ?L Consensus
##
##      aln (235..273)                                names
## [1] QTCTGFRLRPVAGLLSSRDFLGGLAFRVFHCTQYIRHGS PH4H_Rattus_norve...
## [2] QTCTGFRLRPVAGLLSSRDFLGGLAFRVFHCTQYIRHGS PH4H_Mus_musculus
## [3] QTCTGFRLRPVAGLLSSRDFLGGLAFRVFHCTQYIRHGS PH4H_Homo_sapiens
## [4] QSCTGFRLRPVAGLLSSRDFLGGLAFRVFHCTQYIRHGS PH4H_Bos_taurus
## [5] MAATGWKIVAVPGLIPDDVFFEHLANRRFPVTWWLREPH PH4H_Chromobacter...
## [6] MRATGWQIVAVPGLVPDEVFFEHLANRRFPASWWMRRPD PH4H_Ralstonia_so...
## [7] KRLTGWTVVAVPGLVPDDVFFDHLANRRFPAGQFIRKPH PH4H_Caulobacter_...
## [8] QATTGWRVARVPALIPFQTFELLASQQFPVATFIRTPE PH4H_Pseudomonas_...
## [9] RKLTGWEIIAVPGLIPAAPFFDHLANRRFPVTNWLRTQ PH4H_Rhizobium_loti
## Con Q???TGWR???VPGL?P???FF??LA?R?FP?TQ?IR??? Consensus
##
##      aln (274..312)                                names
## [1] KPMYTPEPDICHELLGHVPLFSDRSFAQFSQEIG-LASL PH4H_Rattus_norve...
## [2] KPMYTPEPDICHELLGHVPLFSDRSFAQFSQEIG-LASL PH4H_Mus_musculus
## [3] KPMYTPEPDICHELLGHVPLFSDRSFAQFSQEIG-LASL PH4H_Homo_sapiens
## [4] KPMYTPEPDICHELLGHVPLFSDRSFAQFSQEIG-LASL PH4H_Bos_taurus
## [5] QLDYLQEPDVFHDLFGHVPLLINPVFADYLEAYGKGGVK PH4H_Chromobacter...
## [6] QLDYLQEPDGFHDIFGHVPLLINPVFADYMQAYGQGGLK PH4H_Ralstonia_so...
## [7] ELDYLQEPDIFHDVFGHVPLMLTDPVFADYMQAYGEGGRR PH4H_Caulobacter_...
## [8] ELDYLQEPDIFHEIFGHCPLLTNPWFAEFTHTYKGLGLK PH4H_Pseudomonas_...
## [9] ELDYIVEPDMFHDFFGHVPLVLSQPVFADFMQMYGKKAGD PH4H_Rhizobium_loti
## Con ?LDY??EPDIFHELFGHVPLLSDP?FA?F?Q?YG?LA?? Consensus
##
##      aln (313..351)                                names
## [1] GAPDEYIEKLATIIYWFTVEFGLCKEG-DSIKAYGAGLLS PH4H_Rattus_norve...
## [2] GAPDEYIEKLATIIYWFTVEFGLCKEG-DSIKAYGAGLLS PH4H_Mus_musculus
## [3] GAPDEYIEKLATIIYWFTVEFGLCKQG-DSIKAYGAGLLS PH4H_Homo_sapiens
## [4] GAPDEYIEKLATIIYWFTVEFGLCKQG-DSIKAYGAGLLS PH4H_Bos_taurus
## [5] AKALGALPMLARLYWYTVEFGLINTP-AGMRIYGAGILS PH4H_Chromobacter...
## [6] AARLGALDMLARLYWYTVEFGLIRTP-AGLRIYGAGIVS PH4H_Ralstonia_so...
## [7] ALGLGRLANLARLYWYTVEFGLMNTP-AGLRIYGAGIVS PH4H_Caulobacter_...
## [8] ASKE-ERVFLARLYWMTIEFGLVETD-QGKRIYGGGILS PH4H_Pseudomonas_...
## [9] IIALGGDEMITRLYWYTAEYGLVQEAGQPLKAFGAGLMS PH4H_Rhizobium_loti
## Con ?A?????E?LARLYW?TVEFGL????-???KAYGAGLLS Consensus
##
##      aln (352..390)                                names

```

