Package 'link2GI'

January 22, 2017

Type Package

Title Linking GIS, Remote Sensing and Other Command Line Tools

Version 0.1-0

Date 2016-12-24

Author Chris Reudenbach [cre, aut]

Encoding UTF-8

Maintainer Chris Reudenbach <reudenbach@uni-marburg.de>

Description Functions to simplify the linking of open source GIS and remote sensing related command line interfaces.

License GPL (>= 3) | file LICENSE

Depends R (>= 2.10), methods

Imports sp, raster, rgdal, gdalUtils, tools, rgrass7, sf

RoxygenNote 5.0.1

Suggests knitr, rmarkdown

SystemRequirements GNU make SAGA GIS GRASS GIS (7.x.x) Orfeo ToolBox

NeedsCompilation no

Repository CRAN

Date/Publication 2017-01-22 17:02:18

R topics documented:

`add2Path` *Adds a defined variable and value to the global search path*

Description

Adds a variable to the global search path of the current environment

Usage

```
add2Path(newPath)
```

Arguments

`newPath`	the path that is added

Author(s)

Chris Reudenbach

Examples

```
## Not run:
# add path
addPath("pathtosomewhere")

## End(Not run)
```

checkPCDomain *Checks if running on a specified computer domain*

Description

Checks if the computer belongs to the Marburg Universitys computer pools

Usage

```
checkPCDomain(cliCode = NULL, prefixPC = "PCRZP")
```

Arguments

cliCode	code of the sofware currently saga and otb are supported
prefixPC	contains the an arbitrary part of the computer name. It always starts with the first letter.

Author(s)

CR

Examples

```
## Not run:
# add path
checkPCDomain("saga",prefixPC="PCRZP")

## End(Not run)
```

getGrassParams4W *Initialize and set up 'GRASS GIS' and* rgrass7 *for the usage on 'Windows' OS*

Description

Initialize the enviroment variables on a 'Windows' OS for using 'GRASS GIS' via rgrass7

Usage

```
getGrassParams4W(setDefaultGrass = NULL, DL = "C:", verSelect = FALSE)
```

Arguments

setDefaultGrass	default = NULL forces a full search for 'GRASS GIS' binaries. You may alternatively provide a vector containing pathes and keywords. c("C:/OSGeo4W64","grass-7.0.5","osgeo4W") is valid for a typical osgeo4w installation.
DL	raster or sp object
verSelect	if TRUE you must interactivley selcect between alternative installations

Details

The concept is very straightforward but for an all days usage pretty helpful. You need to provide a raster or a sp object. The derived properties are used to initialize a temporary but static rgrass7 environment. During the rsession you will have full access to GRASS7 both via the wrapper package as well as the command line.

Value

getGrassParams4W initializes the usage of GRASS7.

Examples

```
## Not run:
# automatic retrieval of valid 'GRASS GIS' environment settings
# if more than one is found the user has to choose.
getGrassParams4W()

# typical standalone installation
getGrassParams4W(c("C:/Program Files/GRASS GIS 7.0.5","GRASS GIS 7.0.5","NSIS"))

# typical OSGeo4W64 installation
getGrassParams4W(c("C:/OSGeo4W64","grass-7.0.5","osgeo4W"))

## End(Not run)
```

---

getGrassParams4X *Initialize and set up 'GRASS GIS' and rgrass7 for the usage on 'Linux' OS*

Description

Initialize and set up rgrass7 for 'Linux'

Usage

```
getGrassParams4X(setDefaultGrass = NULL, MP = "/usr", verSelect = FALSE)
```

Arguments

setDefaultGrass	default = NULL will force a search for 'GRASS GIS' You may provide a valid combination as c("C:/OSGeo4W64","grass-7.0.5","osgeo4w")
MP	mount point to be searched. default is "usr"
verSelect	if TRUE you must interactivley selcect between alternative installations

Details

During the rsession you will have full access to GRASS7 GIS via the rgrass7 wrappe. Additionally you may use also use the API calls of GRASS7 via the command line.

