

Package ‘gmt’

July 22, 2025

Version 2.0.3

Date 2022-06-06

Title Interface Between GMT Map-Making Software and R

Imports utils

SystemRequirements gmt

LazyData yes

Description Interface between the GMT map-making software and R, enabling the user to manipulate geographic data within R and call GMT commands to draw and annotate maps in postscript format. The gmt package is about interactive data analysis, rapidly visualizing subsets and summaries of geographic data, while performing statistical analysis in the R console.

License GPL-3

URL <https://www.generic-mapping-tools.org>

NeedsCompilation no

Author Arni Magnusson [aut, cre]

Maintainer Arni Magnusson <thisisarni@gmail.com>

Repository CRAN

Date/Publication 2022-06-06 20:10:08 UTC

Contents

gmt-package	2
deg2num	3
geodist	4
gmt	5
gmt.demo	6
gmt.system	7
num2deg	8
psbar	9
psclose	11
pscoast	12

pstext	13
psxy	14
r2gmt	15

Index	17
--------------	-----------

gmt-package	<i>Interface between the GMT Map-Making Software and R</i>
-------------	--

Description

Interface between the GMT map-making software and R, enabling the user to manipulate geographic data within R and call GMT commands to draw and annotate maps in postscript format.

Details

Initialize GMT session:

gmt set graphical parameters and postscript file

Create a map:

pscoast	draw
psxy	annotate
pstext	annotate
psbar	annotate
psclose	finalize

Convert and calculate:

deg2num	degrees to numeric
geodist	distance between coordinates
num2deg	numeric to degrees

Examples:

demo.par, demo.coast, demo.xy, demo.text, demo.bar strings and data frames to create example map

Internal:

gmt.system, r2gmt helping functions

This package provides functions to draw basic maps with GMT, along with helping functions that can be used to add more advanced features to a map.

GMT users typically write shell scripts to draw maps. The **gmt** package is about interactive data analysis, rapidly visualizing subsets and summaries of geographic data, while performing statistical analysis in the R console.

Author(s)

Arni Magnusson.

References

Wessel, P., Smith, W. H. F., Scharroo, R., Luis, J. and Wobbe, F. *The Generic Mapping Tools: GMT documentation*. Available at <https://docs.generic-mapping-tools.org>.

deg2num

Convert Degrees to Numeric

Description

Convert deg:min:sec string to decimal number.

Usage

deg2num(x)

Arguments

x string or vector of strings.

Details

Degrees, minutes and seconds are separated by colons, and each can have a decimal point as well. First character must be a minus sign or number, and last character must be W, E, S, N, or number.

Value

Numeric representation of the degree string(s).

Note

The string format is adopted from *Appendix B.1.1* in the GMT manual.

See Also

[as.numeric](#) converts strings to numbers when things are straightforward.

deg2num is the opposite of [num2deg](#).

[gmt-package](#) gives an overview of the package.

Examples

```
# The opposite of num2deg() example
deg2num(c("12:30:44.5W", "17.5S", "1:00:05", "200:45E"))
```

geodist

*Distance Between Geographic Coordinates***Description**

Calculate surface distance between geographic coordinates.

Usage

```
geodist(Nfrom, Efrom, Nto, Eto, units="km")
```

Arguments

Nfrom	latitude of origin.
Efrom	longitude of origin.
Nto	latitude of destination.
Eto	longitude of destination.
units	how distance is measured: "km" for kilometres, "nm" for nautical miles.

Details

Latitude and longitude are passed as decimal numbers, e.g. 66.5 for 66°30'N. Vectors of coordinates are supported.

Value

Vector of distances.

Note

The surface distance between geographic coordinates is:

$$D = \cos^{-1}[\sin \theta_1 \sin \theta_2 + \cos \theta_1 \cos \theta_2 \cos(\phi_1 - \phi_2)]$$

where distance and coordinates are expressed in radians. θ_1 and θ_2 is the latitude of origin and destination, and ϕ_1 and ϕ_2 is longitude.

The calculations assume a perfect sphere and elevation differences are ignored. The SI definition of a nautical mile is exactly 1.852 km.

See Also

[diff](#), [Trig](#).

[gmt-package](#) gives an overview of the package.

