Package: Rsagacmd (via r-universe)

January 7, 2025

Type Package

Title Linking R with the Open-Source 'SAGA-GIS' Software

Version 0.4.3

Date 2024-09-07

Maintainer Steven Pawley <dr.stevenpawley@gmail.com>

Description Provides an R scripting interface to the open-source 'SAGA-GIS' (System for Automated Geoscientific Analyses Geographical Information System) software. 'Rsagacmd' dynamically generates R functions for every 'SAGA-GIS' geoprocessing tool based on the user's currently installed 'SAGA-GIS' version. These functions are contained within an S3 object and are accessed as a named list of libraries and tools. This structure facilitates an easier scripting experience by organizing the large number of 'SAGA-GIS' geoprocessing tools (>700) by their respective library. Interactive scripting can fully take advantage of code autocompletion tools (e.g. in 'RStudio'), allowing for each tools syntax to be quickly recognized. Furthermore, the most common types of spatial data (via the 'terra', 'sp', and 'sf' packages) along with non-spatial data are automatically passed from R to the 'SAGA-GIS' command line tool for geoprocessing operations, and the results are loaded as the appropriate R object. Outputs from individual 'SAGA-GIS' tools can also be chained using pipes from the 'magrittr' and 'dplyr' packages to combine complex geoprocessing operations together in a single statement. 'SAGA-GIS' is available under a GPLv2 / LGPLv2 licence from <https://sourceforge.net/projects/saga-gis/> including Windows x86/x64 and macOS binaries. SAGA-GIS is also included in Debian/Ubuntu default software repositories. Rsagacmd has currently been tested on 'SAGA-GIS' versions from 2.3.1 to 9.5.1 on Windows, Linux and macOS.

License GPL-3

Encoding UTF-8

SystemRequirements SAGA-GIS (>= 2.3.1)

RoxygenNote 7.3.2

Depends R (>= 2.10)

Imports generics, sf, terra (>= 1.7.0), stars, tools, utils, foreign, stringr, rlang, tibble, processx, rvest

Suggests dplyr, testthat (>= 3.0.0), covr

Config/testthat/edition 3

URL https://stevenpawley.github.io/Rsagacmd/

BugReports https://github.com/stevenpawley/Rsagacmd/issues

NeedsCompilation no

Author Steven Pawley [aut, cre]

Repository CRAN

Date/Publication 2024-09-08 04:20:01 UTC

Config/pak/sysreqs libgdal-dev gdal-bin libgeos-dev libicu-dev libxml2-dev libssl-dev libproj-dev saga libsqlite3-dev libudunits2-dev

Contents

mrvbf_threshold *Calculate the t_slope value based on DEM resolution for MRVBF*

Description

Calculates the t_slope value for the Multiresolution Index of Valley Bottom Flatness (Gallant and Dowling, 2003) based on input DEM resolution. MRVBF identified valley bottoms based on classifying slope angle and identifying low areas by ranking elevation in respect to the surrounding topography across a range of DEM resolutions. The MRVBF algorithm was developed using a 25 m DEM, and so if the input DEM has a different resolution then the slope threshold t_slope needs to be adjusted from its default value of 16 in order to maintain the relationship between slope and DEM resolution. This function provides a convenient way to perform that calculation.

Usage

```
mrvbf_threshold(res)
```

Arguments

res	numeric, DEM resolution

Value

numeric, t_slope value for MRVBF

Examples

```
mrvbf_threshold(res = 10)
```

print.saga_tool *Generic function to display help and usage information for any SAGA-GIS tool*

Description

Displays a tibble containing the name of the tool's parameters, the argument name used by Rsagacmd, the identifier used by the SAGA-GIS command line, and additional descriptions, default and options/constraints.

Usage

```
## S3 method for class 'saga_tool'
print(x, ...)
```

Arguments

x	A ‘saga_tool‘ object.
...	Additional arguments to pass to print. Currently not used.

Examples

```
## Not run:
# Initialize a saga object
saga <- saga_gis()

# Display usage information on a tool
print(saga$ta_morphometry$slope_aspect_curvature)

# Or simply:
saga$ta_morphometry$slope_aspect_curvature

## End(Not run)
```

---

read_srtm *Get path to the example DEM data*

Description

Rsagacmd comes bundled with a small tile of example Digital Elevation Model (DEM) data from the NASA Shuttle Radar Topography Mission Global 1 arc second V003. This data is stored in GeoTIFF format in ‘inst/extdata‘.

