

Package ‘Ryacas’

February 8, 2019

Version 0.4.1

Title R Interface to the Yacas Computer Algebra System

Maintainer Mikkel Meyer Andersen <mikl@math.aau.dk>

Encoding UTF-8

Description An interface to the yacas computer algebra system.

Depends R (>= 3.3.0)

Imports Rcpp (>= 0.12.0), xml2, stats, settings

LinkingTo Rcpp

Suggests knitr, rmarkdown, testthat, devtools, pkgload, Matrix

License GPL-2

URL <https://github.com/mikldk/ryacas>, <http://www.yacas.org>

BugReports <https://github.com/mikldk/ryacas/issues>

RoxygenNote 6.1.1

VignetteBuilder knitr

NeedsCompilation yes

Author Rob Goedman [aut],
Gabor Grothendieck [aut],
Søren Højsgaard [aut],
Ayal Pinkus [aut],
Grzegorz Mazur [aut],
Mikkel Meyer Andersen [aut, cre]

Repository CRAN

Date/Publication 2019-02-08 12:23:30 UTC

R topics documented:

Ryacas-package	2
as.Sym.character	3
as.Sym.matrix	4

bodyAsExpression	4
Eval	5
get_output_width	6
Ryacas_options	6
set_output_width	7
syacas	7
Sym	8
yacas	9
yacasTranslations	11
yacas_evaluate	13
yacmode	13

Index	15
--------------	-----------

Ryacas-package	<i>R interface to yacas computer algebra package</i>
----------------	--

Description

Ryacas allows one to use the yacas computer algebra package entirely from within R. It takes an R expression, an R one line function or a yacas string and returns an R expression or a variety of other formats. It can be used for symbolic mathematics, exact arithmetic, ASCII pretty printing and R to TeX conversions. The main command is `yacas` and `?yacas` provides some information on installation and startup.

Details

The following are sources of information on "Ryacas":

DESCRIPTION file	<code>library(help = Ryacas)</code>
List of demo files	<code>demo(package = "Ryacas")</code>
Demo file	<code>demo("Ryacas")</code>
Demo	<code>demo("Ryacas-PrettyPrinter")</code>
Demo	<code>demo("Ryacas-Function")</code>
Demo	<code>demo("Ryacas-Sym")</code>
Demo	<code>demo("Ryacas-Expr")</code>
List Vignettes	<code>vignette(package = "Ryacas")</code>
Vignette	<code>vignette("Ryacas")</code>
This File	<code>package?Ryacas</code>
Help files	<code>?yacas, ?yacasTranslations, ?yacmode, ?Sym</code>
Help files - Windows	<code>?yacasInstall</code>
News	<code>RShowDoc("NEWS", package = "Ryacas")</code>
Acknowledgements	<code>RShowDoc("THANKS", package = "Ryacas")</code>
Wish List	<code>RShowDoc("WISHLIST", package = "Ryacas")</code>
Home page	http://code.google.com/p/ryacas/

Note

There is a note in the help file of the [yac](#) command that discusses a number of installation and startup issues.

Examples

```
print(yacas(expression(integrate(1/x, x))))
print(yacas("Integrate(x)1/x"))
x <- Sym("x"); Integrate(1/x, x)
acos(Sym("1/2"))
```

as.Sym.character	<i>Convert character vector to yacas object</i>
------------------	---

Description

Simple and raw conversion to yacas

Usage

```
## S3 method for class 'character'
as.Sym(x, ...)
```

Arguments

x	An R character vector.
...	Not used

Examples

```
x <- c("a", "2", "4", "c", "d", "6")
x
y <- as.Sym(x)
y
Eval(y, list(a = 3, c = 3, d = 3))
```

as.Sym.matrix	<i>Convert character matrix to yacas object</i>
---------------	---

Description

Simple and raw conversion to yacas

Usage

```
## S3 method for class 'matrix'
as.Sym(x, ...)
```

Arguments

x	An R character matrix.
...	Not used

Examples

```
x <- matrix(c("a", "2", "4", "c", "d", "6"), 3, 2)
x
y <- as.Sym(x)
y
Eval(y, list(a = 3, c = 3, d = 3))
```

bodyAsExpression	<i>Get body of function as an expression.</i>
------------------	---

Description

Get body of function as an expression.

Usage

```
bodyAsExpression(x)
```

Arguments

x	An R function.
---	----------------

Details

This function is similar to the R body function except that function returns a call object whereas this one returns an expression usable in Ryacas calculations.

Value

An expression.