```

## [1] SFGELQYCLSD-KPKLLPLELEKTACQEYSVTEFQPLY PH4H_Rattus_norve...
## [2] SFGELQYCLSD-KPKLLPLELEKTACQEYTVTEFQPLY PH4H_Mus_musculus
## [3] SFGELQYCLSE-KPKLLPLELEKTAIQNYTVTEFQPLY PH4H_Homo_sapiens
## [4] SFGELQYCLSD-KPKLLPLELEKTAVQEYTITEFQPLY PH4H_Bos_taurus
## [5] SKSESIYCLDSASPNRVGFDMRIMNTRYRIDTFQKTYF PH4H_Chromobacter...
## [6] SKSESVYALDSASPNRIGFDVHRIMRTRYRIDTFQKTYF PH4H_Ralstonia_so...
## [7] SRTESIFALDDPSPNRIGFDLERVMRTLYRIDDFQVYF PH4H_Caulobacter_...
## [8] SPKETVYSLSD-EPLHQAFNPLEAMRTPYRIDILQPLYF PH4H_Pseudomonas_...
## [9] SFTELQFAVEGKDAHHVPFDLETVMRTGYEIDKFQRAYF PH4H_Rhizobium_loti
## Con SF?ELQYCLSD-?P???PF?LE??M?T?Y?ID?FQPLYF Consensus
##
##      aln (391..429)                names
## [1] VAESFSDAKEKVRTFAATIPRPFVRYDPYTQRVEVLDN PH4H_Rattus_norve...
## [2] VAESFNDAKEKVRTFAATIPRPFVRYDPYTQRVEVLDN PH4H_Mus_musculus
## [3] VAESFNDAKEKVRNFAATIPRPFVRYDPYTQRIEVLDN PH4H_Homo_sapiens
## [4] VAESFNDAKEKVRNFAATIPRPFVHYDPYTQRIEVLDN PH4H_Bos_taurus
## [5] VIDSFQKLFDATA-PDFAPLYLQLADAQPWGAGDVAPDD PH4H_Chromobacter...
## [6] VIDSFQKLFDATA-PDFTPLYEALGTLPTFGAGDVVDGD PH4H_Ralstonia_so...
## [7] VIDSIQTLQEVTL-RDFGAIYERLASVSDIGVAEIVPGD PH4H_Caulobacter_...
## [8] VLPDLKRLFQLAQ-EDIMALVHEAMRLG-LHAPLFPPKQ PH4H_Pseudomonas_...
## [9] VLPSFDALRDAFQTADFEAIVARRKDQKALDPATV---- PH4H_Rhizobium_loti
## Con V??SF??L?E??R??D?T????????????P??????V?D? Consensus
##
##      aln (430..456)                names
## [1] TQQLKILADSINSEVGILCNALQKIKS PH4H_Rattus_norve...
## [2] TQQLKILADSINSEVGILCHALQKIKS PH4H_Mus_musculus
## [3] TQQLKILADSINSEIGILCSALQKIK- PH4H_Homo_sapiens
## [4] TQQLKILADSISSVEILCSALQKIK- PH4H_Bos_taurus
## [5] LVLNAGDRQGWADTEDV----- PH4H_Chromobacter...
## [6] AVLNAGTREGWADTADI----- PH4H_Ralstonia_so...
## [7] AVLTRGT-QAYATAGGRLAGAAAAG--- PH4H_Caulobacter_...
## [8] AA----- PH4H_Pseudomonas_...
## [9] ----- PH4H_Rhizobium_loti
## Con ?????????????????IL??A??- Consensus

print(myFirstAlignment, showConsensus=FALSE, halfNrow=3)

## CLUSTAL 2.1
##
## Call:
##   msa(mySequences)
##
## MsaAAMultipleAlignment with 9 rows and 456 columns
##      aln                names
## [1] MAAVLENGVLSRKLSD...SINSEVGILCNALQKIKS PH4H_Rattus_norve...
## [2] MAAVLENGVLSRKLSD...SINSEVGILCHALQKIKS PH4H_Mus_musculus

```

```

## [3] MSTAVLENPGLGRKLSDF...SINSEIGILCSALQKIK- PH4H_Homo_sapiens
## ... ..
## [7] -----...AYATAGGRLAGAAAG--- PH4H_Caulobacter_...
## [8] -----...----- PH4H_Pseudomonas_...
## [9] -----...----- PH4H_Rhizobium_loti

print(myFirstAlignment, showNames=FALSE, show="complete")

##
## MsaAAMultipleAlignment with 9 rows and 456 columns
##     aln (1..60)
## [1] MAAVLENGVLSRKLSDFGQETSYIEDNSNQNGAISLIFSLKEEVGALAKVLRRLFEEENDI
## [2] MAAVLENGVLSRKLSDFGQETSYIEDNSNQNGAVSLIFSLKEEVGALAKVLRRLFEEENEI
## [3] MSTAVLENPGLGRKLSDFGQETSYIEDNCNQNNGAISLIFSLKEEVGALAKVLRRLFEEENDV
## [4] MSALVLESRALGRKLSDFGQETSYIEGNSDQN-AVSLIFSLKEEVGALARVLRRLFEEENDI
## [5] -----
## [6] -----
## [7] -----
## [8] -----
## [9] -----
## Con -----
##
##     aln (61..120)
## [1] NLTHIESRPSRLNKDEYEFFTYLDKRTKPVLSGSIKSLRNDIGATVHELSDRDKKNTVPW
## [2] NLTHIESRPSRLNKDEYEFFTYLDKRSKPVLSGSIKSLRNDIGATVHELSDRDKKNTVPW
## [3] NLTHIESRPSRLKKDEYEFFTHLDRSLPALTNIKILRHDIGATVHELSDRDKKNTVPW
## [4] NLTHIESRPSRLRKDEYEFFTNLDQRSVPALANI KILRHDIGATVHELSDRDKKNTVPW
## [5] -----
## [6] -----
## [7] -----
## [8] -----
## [9] -----
## Con -----
##
##     aln (121..180)
## [1] FPRTIQELDRFANQILSYGAELDADHPGFKDPVYRARRKQFADIAYNRHHGQPIPRVEYTT
## [2] FPRTIQELDRFANQILSYGAELDADHPGFKDPVYRARRKQFADIAYNRHHGQPIPRVEYTT
## [3] FPRTIQELDRFANQILSYGAELDADHPGFKDPVYRARRKQFADIAYNRHHGQPIPRVEYMT
## [4] FPRTIQELDNFANQVLSYGAELDADHPGFKDPVYRARRKQFADIAYNRHHGQPIPRVEYTT
## [5] -----MNDRADFVVPD-----ITTRKNVGLSHDAN-----DFTLPQPLDRYS
## [6] -----MAIATPTSAAPTPAPAGFTGTLTDKLRQFAEGLDGQTLRPDFTMEQPVHRYT
## [7] -----MSG-----DGLSNGPPPGAR-----PDWTIDQGWETYYT
## [8] -----MKTQYVARQPD-----DNGFIHYP
## [9] -----MSVAEYAR-----DCAAQGLRGDYS--VCRADFTVAQDYD--YS
## Con -----?????????Y?????D?????????D?????????R?Q?????????????P?P?????YT
##

```