Examples

```
## Not run:
# automatic retrieval of the GRASS7 enviroment settings
getGrassParams4X()

# typical standalone installation
getGrassParams4X("/usr/bin/grass72")

# typical user defined installation (compiled sources)
getGrassParams4X("/usr/local/bin/grass72")

## End(Not run)
```

---

getSpatialClass *Checks if x is a raster or sp object*

Description

Checks if x is a raster or sp object

Usage

```
getSpatialClass(obj)
```

Arguments

obj	R raster* or sp object

Author(s)

Chris Reudenbach

Examples

```
## Not run:
# add path
getSpatialClass(x)

## End(Not run)
```

`initProj` *Defines and creates folders and variables*

Description

Defines and creates (if necessary) all folders variables set the SAGA path variables and other system variables exports all variables to the global environment

Usage

```
initProj(projRootDir = getwd(), projFolders = c("data/", "result/", "run/",
  "log/"))
```

Arguments

`projRootDir`	project github root directory (your github name)
`projFolders`	list of subfolders in project

`link2GI` *Linking the GI-World*

Description

A simple wrapper for linking GI/RS functionality to R

Details

Functions for linking GI/RS functionality to R

Note

It is important to keep in mind that all binaries need to be installed correctly on your system. the link2GI package just tries to generate correct settings, system variables and so on for most of the known issues. It is tested for 'Windows 7' and 'Windows 10' as well as for the most popular 'Linux' distributions. The 'OSX' operation system should run but is not tested.

Author(s)

Chris Reudenbach
Maintainer: Chris Reudenbach <reudenbach@uni-marburg.de>

linkgdalUtils *Search and export the settings of* gdalUtils *and 'GDAL' binaries.*

Description

Check and export the gdalUtils and 'GDAL' settings. You need to have installed the 'GDAL' binaries.

Usage

```
linkgdalUtils()
```

Value

A list of the complete capabilities of the current installed GDAL version

Author(s)

CR

Examples

```
## Not run:

# get all available driver
gdal<- linkgdalUtils()

gdal[[1]]$drivers$format_code

# GET BINARY PATH
gdal[[1]]$path

# get additional and available python tools
gdal[[1]]$python_utilities

## End(Not run)
```

linkGRASS7 *Initializes environment variables and pathes for GRASS7*

Description

Initializes the environment and the pathes for 'GRASS GIS 7.x' The correct linkage to 'GRASS GIS' is performed by using an existing and valid raster or sp object.

Usage

```
linkGRASS7(x = NULL, defaultGrass = NULL, searchPath = NULL,
  verSelect = FALSE)
```

Arguments

x	raster or sp object
defaultGrass	if NULL an automatic search will be performed. You may also provide a valid combination as c("C:/OSGeo4W64","grass-7.0.5","osgeo4w")
searchPath	path or mounting point that will be searched
verSelect	boolean if TRUE you may choose interactively the binary version (if found more than one), by default FALSE

Details

The concept is very straightforward but for an all days usage pretty helpful. You need to provide a raster or sp spatial object which has to be correctly georeferenced. The resulting params will be used to initialize a temporary but static rgrass7 environment.

If you want speed up the init process (mainly the search over your hard disk) you can provide a correct parameter set. Best way to do so is to call search

Value

linkGRASS7 initializes the usage of GRASS7.

Note

'GRASS GIS 7' is excellently supported by the rgrass7 wrapper package. Nevertheless 'GRASS GIS' is well known for its high demands regarding the correct workspace and environment setup. This becomes even worse on 'Windows' platforms or if alternative 'GRASS GIS' installations are available. While the setup function initGRASS that is provided by the rgrass7 package, works fine under Linux and for known pathes and environmental variables, one will find that the integration of a 'Windows' based 'GRASS GIS' especially if provided by 'OSGeo4W' http://trac.osgeo.org/osgeo4w/ and/or the parallel installations of different software versions will be cumbersome. The function linkGRASS7 tries to find all valid 'GRASS GIS' binaries by analyzing the startup files of 'GRASS GIS'. After identifying 'GRASS GIS' binaries all necessary system variables and settings will be performed.

If you have more than one valid installation you will be ask to select one.