Examples

```
geodist(55.75,37.63, 39.9,116.4) # Moscow - Beijing
geodist(90,0, -90,0, "nm")      # from pole to pole
```

`gmt`*Initialize GMT Session*

Description

Initialize a GMT session by setting graphical parameters and current postscript file.

Usage

```
gmt(par=NULL, file="map.ps", style="s", quiet=TRUE)
```

Arguments

<code>par</code>	optional string of GMT parameters, "PARAMETER1=value1 PARAMETER2=value2 ...", passed to GMT shell command <code>gmtset</code> .
<code>file</code>	postscript filename used in subsequent calls to GMT plot functions, passed to options.
<code>style</code>	default units and values: "s" for SI (international) or "u" for United States, passed to GMT shell command <code>gmtset</code> .
<code>quiet</code>	whether current settings should be displayed.

Details

The file argument can be supplied with (recommended) or without a full directory path. Without a path, the R working directory is used (see `getwd` and `setwd`).

See the GMT documentation for details on graphical parameters, `gmtdefaults`, `gmtset` and other GMT commands.

Value

List containing the current `options("gmt.file")`.

If `par` is `NULL`, no GMT parameters are changed, but the current parameter values and postscript filename can be reviewed.

If `par` is a string (empty "" will do), a GMT configuration file is created in the current map directory, `dirname(file)`.

See Also

[options](#) could be used to set `gmt.file` directly.

`gmt`, [pscoast](#), [psxy](#), [pstext](#), [psbar](#), and [psclose](#) work together to draw maps.

[gmt-package](#) gives an overview of the package.

Examples

```
## Not run:
# Draw map and save as "map.ps" in R working directory
gmt(demo.par)
pscoast(demo.coast)
psxy(demo.xy)
pstext(demo.text, "-J -R -F+f+a+j -O -K")
psbar(demo.bar, ref=66)
psclose()
# See directory gmt/example for details

## End(Not run)
```

gmt.demo

GMT Example Data

Description

These five objects are provided to demonstrate the functionality of the **gmt** package.

Usage

```
demo.par
demo.coast
demo.xy
demo.text
demo.bar
```

Format

demo.par and demo.coast are simple strings.

demo.xy is a data frame containing:

Lon	Longitude
Lat	Latitude
Size	Size of plot symbol

demo.text is a data frame containing:

Lon	Longitude
Lat	Latitude
Size	Size of plot symbol
Angle	Angle in degrees counter-clockwise from horizontal
Font	Font number
Justify	Justification code
Text	Text label

demo.bar is a data frame containing:

Lon	Longitude
Lat	Longitude
Width	Bar width in degrees
Height	Bar height in degrees

Details

See the GMT documentation for details on psxy, pstext and other GMT commands.

See Also

[gmt-package](#) gives an overview of the package.

Examples

```
## Not run:
# Draw map and save as "map.ps" in R working directory
gmt(demo.par)
pscoast(demo.coast)
psxy(demo.xy)
pstext(demo.text, "-J -R -F+f+a+j -O -K")
psbar(demo.bar, ref=66)
psclose()
# See directory gmt/example for details

## End(Not run)
```

gmt.system

Invoke shell command

Description

This internal function invokes a shell command, possibly directing the output to a file.

Usage

```
gmt.system(cmd, file=NULL, append=FALSE)
```

Arguments

cmd	system command to be invoked, as a string.
file	filename to which output should be directed.
append	whether output should be appended to existing file.

Value

Command output as a vector of strings.

Note

`gmt.system` is a fast platform-independent wrapper for `system`, supporting redirection to file.

It is mainly called by other functions, but users may find it useful for running various GMT commands.

See Also

[system](#), [writelines](#).

[gmt-package](#) gives an overview of the package.

Examples

```
## Not run:
# Assuming bermuda.nc is in R working directory
gmt.system("gmt grdcontour bermuda.nc -JM7i -C250 -A1000 -B2", file="b.ps")

## End(Not run)
```

num2deg	<i>Convert Numeric to Degrees</i>
---------	-----------------------------------

Description

Convert decimal number to deg:min:sec string.