Usage

```
read_srtm()
```

Details

The dataset contains the land surface elevation of an area located near Jasper, Alberta, Canada, with the coordinate reference system (CRS) EPSG code of 3402 (NAD83(CSRS) / Alberta 10-TM (Forest)).

To access the data, use the convenience function of ‘read_srtm()‘ to load the data as a ‘terra::SpatRaster‘ object.

Examples

```
library(Rsagacmd)
library(terra)

dem <- read_srtm()
plot(dem)
```

saga_configure *Generates a custom saga_cmd configuration file*

Description

Creates and edits a saga_cmd configuration file in order to change saga_cmd settings related to file caching and number of available processor cores. Intended to be used internally by saga_gis

Usage

```
saga_configure(
  senv,
  grid_caching = FALSE,
  grid_cache_threshold = 100,
  grid_cache_dir = NULL,
  cores = NULL,
  saga_vers
)
```

Arguments

senv	A saga environment object. Contains the SAGA-GIS environment and settings.
grid_caching	Whether to use file caching. The default is FALSE.
grid_cache_threshold	Any number to use as a threshold (in Mb) before file caching for loaded raster data is activated.
grid_cache_dir	Optionally specify a path to the used directory for temporary files. The default uses 'base::tempdir'.
cores	An integer specifying the maximum number of processing cores. Needs to be set to 1 if file caching is activated because file caching in SAGA-GIS is not thread-safe.
saga_vers	A 'numeric_version' that specifies the version of SAGA-GIS. The generation of a saga_cmd configuration file is only valid for versions > 4.0.0.

Value

A character that specifies the path to custom saga_cmd initiation file.

saga_docs *Browse the online documentation for a saga_tool*

Description

Browse the online documentation for a saga_tool

Usage

```
saga_docs(saga_tool)
```

Arguments

saga_tool	a saga_tool object

Examples

```
## Not run:
library(Rsagacmd)

saga <- saga_gis()

saga_docs(saga$ta_morphometry$slope_aspect_curvature)

## End(Not run)
```

saga_env *Parses valid SAGA-GIS libraries and tools into a nested list of functions*

Description

Establishes the link to SAGA GIS by generating a SAGA help file and parsing all libraries, tools and options from the help files into a nested list of library, module and options, that are contained within an saga environment object object. Intended to be used internally by saga_gis

Usage

```
saga_env(
  saga_bin = NULL,
  opt_lib = NULL,
  raster_backend = "terra",
  vector_backend = "sf"
)
```

Arguments

saga_bin	An optional character vector to specify the path to the saga_cmd executable. Otherwise the function will perform a search for saga_cmd.
opt_lib	A character vector of a subset of SAGA-GIS tool libraries to generate dynamic functions that map to each tool. Used to save time if you only want to import a single library.
raster_backend	A character vector to specify the library to use for handling raster data. Currently, either "terra" or "stars" is supported. The default is "terra".
vector_backend	A character to specify the library to use for handling vector data. Currently, either "sf", "SpatVector" or "SpatVectorProxy" is supported. The default is "sf".

Value

A saga environment S3 object containing paths, settings and a nested list of libraries tools and options.

---

saga_execute *Function to execute SAGA-GIS commands through the command line tool*

Description

Intended to be used internally by each function

Usage

```
saga_execute(
  lib,
  tool,
  senv,
  .intern = NULL,
  .all_outputs = NULL,
  .verbose = NULL,
  ...
)
```

Arguments

lib	A character specifying the name of SAGA-GIS library to execute.
tool	A character specifying the name of SAGA-GIS tool to execute.
senv	A saga environment object.
.intern	A logical specifying whether to load the outputs from the SAGA-GIS geoprocessing operation as an R object.

.all_outputs	A logical to specify whether to automatically output all results from the selected SAGA tool and load them results as R objects (default = TRUE). If .all_outputs = FALSE then the file paths to store the tool's results will have to be manually specified in the arguments.
.verbose	Option to output all message during the execution of saga_cmd. Overrides the saga environment setting.
...	Named arguments and values for SAGA tool.