See Also

[body](#)

Examples

```
# construct an R function for the Burr probability density
# function (PDF) given the Burr cumulative distribution function (CDF)
BurrCDF <- function(x, c = 1, k = 1) 1-(1+x^c)^-k

# transfer CDF to yacas
yacas(BurrCDF)

# create a template for the PDF from the CDF
BurrPDF <- BurrCDF

# differentiate CDF and place resulting expression in body
body(BurrPDF) <- yacas(expression(deriv(BurrCDF(x,c,k))))[[1]]

# test out PDF
BurrPDF(1)
```

Eval

Evaluate a yacas expression.

Description

Evaluate a yacas expression.

Usage

```
Eval(x, env = parent.frame(), ...)
```

Arguments

x	Object to be evaluated.
env	Environment or list in which to perform evaluation.
...	Not currently used.

Examples

```

Eval(yacas(expression(x*x)), list(x=2))

# same
x <- 2
Eval(yacas(expression(x*x)))

```

get_output_width	<i>Get width of yacas output</i>
------------------	----------------------------------

Description

Get width of yacas output

Usage

```
get_output_width()
```

Ryacas_options	<i>Set or get options for the Ryacas package</i>
----------------	--

Description

Set or get options for the Ryacas package

Usage

```
Ryacas_options(...)
```

Arguments

... Option names to retrieve option values or [key] = [value] pairs to set options.

Supported options

The following options are supported

- `module_matvec_enabled` (default TRUE): Print yacas `List()`'s as vectors and `List(List(), ...)`'s as matrices.
- `prettyform_default` (default FALSE): Print yacas as `PrettyForm()` as default.

set_output_width	<i>Set width of yacas output</i>
------------------	----------------------------------

Description

Set width of yacas output

Usage

```
set_output_width(w)
```

Arguments

w	Width in number of characters
---	-------------------------------

yacas	<i>yacas interface – silent version</i>
-------	---

Description

Similar to [yacas\(\)](#) but silent. This can be useful when working with yacas directly.

Usage

```
syacas(x, ...)
```

Arguments

x	A yacas character string or an R expression without terminating semicolon to be processed by yacas.
...	Additional arguments ultimately passed down to yacas.character.

See Also

yacas

Sym

*Sym***Description**

The Symbol interface to yacas.

Usage

```
Sym(...)  
Expr(x)
```

Arguments

...	An R character string or object that can be coerced to a character string.
x	An R expression.

Details

An object of class "Sym" is internally a yacas character string. An object of class "Expr" is internally an R expression. One can combine such objects using the Math and Ops R operators (see `help(Math)` and `help(Ops)` for a list). Also there are methods for a number of R generics: `as.character.Sym`, `as.expression.Sym`, `determinant.Sym`, `deriv.Sym` and `print.Sym` and yacas-oriented functions: `Clear`, `Conjugate`, `Expand`, `Factor`, `Factorial`, `I`, `Identity`, `Infinity`, `Integrate`, `Inverse`, `InverseTaylor`, `Limit`, `List`, `N`, `Newton`, `Pi`, `Precision`, `PrettyForm`, `PrettyPrinter`, `Set`, `Simplify`, `Solve`, `Subst`, `Taylor`, `TeXForm`, `Transpose`, `Ver` and " commands. Try `vignette("Rycas-Sym")` for many examples.

Value

`Sym` returns a "Sym" object and `Expr` returns an "Expr" object.

Note

Currently the only `Expr` methods implemented are `as.character.Expr`, `deriv.Expr`, `Math.Expr`, `Ops.Expr` and `print.Expr`.

See Also

[as.Sym.matrix\(\)](#)

Examples

```
x <- Sym("x")  
x*x  
Integrate(x*x, x)  
Sym("%") %Where% list(x = 10)
```



```
acos(Sym("1/2"))

y <- Exprq(x)
y*y
deriv(y*y, y)
Exprq(acos(1/2))
```

yacas

yacas interface

Description

Interface to the yacas computer algebra system.

Usage

```
yacas(x, ...)
## S3 method for class 'character'
yacas(x, verbose = FALSE, method,
      retclass = c("expression", "character", "unquote"),
      addSemi = TRUE, ...)
```

Arguments

x	A yacas character string or an R expression without terminating semicolon to be processed by yacas.
...	Additional arguments ultimately passed down to <code>yacas.character</code> .
verbose	A logical value indicating verbosity of output or "input" to only show input to yacas but not output from yacas or "output" to only show output from yacas but not input to yacas.
method	method used to communicate with yacas. If "socket" is specified then the same yacas session is used on a sequence of calls. If "system" is specified then a new instance of yacas is used just for the period of that call. "system" does not require that the system be configured to support telnet/sockets and so may be useful in some instances. If no value is specified the default is taken from <code>getOption("yacac.method")</code> and if that is not specified "socket" is used. "socket" and "system" may be abbreviated.
addSemi	If TRUE a semicolon is added to the character string sent to yacas. This can be set to FALSE if it is known that the character string already has a trailing semicolon. It is ignored if <code>retclass="expression"</code> .
retclass	The class of the first component of the yacas structure. It defaults to "expression" but may be specified as "character" or "unquote". "unquote" is the same as "character" except that if the character string returned would have otherwise had quotes in the first and last positions then they are stripped.