```

##     aln (181..240)
## [1] EEEKQWGTVFRTLKALYKTHACYEHNHIFPALLEKYCGFREDNIPQLEDVVSQFLQTCTGF
## [2] EEERKTWGTVFRTLKALYKTHACYEHNHIFPALLEKYCGFREDNIPQLEDVVSQFLQTCTGF
## [3] EEEKKTWGTVFKTLKSLYKTHACYEYHNHIFPALLEKYCGFHEDNIPQLEDVVSQFLQTCTGF
## [4] EEEKKTWGTVFRTLKSLYKTHACYEHNHIFPALLEKYCGFREDNIPQLEEVVSQFLQSCTGF
## [5] AEDHATWATLYQRQCKLLPGRACDEFMEGL----ERLEVDADRVPDFNKLNQKLMAATGW
## [6] AADHATWRTL YDRQEALLPGRACDEFQLGGL----STLGMSREGVPSFDRNLNETLMRATGW
## [7] QAEHDVWITLYERQTDMLHGRACDEFMRGL----DALDLHRSGIPDFARINEELKRLTGW
## [8] ETEHQVWNTLITRQLKVI EGRACQEYLDGI----EQLGLPHERIPQLDEINRVLQATTGW
## [9] DEEQAVWRTLCDRQTKLTRKLAHHSYLDGV----EKLGL-LDRIPDFEDVSTKLRKLTGW
## Con EEE??TW?TL??RQ??L????AC?E???G?----??LG???D?IPQLE?VSQ?LQ??TGW
##
##     aln (241..300)
## [1] RLRPVAGLLSSRDFLGGLAFRVFHCTQYIRHGSKPMYTPEPDICHELLGHVPLFSDRSFA
## [2] RLRPVAGLLSSRDFLGGLAFRVFHCTQYIRHGSKPMYTPEPDICHELLGHVPLFSDRSFA
## [3] RLRPVAGLLSSRDFLGGLAFRVFHCTQYIRHGSKPMYTPEPDICHELLGHVPLFSDRSFA
## [4] RLRPVAGLLSSRDFLGGLAFRVFHCTQYIRHGSKPMYTPEPDICHELLGHVPLFSDRSFA
## [5] KIVAVPGLIPDDVFFEHLANRRFPVTWWLREPHQLDYLQEPDVFHDLFGHVPLLINPVFA
## [6] QIVAVPGLVPDEVFFEHLANRRFPASWMMRRPDQDYLQEPDGFHDIFGHVPLLINPVFA
## [7] TVVAVPGLVPDDVFFDHLANRRFPAGQFIRKPHELDYLQEPDIFHDVFGHVPMMLTDPVFA
## [8] RVARVPALIPFQTFEELLASQQFPVATFIRTPEELDYLQEPDIFHEIFGHCPLLTNPWFA
## [9] EIIAVPGLIPAAPFFDHLANRRFPVTNWL RTRQELDYIVEPDMFHDFFGHVPLSQPVFA
## Con R???VPGL?P???FF??LA?R?FP?TQ?IR????LDY??EPDIFHELFGHVPLSDP?FA
##
##     aln (301..360)
## [1] QFSQEIG-LASLGAPDEYIEKLATIIYWFTVEFGLCKEG-DSIKAYGAGLLSSFGE LQYCL
## [2] QFSQEIG-LASLGAPDEYIEKLATIIYWFTVEFGLCKEG-DSIKAYGAGLLSSFGE LQYCL
## [3] QFSQEIG-LASLGAPDEYIEKLATIIYWFTVEFGLCKQG-DSIKAYGAGLLSSFGE LQYCL
## [4] QFSQEIG-LASLGAPDEYIEKLATIIYWFTVEFGLCKQG-DSIKAYGAGLLSSFGE LQYCL
## [5] DYLEAYGKGGVKAKALGALPMLARLYWYTVEFGLINTP-AGMRIYGAGILSSKSESIYCL
## [6] DYMQAYGQGGLKAARLGALDMLARLYWYTVEFGLIRTP-AGLRIYGAGIVSSKSESVYAL
## [7] DYMQAYGEGRRALGLGRLANLARLYWYTVEFGLMNTP-AGLRIYGAGIVSSRTESIFAL
## [8] EFTHTYKGLGLKASKE-ERVFLARLYWMTIEFGLVETD-QGKRIYGGGILSSPKETVYSL
## [9] DFMQMYGKKAGDIIALGGDEMITRLYWYTA EYGLVQEAGQPLKAFGAGLMSSTELQFAV
## Con ?F?Q?YG?LA???A?????E?LARLYW?TVEFGL????-???KAYGAGLLSSF?ELQYCL
##
##     aln (361..420)
## [1] SD-KPKLLPLELEKTACQEYSVTEFQPLYYYVAESFSDAKEKVRTFAATIPRPFVRYDPY
## [2] SD-KPKLLPLELEKTACQEYTVTEFQPLYYYVAESFNDAKEKVRTFAATIPRPFVRYDPY
## [3] SE-KPKLLPLELEKTAIQNYTVTEFQPLYYYVAESFNDAKEKVRNFAATIPRPFVRYDPY
## [4] SD-KPKLLPLELEKTAVQEYTI TEFQPLYYYVAESFNDAKEKVRNFAATIPRPFVRYDPY
## [5] DSASPNRVGFDLMRIMNTRYRIDTFQKTYFVIDSFQKLFDATA-PDFAPLYLQLADAPW
## [6] DSASPNRIGFDVHRIMRTRYRIDTFQKTYFVIDSFQKLFDATA-PDFTPLYEALGTLPTF
## [7] DDPSPNRIGFDLERVMRTLYRIDDFQQVYFVIDSIQTLQEVTL-RDFGAIYERLASVSDI
## [8] SD-EPLHQAFNPLEAMRTPYRIDILQPLYFVLPDLKRLFQLAQ-EDIMALVHEAMRLG-L
## [9] EGKDAHHPFDLETVMRTGYEIDKFRAYFVLP SFDALRDAFQTADFEAIVARRKDQKAL

```