Author(s)

Chris Reudenbach

Examples

```
## Not run:
# get meuse data
library(sp)
data(meuse)
coordinates(meuse) <- ~x+y
proj4string(meuse) <-CRS("+init=epsg:28992")

# automatic search and find of GRASS binaries if
# more than one you have to choose.
linkGRASS7(meuse)

# call if you do not have any idea if and where GRASS is installed
# Actually this linking procedure is highly recommended
linkGRASS7(meuse)

# assuming a typical standalone non-OSGeo4W installation
# You must provide at
linkGRASS7(meuse,c("C:/Program Files/GRASS GIS7.0.5","GRASS GIS 7.0.5","NSIS"))

# assuming a typical OSGeo4W installation
linkGRASS7(meuse,c("C:/OSGeo4W64","grass-7.0.5","osgeo4W"))

# string for Linux c("/usr/bin","grass72") '

## End(Not run)
```

`linkOTB` *Locate and set up 'Orfeo ToolBox' API bindings*

Description

Locate and set up 'Orfeo ToolBox' API bindings

Usage

```
linkOTB(binPathOtb = NULL, rootPathOtb = NULL, otbType = NULL,
  DL = "C:", verSelect = FALSE)
```

Arguments

`binPathOtb`	string contains path to where the otb binaries are located
`rootPathOtb`	string provides the root folder of the `binPathOtb`
`otbType`	string
`DL`	string hard drive letter default is `C:`
`verSelect`	boolean default is FALSE. If there is more than one 'OTB' installation and `verSelect` = TRUE the user can select interactively the preferred 'OTB' version

Details

It looks for the `otb_cli.bat` file. If the file is found in a `bin` folder it is assumed to be a valid 'OTB' binary installation.

if called without any parameter `linkOTB()` it performs a full search over the hardrive `C:`. If it finds one or more 'OTB' binaries it will take the first hit. You have to set `verSelect = TRUE` for an interactive selection of the preferred version.

Value

add otb pathes to the enviroment and creates global variables otbPath

Note

You may also set the path manually. Using a 'OSGeo4W64' http://trac.osgeo.org/osgeo4w/ installation it is typically `C:/OSGeo4W64/bin/`

Author(s)

Chris Reudenbach

Examples

```
## Not run:
# call if you do not have any idea if and where OTB is installed
linkOTB()

# call it for a default OSGeo4W installation of the OTB
linkOTB("C:/OSGeo4W64/bin/")

# call it for a default Linux installation of the OTB
linkOTB("/usr/bin/")

## End(Not run)
```

---

linkSAGA *Locate and bind valid SAGA installation(s)*

Description

Locate and bind valid SAGA GIS installation(s). It returns the pathes and correct environment settings. All valid means that it looks for the `saga_cmd` or `saga_cmd.exe` executables. If the file is found it is assumed to be a valid 'SAGA GIS' installation.

Usage

```
linkSAGA(defaultSAGA = NULL, DL = "C:", MP = "/usr", verSelect = FALSE)
```

Arguments

defaultSAGA	string contains path to SAGA binaries
DL	drive letter
MP	mount point
verSelect	boolean default is FALSE. If there is more than one 'SAGA GIS' installation and verSelect = TRUE the user can select interactively the preferred 'SAGA GIS' version

Details

If called without any parameter linkSAGA() it performs a full search over C:. If it finds one or more 'SAGA GIS' binaries it will take the first hit. You have to set verSelect = TRUE for an interactive selection of the preferred version. Additionally the selected SAGA pathes are added to the environment and the global variables sagaPath, sagaModPath and sagaCmd will be created.

Note

The excellent 'SAGA GIS' wrapper RSAGA is NOT used because the the developemt of 'SAGA GIS' breaks permanently the API call syntax. This fact makes it highly impracticable to keep with the wrapper adaptions in line. RSAGA will meet perfectly your needs if you use 'SAGA GIS' versions from 2.0.4 - 2.2.3.

Examples

```
## Not run:

# call if you do not have any idea if and where SAGA GIS is installed
linkSAGA()

# typical OSGeo4W64 installation
linkSAGA(c("C:/OSGeo4W64/apps/saga","C:/OSGeo4W64/apps/saga/modules"))

## End(Not run)
```

---

makGlobalVar *Generates a variable with a certain value in the R environment*

Description

Generates a variable with a certain value in the R environment

Usage

```
makGlobalVar(name, value)
```

Arguments

name	character string name of the variable
value	character string value of the variable

Examples

```
## Not run:

# creates the global var \code{pathToData} with the value \code{~/home/data}
makGlobalVar("pathToData","~/home/data")


## End(Not run)
```

---

searchGRASSW *Search recursivly valid 'GRASS GIS' installation(s) on a given 'Windows' drive*

Description

Provides an list of valid 'GRASS GIS' installation(s) on your 'Windows' system. There is a major difference between osgeo4W and standalone installations. The functions trys to find all valid installations by analysing the calling batch scripts.