Value

output of SAGA-GIS tool loaded as an R object.

`saga_gis` *Initiate a SAGA-GIS geoprocessor object*

Description

Dynamically generates functions to all valid SAGA-GIS libraries and tools. These functions are stored within a saga S3 object as a named list of functions

Usage

```
saga_gis(
  saga_bin = NULL,
  grid_caching = FALSE,
  grid_cache_threshold = 100,
  grid_cache_dir = NULL,
  cores = NULL,
  raster_backend = "terra",
  vector_backend = "sf",
  raster_format = "SAGA",
  vector_format = c("ESRI Shapefile", "GeoPackage"),
  all_outputs = TRUE,
  intern = TRUE,
  opt_lib = NULL,
  temp_path = NULL,
  verbose = FALSE
)
```

Arguments

saga_bin	The path to saga_cmd executable. If this argument is not supplied then an automatic search for the saga_cmd executable will be performed.
grid_caching	A logical whether to use file caching in saga_cmd geoprocessing operations for rasters that are too large to fit into memory.

`grid_cache_threshold`
: A number to act as a threshold (in Mb) before file caching is activated for loaded raster data.

`grid_cache_dir`
: The path to directory for temporary files generated by file caching. If not provided then the result from ‘base::tempdir()‘ is used.

`cores`
: An integer for the maximum number of processing cores. By default all cores are utilized. Needs to be set to 1 if file caching is activated.

`raster_backend`
: A character vector to specify the library to use for handling raster data. Supported options are "terra" or "stars". The default is "terra".

`vector_backend`
: A character to specify the library to use for handling vector data. Currently, "sf", "SpatVector" or "SpatVectorProxy" is supported. The default is "sf", however for large vector datasets, using the "SpatVectorProxy" backend from the ‘terra‘ package has performance advantages because it allows file-based which can reduce repeated reading/writing when passing data between R and SAGA-GIS.

`raster_format`
: A character to specify the default format used to save raster data sets that are produced by SAGA-GIS. Available options are one of "SAGA", "SAGA Compressed" or "GeoTIFF". The default is "SAGA".

`vector_format`
: A character to specify the default format used for vector data sets that are produced by SAGA-GIS, and also used to save in-memory objects to be read by SAGA-GIS. Available options are of of "ESRI Shapefile", "GeoPackage", or "GeoJSON". The default is "ESRI Shapefile" for SAGA versions < 7.0 and GeoPackage for more recent versions. Attempting to use anything other than "ESRI Shapefile" for SAGA-GIS versions < 7.0 will raise an error.

`all_outputs`
: A logical to indicate whether to automatically use temporary files to store all output data sets from each SAGA-GIS tool. Default = TRUE. This argument can be overridden by the ‘.all_outputs‘ parameter on each individual SAGA-GIS tool function that is generated by ‘Rsagacmd::saga_gis()‘.

`intern`
: A logical to indicate whether to load the SAGA-GIS geoprocessing results as an R object, default = TRUE. For instance, if a raster grid is output by SAGA-GIS then this will be loaded as either as a ‘SpatRaster‘ or ‘stars‘ object, depending on the ‘raster_backend‘ setting that is used. Vector data sets are always loaded as ‘sf‘ objects, and tabular data sets are loaded as tibbles. The ‘intern‘ settings for the ‘saga‘ object can be overridden for individual tools using the ‘.intern‘ argument.

`opt_lib`
: A character vector with the names of a subset of SAGA-GIS libraries. Used to link only a subset of named SAGA-GIS tool libraries, rather than creating functions for all available tool libraries.

`temp_path`
: The path to use to store any temporary files that are generated as data is passed between R and SAGA-GIS. If not specified, then the system ‘base::tempdir()‘ is used.

`verbose`
: Logical to indicate whether to output all messages made during SAGA-GIS commands to the R console. Default = FALSE. This argument can be overriden by using the ‘.verbose‘ argument on each individual SAGA-GIS tool function that is generated by ‘Rsagacmd::saga_gis()‘.

Value

A S3 ‘saga‘ object containing a nested list of functions for SAGA-GIS libraries and tools.