Details

The user supplies an R expression, an R function name corresponding to a function with a single line body, a formula or a yacas input string. In the case of a formula it is regarded as an expression represented by the right hand side of the formula while the left hand side, if any, is ignored.

Note the silent version `syacas()`.

Value

An R object of class "yacas" is returned. If `PrettyPrinter("OMForm")` is in effect, which it is by default, then the first component is an R expression and the `OMForm` component contains Open-Math XML code. In other cases the first component is `NULL` and the `YacasForm` or `PrettyForm` components have display information.

Note

Windows Installation. On Windows one can install Ryacas by issuing the commands:

```
install.packages("Ryacas", dep = TRUE)
library(Ryacas)
yacasInstall()
```

or by using the `Packages | Install package(s)` menu in place of the first command. The second command downloads `scripts.dat` and `yacas.exe` from the internet and installs them into `R_HOME/library/Ryacas/yacdir` where `R_HOME` is the location of your R installation.

Normally the default locations of yacas, its initialization file and the scripts file are sufficient but, if necessary, they can be overridden via the environment variables: `YACAS_HOME`, `YACAS_INIT` and `YACAS_SCRIPTS`. The `YACAS_INVOKE_STRING` environment variable discussed in the next section overrides all three of these.

All OS Installation. The `YACAS_INVOKE_STRING` environment variable can be used to override the invocation string for yacas. Normally it is not used. If it does need to be used then a typical use might be:

```
library(Ryacas)
# only need to do the file.copy command once
file.copy(system.file("yacdir/R.js", package = "Ryacas"), "~/yacsrc")
# this needs to be done once per session
Sys.setenv(YACAS_INVOKE_STRING = "yacas -pc --server 9734")
demo(Ryacas) # test it out
```

`yacmode`. There is also a utility `yacmode` which is called without arguments and just turns R into a terminal into yacas until one quits out of it (and back to R) by entering `stop`, `end`, `quit`, `exit` or `e`.

Startup. `yacas` starts up when `yacasStart()` is called or the first time `yacas` is called. `yacas` is shut down when `yacasStop()` is called or when the package is detached using the `detach()` R command. On Windows, when `yacas` is shut down, the `yacas` process is terminated on Windows XP

Pro but not on other versions of Windows. In those cases there will be a dangling process that the user must terminate manually.

Translation. The translation process occurs in several steps. If the input to the yacas function is an expression then it is translated to a valid yacas character string (otherwise, it is sent to yacas unprocessed). Yacas then processes the string and if `retclass="expression"` it is translated back to an R expression (otherwise it is sent back unprocessed). Examples of translations are:

R	yacas
<code>sin(x)</code>	<code>Sin(x)</code>
<code>deriv(sin, x)</code>	<code>Deriv(x)Sin(x)</code>
<code>log(x)</code>	<code>Ln(x)</code>

References

<http://yacas.sourceforge.net>

Examples

```
yacas(expression(Factor(x^2-1)))
exp1 <- expression(x^2 + 2 * x^2)
exp2 <- expression(2 * exp0)
exp3 <- expression(6 * pi * x)
exp4 <- expression((exp1 * (1 - sin(exp3))) / exp2)
print(yacas(exp4))

print(yacas("Version()")) # yacas version

# see demo("Ryacac-Function")
```

yacasTranslations *Yacas translations*

Description

Translations from R to the yacas computer algebra system.

Note

The translation process occurs in several steps. If the input to the yacas function is an expression then it is translated to a valid yacas character string (otherwise, it is sent to yacas unprocessed). Yacas then processes the string and if `retclass="expression"` it is translated back to an R expression (otherwise it is sent back unprocessed). Currently supported translations are:

CONSTANTS	
R	yacas

=	=====
pi	Pi

OPERATORS

R	yacas
=	=====
7 %% 3	Mod(7, 3)
7 %!% 3	Div(7, 3)

FUNCTIONS

R	yacas
=	=====
sin(x)	Sin(x)
cos(x)	Cos(x)
tan(x)	Tan(x)
asin(x)	ArcSin(x)
acos(x)	ArcCos(x)
atan(x)	ArcTan(x)
exp(x)	Exp(x)
sqrt(x)	Sqrt(x)
log(x)	Ln(x)
choose(n, k)	Bin(n, k)
gamma(x)	Gamma(x)
deriv(sin, x)	Deriv(x)Sin(x)
integrate(f, a, b)	Integrate(x, a, b)f(x)
list()	List()
factorial(n)	n!