Usage

```
num2deg(x, lat=NA, dec=FALSE, digits=0, zero=FALSE)
```

Arguments

<code>x</code>	number or vector of numbers.
<code>lat</code>	whether <code>x</code> is latitude.
<code>dec</code>	whether to return decimal degrees instead of deg:min:sec.
<code>digits</code>	precision used when rounding decimal degrees or seconds.
<code>zero</code>	whether trailing :00 zeros should be retained.

Details

Element-specific format is supported, using vectors for `lat`, `digits`, and `precision`.

The resulting string ends with N or S when `lat` is TRUE, E or W when `lat` is FALSE, or a number when `lat` is NA.

Value

deg:min:sec string representation of the number(s).

Note

The string format is adopted from *Appendix B.1.1* in the GMT manual.

See Also

[as.character](#) converts plain numbers to strings.

num2deg is the opposite of [deg2num](#).

[gmt-package](#) gives an overview of the package.

Examples

```
# The opposite of deg2num() example
num2deg(c(-12.51236, -17.5, 1.00139, 200.75),
        lat=c(FALSE, TRUE, NA, FALSE),
        dec=c(FALSE, TRUE, FALSE, FALSE),
        digits=c(1, 1, 0, 0))
```

psbar

Add Bars to GMT Mercator Map

Description

Call GMT to add bars to a map and save in postscript format.

Usage

```
psbar(x, cmd="-J -R -W1p -G180 -O -K", file=getOption("gmt.file"),
      ref=0, digits=getOption("digits"))
```

Arguments

x	data frame, matrix, or filename containing the data to be plotted.
cmd	string of arguments passed to GMT shell command psxy.
file	filename where the map is saved.
ref	reference latitude where height 1 renders a bar 1 degree high.
digits	precision used when rounding the geographic coordinates.

Details

The data are arranged in four columns: Lon, Lat, Width, and Height, in that order.

If `x` is a filename, the data should be tabular with or without a header, separated by commas or whitespace. The first line is interpreted as header if the first non-whitespace character is not minus, point, or number.

This function provides an alternative to `psxy -Sb` and `psxy -Sr` for drawing bars on a Mercator map. See the GMT documentation for details on `psxy` and other GMT commands.

Value

Null, but the map is annotated and saved in postscript format.

The temporary GMT input file `'bar.gmt'` is saved in directory `dirname(tempdir())`, for the user to view or edit. It is later removed by `psclose()`.

Note

This function does the necessary calculations to draw bars in standard height given a Mercator-projected map. It is not intended for other projections.

The derivative of the Mercator projection is used to standardize the bar height:

$$f'(\theta) = \frac{1}{2 \tan\left(\frac{\pi}{4} + \frac{\theta}{2}\right) \cos^2\left(\frac{\pi}{4} + \frac{\theta}{2}\right)}$$

where θ is latitude in radians and $f(\theta)$ is the y-axis coordinate. The bar height at a given latitude is $h \times f'(\theta_{\text{ref}})/f'(\theta)$, where h is the height passed by the user and θ_{ref} is a reference latitude where $h=1$ renders a bar 1 degree high.

See Also

Similar to [barplot](#) and [postscript](#) in native R graphics.

[gmt](#), [pscoast](#), [psxy](#), [pstext](#), `psbar`, and [psclose](#) work together to draw maps.

[gmt-package](#) gives an overview of the package.

Examples

```
## Not run:
# Draw map and save as "map.ps" in R working directory
gmt(demo.par)
pscoast(demo.coast)
psxy(demo.xy)
pstext(demo.text, "-J -R -F+f+a+j -O -K")
psbar(demo.bar, ref=66)
psclose()
# See directory gmt/example for details

## End(Not run)
```

psclose

Finalize GMT Map

Description

Call GMT to finalize a map and save in postscript format.

Usage

```
psclose(file=getOption("gmt.file"), trailer=TRUE)
```

Arguments

file	filename where the map is saved.
trailer	whether a closing trailer should be appended to the postscript file.

Details

A closing trailer is required if the last plotting command included `-K` (default behaviour).

Value

NULL, but the map is finalized and saved in postscript format.