Examples

```
## Not run:
# Initialize a saga object
library(Rsagacmd)
library(terra)

saga <- saga_gis()

# Alternatively initialize a saga object using file caching to handle large
# raster files
saga <- saga_gis(grid_caching = TRUE, grid_cache_threshold = 250, cores = 1)

# Example terrain analysis
# Generate a random DEM
dem <- saga$grid_calculus$random_terrain(radius = 100)

# Use Rsagacmd to calculate the Terrain Ruggedness Index
tri <- saga$ta_morphometry$terrain_ruggedness_index_tri(dem = dem)
plot(tri)

# Optionally run command and do not load result as an R object
saga$ta_morphometry$terrain_ruggedness_index_tri(dem = dem, .intern = FALSE)

# Initialize a saga object but do not automatically save all results to
# temporary files to load into R. Use this if you are explicitly saving each
# output because this will save disk space by not saving results from tools
# that output multiple results that you may be want to keep.
saga <- saga_gis(all_outputs = FALSE)

## End(Not run)
```

---

saga_remove_tmpfiles *Removes temporary files created by Rsagacmd*

Description

For convenience, functions in the Rsagacmd package create temporary files if any outputs for a SAGA-GIS tool are not specified as arguments. Temporary files in R are automatically removed at the end of each session. However, when dealing with raster data, these temporary files potentially can consume large amounts of disk space. These temporary files can be observed during a session by using the saga_show_tmpfiles function, and can be removed using the saga_remove_tmpfiles function. Note that this function also removes any accompanying files, i.e. the ’.prj’ and ’.shx’ files that may be written as part of writing a ESRI Shapefile ’.shp’ format

Usage

```
saga_remove_tmpfiles(h = 0)
```

Arguments

h	Remove temporary files that are older than h (in number of hours).

Value

Nothing is returned.

Examples

```
## Not run:
# Remove all temporary files generated by Rsagacmd
saga_remove_tmpfiles(h = 0)

## End(Not run)
```

---

saga_show_tmpfiles *List temporary files created by Rsagacmd*

Description

For convenience, functions in the Rsagacmd package create temporary files if any outputs for a SAGA-GIS tool are not specified as arguments. Temporary files in R are automatically removed at the end of each session. However, when dealing with raster data, these temporary files potentially can consume large amounts of disk space. These temporary files can be observed during a session by using the saga_show_tmpfiles function, and can be removed using the saga_remove_tmpfiles function.

Usage

```
saga_show_tmpfiles()
```

Value

returns the file names of the files in the temp directory that have been generated by Rsagacmd. Note this list of files only includes the primary file extension, i.e. '.shp' for a shapefile without the accessory files (e.g. .prj, .shx etc.).

Examples

```
## Not run:
# Show all temporary files generated by Rsagacmd
saga_remove_tmpfiles(h = 0)

## End(Not run)
```

saga_version *Return the installed version of SAGA-GIS*

Description

Intended to be used internally by saga_env. Uses a system call to saga_cmd to output version of installed SAGA-GIS on the console

Usage

```
saga_version(saga_cmd)
```

Arguments

saga_cmd	The path of the saga_cmd binary.

Value

A numeric_version with the version of SAGA-GIS found at the cmd path.

search_saga *Automatic search for the path to a SAGA-GIS installation*

Description

Returns the path to the saga_cmd executable.

Usage

```
search_saga()
```

Details

On Microsoft Windows, automatic searching will occur first in 'C:/Program Files/SAGA-GIS'; 'C:/Program Files (x86)/SAGA-GIS'; 'C:/SAGA-GIS'; 'C:/OSGeo4W'; and 'C:/OSGeo4W64'.

On Linux, saga_cmd is usually included in PATH, if not an automatic search is performed in the '/usr/' folder.

For macOS, since version 8.5, SAGA-GIS is available as an standalone macOS app from SourceForge. The 'SAGA.app' package is searched first (assuming that it is installed in the '/Applications/' folder). Other macOS locations that are searched include '/usr/local/bin/' (for Homebrew installations) and within the QGIS application (SAGA-GIS is bundled with the QGIS application on macOS by default).

If multiple versions of SAGA-GIS are installed on the system, the path to the newest version is returned.