Usage

```
searchGRASSW(DL = "C:")
```

Arguments

DL	drive letter to be searched, default is "C:"

Value

A dataframe with the 'GRASS GIS' root folder(s), version name(s) and installation type code(s)

Author(s)

Chris Reudenbach

Examples

```
## Not run:
# get all valid 'GRASS GIS' installation folders and params at "C:"
searchGRASSW()

## End(Not run)
```

searchGRASSX *Search recursivly valid 'GRASS GIS' installation(s) at a given 'Linux' mount point*

Description

Search for valid 'GRASS GIS' installations at a given 'Linux' mount point

Usage

```
searchGRASSX(MP = "/usr")
```

Arguments

MP	default is /usr

Value

A dataframe containing 'GRASS GIS' binary folder(s), version name(s) and installation type code(s)

Author(s)

Chris Reudenbach

Examples

```
## Not run:
# get all valid 'GRASS GIS' installation folders in the /usr directory (typical location)
searchGRASSX("~/usr")

# get all valid 'GRASS GIS' installation folders in the home directory
searchGRASSX("~/")

## End(Not run)
```

searchOTBW *Search recursively for valid 'OTB' installation(s) on a 'Windows' OS*

Description

Search for valid 'OTB' installations on a 'Windows' OS

Usage

```
searchOTBW(DL = "C:")
```

Arguments

DL	drive letter default is "C:"

Value

A dataframe with the 'OTB' root folder(s) the version name(s) and the installation type(s).

Author(s)

Chris Reudenbach

Examples

```
## Not run:
#### Examples how to use RSAGA and OTB bindings from R

# get all valid OTB installation folders and params
searchOTBW()

## End(Not run)
```

---

searchSAGAW *Search recursively for valid 'Windows' 'SAGA GIS' installation(s)*

Description

Search for valid 'SAGA GIS' installation(s) on a given 'Windows' drive

Usage

```
searchSAGAW(DL = "C:", verSelect = FALSE)
```

Arguments

DL	drive letter default is "C:"
verSelect	boolean default is FALSE. If there is more than one 'SAGA GIS' installation and verSelect = TRUE the user can select interactively the preferred 'SAGA GIS' version

Value

A dataframe contasining the 'SAGA GIS' root folder(s), the version name(s) and the installation type(s)

Author(s)

Chris Reudenbach

Examples

```
## Not run:
#### Examples how to use searchSAGAW

# get all valid SAGA installation folders and params
sagaParams<- searchSAGAW()

## End(Not run)
```

`setGrassEnv4W` *Initializes and set up access to 'GRASS GIS 7.xx' via the rgrass7 wrapper or command line packages*

Description

Initializes and set up access to 'GRASS GIS 7.xx' via the rgrass7 wrapper or command line packages

Usage

```
setGrassEnv4W(grassRoot = "C:\\OSGEO4~1", grassVersion = "grass-7.0.5",
  installationType = "osgeo4W", jpgmem = 1e+06)
```

Arguments

`grassRoot`	grass root directory i.e. "C:\OSGEO4~1",
`grassVersion`	grass version name i.e. "grass-7.0.5"
`installationType`	two options "osgeo4w" as installed by the 'OSGeo4W'-installer and "NSIS" that is typical for a standalone installtion of 'GRASS GIS'.
`jpgmem`	jpeg2000 memory allocation size. Default is 1000000

Value

Set all necessary environment variables and additionally returns the GISBASE directory as string.

Author(s)

Chris Reudenbach

Examples

```
## Not run:
# get all valid 'GRASS GIS' installation folders and params
grassParam<- setGrassEnv4W()

## End(Not run)
```

setOTBEnv *Initializes and set up access to the 'OTB' command line interface*

Description

Initializes and set up access to the 'OTB' command line interface

Usage

```
setOTBEnv(binPathOtb = NULL, rootPathOtb = NULL)
```

Arguments

binPathOtb	string contains the path to the 'OTB' binaries
rootPathOtb	string contains the full string to the root folder containing the 'OTB' installation'

Value

Adds 'OTB' pathes to the enviroment and creates the variable global string variable otbCmd, that contains the path to the 'OTB' binaries.

Examples

```
## Not run:
## example for the most common default OSGeo4W64 installation of OTB
setOTBEnv(binPathOtb="C:\\OSGeo4W64\\bin\\",rootPathOtb="C:\\OSGeo4W64")

## End(Not run)
```

Index