Note

This function performs two tasks:

1. Appends a closing trailer to the postscript file (optional).
2. Removes GMT files in temporary directory.

See Also

Analogous to `dev.off` and `postscript` in native R graphics.
[gmt](#), [pscoast](#), [psxy](#), [pstext](#), [psbar](#), and `psclose` work together to draw maps.
[gmt-package](#) gives an overview of the package.

Examples

```
## Not run:  
# Draw map and save as "map.ps" in R working directory  
gmt(demo.par)  
pscoast(demo.coast)  
psxy(demo.xy)  
pstext(demo.text, "-J -R -F+f+a+j -O -K")  
psbar(demo.bar, ref=66)  
psclose()  
# See directory gmt/example for details
```

```
## End(Not run)
```

pscoast

Draw GMT Map

Description

Call GMT to draw a map (coastlines, borders, rivers) and save in postscript format.

Usage

```
pscoast(cmd, file=getOption("gmt.file"))
```

Arguments

cmd	string of arguments passed to GMT shell command pscoast.
file	filename where the map will be saved.

Details

The file argument can be supplied with (recommended) or without a full directory path. Without a path, the current working directory is used (see `getwd` and `setwd`).

See the GMT documentation for details on pscoast and other GMT commands.

Value

NULL, but a map is drawn and saved in postscript format.

See Also

Similar to [plot](#) and [postscript](#) in native R graphics.

[gmt](#), [pscoast](#), [psxy](#), [pstext](#), [psbar](#), and [psclose](#) work together to draw maps.

[gmt-package](#) gives an overview of the package.

Examples

```
## Not run:
# Draw map and save as "map.ps" in current working directory
gmt(demo.par)
pscoast(demo.coast)
psxy(demo.xy)
pstext(demo.text, "-J -R -F+f+a+j -O -K")
psbar(demo.bar, ref=66)
psclose()
# See directory gmt/example for details

# Map in one call
```

```
pscoast("-JM12c -R7E/38E/29N/48N -G100 -B5", "x.ps")

## End(Not run)
```

pstext

Add Text/Symbols to GMT Map

Description

Call GMT to add text/symbols to a map and save in postscript format.

Usage

```
pstext(x, cmd="-J -R -O -K", file=getOption("gmt.file"))
```

Arguments

x	data frame, matrix, or filename containing the data to be plotted.
cmd	string of arguments passed to GMT shell command pstext.
file	filename where the map is saved.

Details

The cmd argument can be used to specify the data columns:

Format (cmd)	Data columns	Notes
-J -R -O -K	Lon Lat Text	(default)
-J -R -F+f+a+j -O -K	Lon Lat Font Angle Justify Text	(see example)

If x is a filename, the data should be tabular with or without a header, separated by commas or whitespace. The first line is interpreted as header if the first non-whitespace character is not minus, point, or number.

See the GMT documentation for details on pstext and other GMT commands.

Value

NULL, but the map is annotated and saved in postscript format.

The temporary GMT input file 'text.gmt' is saved in directory `dirname(tempdir())`, for the user to view or edit. It is later removed by `psclose()`.

See Also

Similar to [text](#) and [postscript](#) in native R graphics.

[gmt](#), [pscoast](#), [psxy](#), [pstext](#), [psbar](#), and [psclose](#) work together to draw maps.

[gmt-package](#) gives an overview of the package.

Examples

```
## Not run:
# Draw map and save as "map.ps" in current working directory
gmt(demo.par)
pscoast(demo.coast)
psxy(demo.xy)
pstext(demo.text, "-J -R -F+f+a+j -O -K")
psbar(demo.bar, ref=66)
psclose()
# See directory gmt/example for details

## End(Not run)
```

psxy

Add Lines/Symbols to GMT Map

Description

Call GMT to add lines/symbols to a map and save in postscript format.

Usage

```
psxy(x, cmd="-J -R -Scp -W2p -O -K", file=getOption("gmt.file"))
```

Arguments

x	data frame, matrix, or filename containing the data to be plotted.
cmd	string of arguments passed to GMT shell command psxy.
file	filename where the map is saved.

Details

The data are arranged in two (Lon, Lat) or more columns, depending on the `-S` argument.

If `x` is a filename, the data should be tabular with or without a header, separated by commas or whitespace. The first line is interpreted as header if the first non-whitespace character is not minus, point, or number.