Note the Limit example in demo(Ryacas) for adding translations on the fly.

The complete table under development.

Author(s)

Rob J Goedman

References

<http://yacass.sourceforge.net>

yacacs_evaluate	<i>Evaluate yacacs expression</i>
-----------------	-----------------------------------

Description

This is a low-level function for evaluating yacacs expression represented as string.

Usage

```
yacacs_evaluate(expr)
```

Arguments

expr	Yacacs expression
------	-------------------

Value

Result of evaluating expr by yacacs in OpenMath format and side-effects of the evaluation

Examples

```
yacacs_evaluate("D(x)Sin(x^2)")
```

yacmode	<i>yacmode interface</i>
---------	--------------------------

Description

Interactive interface to the yacacs

Usage

```
yacmode()
```

Details

The user types valid yacacs input and presses return. Type 'quit' to return to R prompt.

Value

Output of yacacs is returned.

Note

Note that command recall will recall previous R commands, not previous yacas input. Yacas is given a limited amount of time to complete, otherwise `\[1\]` `CommandLine(1)` : User interrupted calculation is returned. E.g. `Taylor(x,0,5) 1/(1+x)` will work, but `Taylor(x,0,12) 1/(1+x)` is likely to take too long.

References

<http://yacas.sourceforge.net>

Examples

```
## Not run:
yacmode()
(x+y)^3-(x-y)^3
Simplify(%)
q

## End(Not run)
```

Index

- *Topic **programming**
 - Ryacas-package, 2
- *Topic **symbolmath**
 - bodyAsExpression, 4
 - Eval, 5
 - Sym, 8
 - yacas, 9
 - yacasTranslations, 11
 - yacmode, 13
- %Where% (Sym), 8

- addSemi (yacas), 9
- as.character.Expr (Sym), 8
- as.character.Sym (Sym), 8
- as.character.yacas (yacas), 9
- as.Expr.formula (Sym), 8
- as.expression.Sym (Sym), 8
- as.expression.yacas (yacas), 9
- as.language (bodyAsExpression), 4
- as.Sym (Sym), 8
- as.Sym.character, 3
- as.Sym.matrix, 4
- as.Sym.matrix(), 8

- body, 5
- bodyAsExpression, 4

- CharacteristicEquation (Sym), 8
- Clear (Sym), 8
- Conjugate (Sym), 8

- deriv.Expr (Sym), 8
- deriv.Sym (Sym), 8
- determinant.Expr (Sym), 8
- determinant.Sym (Sym), 8

- EigenValues (Sym), 8
- Eval, 5
- Expand (Sym), 8
- Expr (Sym), 8
- Exprq (Sym), 8

- Factor (Sym), 8
- Factorial (Sym), 8
- FindRoots (Sym), 8

- get_output_width, 6

- I (Sym), 8
- Identity (Sym), 8
- Infinity (Sym), 8
- Integrate (Sym), 8
- Inverse (Sym), 8
- InverseTaylor (Sym), 8

- Limit (Sym), 8
- List (Sym), 8

- Math.Expr (Sym), 8
- Math.Sym (Sym), 8

- N (Sym), 8
- Newton (Sym), 8

- OpenMath2R (Sym), 8
- Ops.Expr (Sym), 8
- Ops.Sym (Sym), 8
- Ops.yacas.symbol (Sym), 8

- Pi (Sym), 8
- Precision (Sym), 8
- PrettyForm (Sym), 8
- PrettyPrinter (Sym), 8
- print.Expr (Sym), 8
- print.Sym (Sym), 8
- print.yacas (Sym), 8

- root (yacasTranslations), 11
- Ryacas-package, 2
- Ryacas_options, 6

- Set (Sym), 8
- set_output_width, 7

Simplify (Sym), 8
Solve (Sym), 8
Subst (Sym), 8
syacas, 7
syacas(), 10
Sym, 8
SymExpr (Sym), 8

Taylor (Sym), 8
TeXForm (Sym), 8
trans (Sym), 8
Transpose (Sym), 8
transtab (Sym), 8

Ver (Sym), 8

yacas, 3, 9
yacas(), 7
yacas.symbol.value (Sym), 8
yacas_evaluate, 13
yacasTranslations, 11
yacmode, 13
yAssignFunction (yacas), 9
yDeriv (Sym), 8
yFactorial (Sym), 8
yIntegrate (Sym), 8
yLimit (Sym), 8
ynext (yacas), 9
yparse (yacas), 9
yrewrite (Sym), 8
ySequence (yacas), 9
ysub (yacas), 9
yUnlist (Sym), 8