Value

The path to installed saga_cmd binary.

`search_tools` *Search for a SAGA-GIS tool*

Description

Search for a SAGA-GIS tool

Usage

```
search_tools(x, pattern)
```

Arguments

`x`	saga object
`pattern`	character, pattern of text to search for within the tool name

Value

a tibble containing the libraries, names and parameters of the tools that match the pattern of the search text and their host library

Examples

```
## Not run:
# initialize Rsagacmd
saga <- saga_gis()

# search for a tool
search_tools(x = saga, pattern = "terrain")

## End(Not run)
```

`show_raster_formats` *List the available raster formats that can be set as defaults for a 'saga' object.*

Description

List the available raster formats that can be set as defaults for a 'saga' object.

Usage

```
show_raster_formats()
```

Value

tibble

Examples

```
show_raster_formats()
```

`show_vector_formats` *List the available vector formats that can be set as defaults for a 'saga' object.*

Description

List the available vector formats that can be set as defaults for a 'saga' object.

Usage

```
show_vector_formats()
```

Value

tibble

Examples

```
show_vector_formats()
```

`tidy.saga` *Summarize the libraries that are available within a saga object and return these as a tibble.*

Description

Summarize the libraries that are available within a saga object and return these as a tibble.

Usage

```
## S3 method for class 'saga'
tidy(x, ...)
```

Arguments

`x`	a 'saga' object
`...`	additional arguments. Currently unused.

Value

a tibble that describes libraries, their descriptions and number of tools that are available in SAGA-GIS.

Examples

```
## Not run:
# Initialize a saga object
saga <- saga_gis()

# tidy the saga object's parameters into a tibble
tidy(saga)

## End(Not run)
```

`tidy.saga_library` *Summarize the tools that are available within a saga library and return these as a tibble.*

Description

Summarize the tools that are available within a saga library and return these as a tibble.

Usage

```
## S3 method for class 'saga_library'
tidy(x, ...)
```

Arguments

`x`	a ‘saga_library‘ object
`...`	additional arguments. Currently unused.

Value

a tibble that describes the tools and their descriptions within a particular SAGA-GIS library.

Examples

```
## Not run:
# Initialize a saga object
saga <- saga_gis()

# tidy the library's parameters into a tibble
tidy(saga$climate_tools)

## End(Not run)
```

tidy.saga_tool *Summarize the parameters that are available within a SAGA-GIS tool and return these as a tibble.*

Description

Summarize the parameters that are available within a SAGA-GIS tool and return these as a tibble.

Usage

```
## S3 method for class 'saga_tool'
tidy(x, ...)
```

Arguments

x	a ‘saga_tool‘ object
...	additional arguments. Currently unused.

Value

a tibble that describes tools, identifiers used by the saga_cmd command line tool, the equivalent argument name used by Rsagacmd, and other options and descriptions.

Examples

```
## Not run:
# Initialize a saga object
saga <- saga_gis()

# tidy the tools parameters into a tibble
tidy(saga$ta_morphometry$slope_aspect_curvature)

## End(Not run)
```

tile_geoprocessor *Split a raster grid into tiles for tile-based processing*

Description

Split a raster grid into tiles. The tiles are saved as Rsagacmd temporary files, and are loaded as a list of R objects for further processing. This is a function to make the the SAGA-GIS grid_tools / tiling tool more convenient to use.

Usage

```
tile_geoprocessor(x, grid, nx, ny, overlap = 0, file_path = NULL)
```

Arguments

x	A ‘saga‘ object.
grid	A path to a GDAL-supported raster to apply tiling, or a SpatRaster.
nx	An integer with the number of x-pixels per tile.
ny	An integer with the number of y-pixels per tile.
overlap	An integer with the number of overlapping pixels.
file_path	An optional file file path to store the raster tiles.

Value

A list of SpatRaster objects representing tiled data.

Examples

```
## Not run:
# Initialize a saga object
saga <- saga_gis()

# Generate a random DEM
dem <- saga$grid_calculus$random_terrain(radius = 15, iterations = 500)

# Return tiled version of DEM
tiles <- tile_geoprocessor(x = saga, grid = dem, nx = 20, ny = 20)

## End(Not run)
```

Index