See the GMT documentation for details on psxy and other GMT commands.

Value

NULL, but the map is annotated and saved in postscript format.

The temporary GMT input file 'xy.gmt' is saved in directory `dirname(tempdir())`, for the user to view or edit. It is later removed by `psclose()`.

See Also

Similar to [points](#), [lines](#), and [postscript](#) in native R graphics.
[gmt](#), [pscoast](#), [psxy](#), [pstext](#), [psbar](#), and [psclose](#) work together to draw maps.
[gmt-package](#) gives an overview of the package.

Examples

```
## Not run:
# Draw map and save as "map.ps" in current working directory
gmt(demo.par)
pscoast(demo.coast)
psxy(demo.xy)
pstext(demo.text, "-J -R -F+f+a+j -O -K")
psbar(demo.bar, ref=66)
psclose()
# See directory gmt/example for details

# Simple map and circles
pscoast("-JM12c -R158/192/-42/-8 -Di -Gdarkgreen -B10f5 -A2000 -K", "quake.ps")
psxy(data.frame(lon=quakes$lon, lat=quakes$lat, mag=10^quakes$mag/1e6),
      "-J -R -Scp -W0.5p,red -O", "quake.ps")

## End(Not run)
```

r2gmt

Prepare Data for GMT

Description

This internal function reads data, from a filename or R object, and writes them to a GMT input file.

Usage

```
r2gmt(x, datafile, append=FALSE)
```

Arguments

x	data frame, matrix, or filename containing the data to be written to a temporary file.
datafile	filename where the data will be written in GMT format, tab-separated without header.
append	whether data should be appended to an existing file, separating segments with ">" lines.

Details

If x is a filename, the data should be tabular with or without a header, separated by commas or whitespace. The first line is interpreted as header if the first non-whitespace character is not minus, point, or number.

Value

The data frame that was written to datafile.

Note

r2gmt is like `write.table`, except it allows `x` to be a filename, and appends tables with the GMT > separator.

It is mainly called by other functions, but users may find it useful for writing input data for GMT commands.

See Also

[scan](#), [read.table](#), [write](#), [write.table](#).

[gmt-package](#) gives an overview of the package.

Examples

```
LonLat1 <- data.frame(Lon=1:3, Lat=4:6)
LonLat2 <- data.frame(Lon=7:8, Lat=9:10)
## Not run:
r2gmt(LonLat1, "temp.gmt")
r2gmt(LonLat2, "temp.gmt", append=TRUE)

## End(Not run)
```

Index

as.character, [9](#)
as.numeric, [3](#)

barplot, [10](#)

deg2num, [2](#), [3](#), [9](#)
demo.bar, [2](#)
demo.bar (gmt.demo), [6](#)
demo.coast, [2](#)
demo.coast (gmt.demo), [6](#)
demo.par, [2](#)
demo.par (gmt.demo), [6](#)
demo.text, [2](#)
demo.text (gmt.demo), [6](#)
demo.xy, [2](#)
demo.xy (gmt.demo), [6](#)
dev.off, [11](#)
diff, [4](#)

geodist, [2](#), [4](#)
gmt, [2](#), [5](#), [10–13](#), [15](#)
gmt-package, [2](#)
gmt.demo, [6](#)
gmt.system, [2](#), [7](#)

lines, [15](#)

num2deg, [2](#), [3](#), [8](#)

options, [5](#)

plot, [12](#)
points, [15](#)
postscript, [10–13](#), [15](#)
psbar, [2](#), [5](#), [9](#), [11–13](#), [15](#)
psclose, [2](#), [5](#), [10](#), [11](#), [12](#), [13](#), [15](#)
pscoast, [2](#), [5](#), [10](#), [11](#), [12](#), [13](#), [15](#)
pstext, [2](#), [5](#), [10–12](#), [13](#), [15](#)
psxy, [2](#), [5](#), [10–13](#), [14](#)

read.table, [16](#)

scan, [16](#)
system, [8](#)

text, [13](#)
Trig, [4](#)

write, [16](#)
write.table, [16](#)
writeLines, [8](#)

r2gmt, [2](#), [15](#)