```
## Con SD-?P???PF?LE??M?T?Y?ID?FQPLYFV??SF??L?E??R??D?T??????????P?
##
##      aln (421..456)
## [1] TQRVEVDNTQQLKILADSINSEVGILCNALQKIKS
## [2] TQRVEVDNTQQLKILADSINSEVGILCHALQKIKS
## [3] TQRIEVDNTQQLKILADSINSEIGILCSALQKIK-
## [4] TQRIEVDNTQQLKILADSISSVEVEILCSALQKIK-
## [5] GAGDVAPDDLVLNAGDRQGWADTEDV-----
## [6] GAGDVVDGDAVLNAGTREGWADTADI-----
## [7] GVAEIVPGDAVLTRGT-QAYATAGGRLAGAAAG---
## [8] HAPLFPPKQAA-----
## [9] DPATV-----
## Con ?????V?D?????????????????IL??A???
```

6 Processing Multiple Alignments

The classes defined by the `msa` package for storing multiple alignment results have been derived from the corresponding classes defined by the `Biostings` package. Therefore, all methods for processing multiple alignments are available and work without any practical limitation. In this section, we highlight some of these.

The classes used for storing multiple alignments allow for defining masks on sequences and sequence positions via their row and column mask slots. They can be set by `rowmask()` and `colmask()` functions which serve both as setter and getter functions. To set row or column masks, an `IRanges` object must be supplied:

```
myMaskedAlignment <- myFirstAlignment
rowM <- IRanges(start=1, end=2)
rowmask(myMaskedAlignment) <- rowM
myMaskedAlignment

## CLUSTAL 2.1
##
## Call:
##      msa(mySequences)
##
## MsaAAMultipleAlignment with 9 rows and 456 columns
##      aln                                     names
## [1] #####...##### PH4H_Rattus_norve...
## [2] #####...##### PH4H_Mus_musculus
## [3] MSTAVLENPGLGRKLSDF...SINSEIGILCSALQKIK- PH4H_Homo_sapiens
## [4] MSALVLESRALGRKLSDF...SISSEVEILCSALQKIK- PH4H_Bos_taurus
## [5] -----...GWADTEDV----- PH4H_Chromobacter...
## [6] -----...GWADTADI----- PH4H_Ralstonia_so...
```

```
## [7] -----...AYATAGGRLAGAAAG--- PH4H_Caulobacter_...
## [8] -----...----- PH4H_Pseudomonas_...
## [9] -----...----- PH4H_Rhizobium_loti
## Con -----...???????IL??A???--- Consensus
```

The `unmasked()` allows for removing these masks, thereby casting the multiple alignment to a set of aligned Biostrings sequences (class `AAStringSet`, `DNAStrngSet`, or `RNAStringSet`):

```
unmasked(myMaskedAlignment)

## A AAStringSet instance of length 9
## width seq names
## [1] 456 MAAVLENGVLSRKLS...SEVGILCNALQKIKS PH4H_Rattus_norve...
## [2] 456 MAAVLENGVLSRKLS...SEVGILCHALQKIKS PH4H_Mus_musculus
## [3] 456 MSTAVLENPGLGRKLS...SEIGILCSALQKIK- PH4H_Homo_sapiens
## [4] 456 MSALVLESRALGRKLS...SEVEILCSALQKLK- PH4H_Bos_taurus
## [5] 456 -----...DTEDV----- PH4H_Chromobacter...
## [6] 456 -----...DTADI----- PH4H_Ralstonia_so...
## [7] 456 -----...TAGGRLAGAAAG--- PH4H_Caulobacter_...
## [8] 456 -----...----- PH4H_Pseudomonas_...
## [9] 456 -----...----- PH4H_Rhizobium_loti
```

Consensus matrices can be computed conveniently as follows:

```
conMat <- consensusMatrix(myFirstAlignment)
dim(conMat)

## [1] 21 456

conMat[, 101:110]

## [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10]
## - 5 5 5 5 5 5 5 5 5 5
## A 0 0 0 4 0 0 0 0 0 0
## C 0 0 0 0 0 0 0 0 0 0
## D 4 0 0 0 0 0 0 0 0 0
## E 0 0 0 0 0 0 0 4 0 0
## F 0 0 0 0 0 0 0 0 0 0
## G 0 0 4 0 0 0 0 0 0 0
## H 0 0 0 0 0 0 4 0 0 0
## I 0 4 0 0 0 0 0 0 0 0
## K 0 0 0 0 0 0 0 0 0 0
## L 0 0 0 0 0 0 0 0 4 0
## M 0 0 0 0 0 0 0 0 0 0
```

```
## N    0    0    0    0    0    0    0    0    0    0
## P    0    0    0    0    0    0    0    0    0    0
## Q    0    0    0    0    0    0    0    0    0    0
## R    0    0    0    0    0    0    0    0    0    0
## S    0    0    0    0    0    0    0    0    0    4
## T    0    0    0    0    4    0    0    0    0    0
## V    0    0    0    0    0    4    0    0    0    0
## W    0    0    0    0    0    0    0    0    0    0
## Y    0    0    0    0    0    0    0    0    0    0
```

Note that `consensusMatrix()` cannot handle alignments with active masks. So, the masks in multiple alignment objects must be removed prior to the computation of the consensus matrix:

```
conMat <- consensusMatrix(unmasked(myMaskedAlignment))
```

Consensus strings can be computed from consensus matrices:

```
## auxiliary function for splitting a string into displayable portions
printSplitString <- function(x, width=getOption("width") - 1)
{
  starts <- seq(from=1, to=nchar(x), by=width)

  for (i in 1:length(starts))
    cat(substr(x, starts[i], starts[i] + width - 1), "\n")
}

printSplitString(consensusString(conMat))

## -----
## -----?
## ??????????Y????D?????D?????R?Q????????????P?P???YTEEE??TW?TL??
## RQ??L????AC?E???G?----??LG???D?IPQLE?VSQ?LQ??TGWR???VPGL?P????FF?
## ?LA?R?FP?TQ?IR????LDY??EPDIFHELFGHVPLLSDP?FA?F?Q?YG?LA???A?????E
## ?LARLYW?TVEFGL????-???KAYGAGLLSSF?ELQYCLSD-?P???PF?LE??M?T?Y?ID?
## FQPLYFV??SF??L?E??R??D?T????????????P??????V?D????????????????IL?
## ?A???---
```

Consensus sequences can also be computed directly without computing intermediate consensus matrices. However, the `consensusString()` function cannot handle the masks contained in the multiple alignment objects (no matter whether there are active masks or not). Therefore, it is necessary to remove the masks beforehand:

```
printSplitString(consensusString(unmasked(myFirstAlignment)))
```

```
## -----
## -----?
## ?????????Y????D????????D??????R?Q????????????P?P???YTEEE??TW?TL??
## RQ??L????AC?E???G?----??LG???D?IPQLE?VSQ?LQ??TGWR???VPGL?P???FF?
## ?LA?R?FP?TQ?IR????LDY??EPDIFHELFGHVPLLSDP?FA?F?Q?YG?LA???A?????E
## ?LARLYW?TVEFGL????-???KAYGAGLLSSF?ELQYCLSD-?P???PF?LE??M?T?Y?ID?
## FQPLYFV??SF??L?E??R??D?T????????????P??????V?D????????????????IL?
## ?A???---

printSplitString(consensusString(unmasked(myMaskedAlignment)))

## -----
## -----?
## ?????????Y????D????????D??????R?Q????????????P?P???YTEEE??TW?TL??
## RQ??L????AC?E???G?----??LG???D?IPQLE?VSQ?LQ??TGWR???VPGL?P???FF?
## ?LA?R?FP?TQ?IR????LDY??EPDIFHELFGHVPLLSDP?FA?F?Q?YG?LA???A?????E
## ?LARLYW?TVEFGL????-???KAYGAGLLSSF?ELQYCLSD-?P???PF?LE??M?T?Y?ID?
## FQPLYFV??SF??L?E??R??D?T????????????P??????V?D????????????????IL?
## ?A???---
```

Actually, the `print()` method (see Section 5 above) uses this function to compute the consensus sequence.

7 Pretty-Printing Multiple Sequence Alignments

As already mentioned above, the `msa` package offers the function `msaPrettyPrint()` which allows for pretty-printing multiple sequence alignments using the \LaTeX package `TeXshade` [1]. Which prerequisites are necessary to take full advantage of the `msaPrettyPrint()` function is described in Section 2.

The `msaPrettyPrint()` function writes a multiple sequence alignment to an alignment (`.aln`) file and then creates \LaTeX code for pretty-printing the multiple sequence alignment on the basis of the \LaTeX package `TeXshade`. Depending on the choice of the output argument, the function `msaPrettyPrint()` either prints a \LaTeX fragment to the R session (choice `output="asis"`) or writes a \LaTeX source file (choice `output="tex"`) that it processes to a DVI file (choice `output="dvi"`) or PDF file (choice `output="pdf"`). Note that no extra software is needed for choices `output="asis"` and `output="tex"`. For `output="dvi"` and `output="pdf"`, however, a \TeX/\LaTeX distribution must be installed in order to translate the \LaTeX source file into the desired target format (DVI or PDF).

The function `msaPrettyPrint()` allows for making the most common settings directly and conveniently via an R interface without the need to know the details of \LaTeX or `TeXshade`. In the following, we will describe some of these customizations. For all possibilities, the user is referred to the documentation of `TeXshade`.⁵

⁵<https://www.ctan.org/pkg/texshade>

7.1 Consensus Sequence and Sequence Logo

The consensus sequence of the alignment is one of the most important results of a multiple sequence alignment. `msaPrettyPrint()` has a standard possibility to show this consensus sequence with the parameter `showConsensus`. The default value is "bottom", which results in the following:

```
msaPrettyPrint(myFirstAlignment, output="asis", y=c(164, 213),
               subset=c(1:6), showNames="none", showLogo="none",
               consensusColor="ColdHot", showLegend=FALSE,
               askForOverwrite=FALSE)
```

```
IAYNYRHGQPIPRVEYTEEEKQ TWGTVFRTLKALYKTHACYEHNHIFPLL 213
IAYNYRHGQPIPRVEYTEEERKTWGTVFRTLKALYKTHACYEHNHIFPLL 213
IAYNYRHGQPIPRVEYMEEEKKTWGTVFKTLKSLYKTHACYEYHNHIFPLL 213
IAYNYRHGQPIPRVEYTEEEKKTWGTVFRTLKSLYKTHACYEHNHIFPLL 212
.....DFTLPQPLDRYSAE DHATWATLYQRQCKLLPGRACDEFMEGL... 67
QTLRPDFTMEQPVHRYTAADHATWRTL YDRQEALLPGRACDEF LQGL... 83
*****!***!*** !!*!* ** !***!!*! *****
```

Consensus sequences can also be displayed on top of a multiple sequence alignment or omitted completely.

In the above example, an exclamation mark '!' in the consensus sequence stands for a conserved letter, i.e. a sequence positions in which all sequences agree, whereas an asterisk '*' stands for positions in which there is a majority of sequences agreeing. Positions in which the sequences disagree are left blank in the consensus sequence. For a more advanced example how to customize the consensus sequence, see the example in Subsection 7.4 below.

The color scheme of the consensus sequence can be configured with the `consensusColors` parameter. Possible values are "ColdHot", "HotCold", "BlueRed", "RedBlue", "GreenRed", "RedGreen", or "Gray". The above example uses the color scheme "RedGreen".

Additionally, `msaPrettyPrint()` also offers a more sophisticated visual representation of the consensus sequence — sequence logos. Sequence logos can be displayed either on top of the multiple sequence alignment (`showLogo="top"`), below the multiple sequence alignment (`showLogo="bottom"`), or omitted at all (`showLogo="none"`):

```
msaPrettyPrint(myFirstAlignment, output="asis", y=c(164, 213),
               subset=c(1:6), showNames="none", showLogo="top",
               logoColors="rasmol", shadingMode="similar",
               showLegend=FALSE, askForOverwrite=FALSE)
```


and "accessible area". The following example uses shading mode "functional" along with shadingModeArg set to "structure":

```
msaPrettyPrint(myFirstAlignment, output="asis", y=c(164, 213),
               showNames="none", shadingMode="functional",
               shadingModeArg="structure",
               askForOverwrite=FALSE)
```

```
I A Y N Y R H T P Q P R V E Y T E E E K Q T W G T V F R T L K A L Y K T H A C Y E H N H I F P L L 213
I A Y N Y R H G Q P I P R V E Y T E E E R K T W G T V F R T L K A L Y K T H A C Y E H N H I F P L L 213
I A Y N Y R H G Q P I P R V E Y M E E E K K T W G T V F K T L K S L Y K T H A C Y E Y N H I F P L L 213
I A Y N Y R H G Q P I P R V E Y T E E E K K T W G T V F R T L K S L Y K T H A C Y E H N H I F P L L 212
. . . . . D F T L P Q P L D R Y S A E D H A T W A T L Y Q R Q C K L L P G R A C D E F M E G L . . . 67
Q T L R P D F T M E Q P V H R Y T A A D H A T W R T L Y D R Q E A L L P G R A C D E F L Q G L . . . 83
. . . . . P D W T I D Q G W E T Y T Q A E H D V W I T L Y E R Q T D M L H G R A C D E F M R G L . . . 58
. . . . . . . . . . D N G F T H Y P E T E H Q V W N T L I T R Q L K V I E G R A C Q E Y L D G I . . . 50
. V C R A D F T V A Q D Y D . Y S D E E Q A V W R T L C D R Q T K L T R K L A H H S Y L D G V . . . 65
```

X external
 X ambivalent
 X internal

In the above example, a legend is shown that specifies the meaning of the color codes with which the letters are shaded. In some of the other examples above, we have suppressed this legend with the option `showLegend=FALSE`. The default, however, is that a legend is printed underneath the multiple sequence alignment like in the previous example.

7.3 Subsetting

In case that not the complete multiple sequence alignment should be printed, `msaPrettyPrint()` offers two ways of sub-setting. On the one hand, the `subset` argument allows for selecting only a subset of sequences. Not surprisingly, `subset` must be a numeric vector with indices of sequences to be selected. On the other hand, it is also possible to slice out certain positions of the multiple sequence alignment using the `y` argument. In the simplest case, `y` can be a numeric vector with two elements in ascending order which correspond to the left and right bounds between which the multiple sequence alignment should be displayed. However, it is also possible to slice out multiple windows. For this purpose, the argument `y` must be an `IRanges` object containing the starts and ends of the windows to be selected.

7.4 Additional Customizations

The `msaPrettyPrint()` function provides an interface to the most common functionality of `TeXshade` in a way that the user does not need to know the specific commands of `TeXshade`.

`TeXshade`, however, provides a host of additional customizations many of which are not covered by the interface of the `msaPrettyPrint()` function. In order to allow users to make use of all functionality of `TeXshade`, `msaPrettyPrint()` offers the `furtherCode` argument through which users can add `LaTeX` code to the `texshade` environment that is created by `msaPrettyPrint()`. Moreover, the `code` argument can be used to bypass all of `msaPrettyPrint()`'s generation of `TeXshade` code.

Here is an example how to use the `furtherCode` argument in order to customize the consensus sequence and to show a ruler on top:

```
msaPrettyPrint(myFirstAlignment, output="asis", y=c(164, 213),
               subset=c(1:6), showNames="none", showLogo="none",
               consensusColor="ColdHot", showLegend=FALSE,
               shadingMode="similar", askForOverwrite=FALSE,
               furtherCode=c("\\defconsensus{.}{lower}{upper}",
                             "\\showruler{1}{top}"))
```

```

      170          180          190          200          210
IAYNYRHGQPIPRVEYTEEEKQTWGTVFR TLKALYKTHACYEHNHIFPLL 213
IAYNYRHGQPIPRVEYTEEERKTWGTVFR TLKALYKTHACYEHNHIFPLL 213
IAYNYRHGQPIPRVEYMEEKKTWGTVFR TLKSLYKTHACYEYNNHIFPLL 213
IAYNYRHGQPIPRVEYTEEEKKTWGTVFR TLKSLYKTHACYEHNHIFPLL 212
.....DFTLPQPLDRYS AEDHATWATLYQRQCKLLPGRACDEFMEGL... 67
QTLRPDFTMEQPVHRYTAADHATWR TLYDRQEALLPGRACDEF LQGL... 83
iaynyrhgqpiPrveYteeek.TWgTvf r t l k . LykthACyE.nhifpll
```

7.5 Sweave or knitr Integration

The function `msaPrettyPrint()` is particularly well-suited for pretty-printing multiple alignments in Sweave [5] or knitr [11] documents. The key is to set `output` to "asis" when calling `msaPrettyPrint()` and, at the same time, to let the R code chunk produce output that is directly included in the resulting `LaTeX` document as it is. This can be accomplished with the code chunk option `results="tex"` in Sweave and with the code chunk option `results="asis"` in knitr. Here is an example of a Sweave code chunk that displays a pretty-printed multiple sequence alignment inline:

```
<<AnyChunkName,results="tex">>=
msaPrettyPrint(myFirstAlignment, output="asis")
@
```

The same example in knitr:

```
<<AnyChunkName,results="asis">>=
msaPrettyPrint(myFirstAlignment, output="asis")
@
```

Note that, for processing the resulting L^AT_EX source document, the T_EX`shade` package must be installed (see Section 2) and the T_EX`shade` package must be loaded in the preamble:

```
\usepackage{texshade}
```

7.6 Further Caveats

- Note that `texi2dvi()` and `ttexi2pdf()` always save the resulting DVI/PDF files to the current working directory, even if the L^AT_EX 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.
- T_EX`shade` has a wide array of functionalities. Only the most common ones have been tested for interoperability with R. So the use of the arguments `furtherCode` and `code` is the user's own risk!

8 Known Issues

Memory Leaks

The original implementations of ClustalW, ClustalOmega, and MUSCLE are stand-alone command line programs which are only run once each time a multiple sequence alignment is performed. During the development of the `msa` package, we performed memory management checks using Valgrind [7] and discovered multiple memory leaks in ClustalW and MUSCLE. These memory leaks have no effect for the command line tools, since the program is closed each time the alignment is finished. In the implementation of the `msa` package, however, these memory leaks may have an effect if the same algorithm is run multiple times.

For MUSCLE, we managed to eliminate all memory leaks by deactivating the two parameters `weight1` and `weight2`. ClustalOmega did not show any memory leaks. ClustalW indeed has several memory leaks which are benign if the algorithm is run only a few times, but which may have more severe effects if the algorithm is run many times. ClustalOmega also has a minor memory leak, but the loss of data is so small that no major problems are to be expected except for thousands of executions of ClustalOmega.

ClustalOmega vs. Older GCC Versions on Linux/Unix

We have encountered peculiar behavior of ClustalOmega if the package was built using an older GCC version: if we built the package on an `x86_64` Linux system with GCC 4.4.7, ClustalOmega built smoothly and could be executed without any errors. However, the resulting multiple sequence alignment was more than sub-optimal. We could neither determine the source of this problem nor which GCC versions show this behavior. We therefore recommend Linux/Unix users to use an up-to-date GCC version (we used 4.8.2 during package development, which worked nicely) or, in case they encounter dubious results, to update to a newer GCC version and re-install the package.

ClustalOmega: OpenMP Support on Mac OS

ClustalOmega is implemented to make use of OpenMP (if available on the target platform). Due to issues on one of the Bioconductor build servers running Mac OS, we had to deactivate OpenMP generally for Mac OS platforms. If a Mac OS user wants to re-activate OpenMP, he/she should download the source package tarball, untar it, comment/uncomment the corresponding line in `msa/src/ClustalOmega/msaMakefile` (see first six lines), and build/install the package from source.

9 Future Extensions

We envision the following changes/extensions in future versions of the package:

- Integration of more multiple sequence alignment algorithms, such as, T-Coffee [8] or DIALIGN [6]
- Support for retrieving guide trees from the multiple sequence alignment algorithms
- Interface to methods computing phylogenetic trees (e.g. as contained in the original implementation of ClustalW)
- Elimination of memory leaks described in Section 8 and re-activation of parameters that have been deactivated in order to avoid memory leaks
- More tolerant handling of custom substitution matrices (MUSCLE interface)

10 How to Cite This Package

If you use this package for research that is published later, you are kindly asked to cite it as follows:

U. Bodenhofer, E. Bonatesta, C. Horejš-Kainrath, and S. Hochreiter (2015). `msa`: an R package for multiple sequence alignment. *Bioinformatics* **31**(24):3997–3999. DOI: [bioinformatics/btv494](https://doi.org/10.1093/bioinformatics/btv494).

To obtain a BibTeX entries of the reference, enter the following into your R session:

```
toBibtex(citation("msa"))
```

Moreover, we insist that, any time you cite the package, you also cite the original paper in which the original algorithm has been introduced (see bibliography below).

11 Change Log

Version 1.2.1:

- fixes in Makefiles and Makevars files to account for changes in build system
- update of citation information

Version 1.2.0:

- new branch for Bioconductor 3.2 release

Version 1.1.3:

- bug fix related to custom substitution matrices in the MUSCLE interface
- corrections and updates of documentation

Version 1.1.2:

- new `print()` function for multiple alignments that also allows for displaying alignments in their entirety (plus additional customizations)
- strongly improved handling of custom substitution matrices by `msaClustalW()`: now custom matrices can also be supplied for nucleotide sequences which can also be passed via the `substitutionMatrix` argument. The `dnamatrix` argument is still available for the sake of backwards compatibility.
- strongly improved handling of custom substitution matrices by `msaMuscle()`
- fix of improperly aligned sequence logos produced by `msaPrettyPrint()`
- updated citation information

Version 1.1.1: fix of `msa()` function

Version 1.1.0: new branch for Bioconductor 3.2 devel

Version 1.0.0: first official release as part of Bioconductor 3.1

References

- [1] E. Beitz. `TEXshade`: shading and labeling of multiple sequence alignments using `LATEX2ε`. *Bioinformatics*, 16(2):135–139, 2000.
- [2] R. C. Edgar. MUSCLE: a multiple sequence alignment method with reduced time and space complexity. *BMC Bioinformatics*, 5(5):113, 2004.
- [3] R. C. Edgar. MUSCLE: multiple sequence alignment with high accuracy and high throughput. *Nucleic Acids Res.*, 32(5):1792–1797, 2004.
- [4] L. Lamport. *L^AT_EX — A Document Preparation System. User's Guide and Reference Manual*. Addison-Wesley Longman, Amsterdam, 1999.

- [5] F. Leisch. Sweave: dynamic generation of statistical reports using literate data analysis. In W. Härdle and B. Rönz, editors, *Compstat 2002 — Proceedings in Computational Statistics*, pages 575–580, Heidelberg, 2002. Physica-Verlag.
- [6] B. Morgenstern. DIALIGN 2: improvement of the segment-to-segment approach to multiple sequence alignment. *Bioinformatics*, 15(3):211–218, 1999.
- [7] N. Nethercote and J. Seward. Valgrind: A framework for heavyweight dynamic binary instrumentation. In *Proc. of the ACM SIGPLAN 2007 Conf. on Programming Language Design and Implementation*, San Diego, CA, 2007.
- [8] C. Notredame, D. G. Higgins, and J. Heringa. T-Coffee: A novel method for fast and accurate multiple sequence alignment. *J. Mol. Biol.*, 302(1):205–217, 2000.
- [9] F. Sievers, A. Wilm, D. Dineen, T. J. Gibson, K. Karplus, W. Li, R. Lopez, H. McWilliam, M. Remmert, J. Söding, J. D. Thompson, and D. G. Higgins. Fast, scalable generation of high-quality protein multiple sequence alignments using Clustal Omega. *Mol. Syst. Biol.*, 7:539, 2011.
- [10] J. D. Thompson, D. G. Higgins, and T. J. Gibson. 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, 2004.
- [11] Y. Xie. *Dynamic Documents with R and knitr*. Chapman & Hall/CRC, 2014.