# Package ‘Rlpj’

May 12, 2018

**Title** A R package that wraps the LPJ-GUESS 3.1 model

**Version** 1.0.1

**Description** The RLPJ package provides functions to run LPJ-Guess within R. It also allows to parallelize the model execution on personal laptops and on HPC.

**Depends** R (>= 3.1.0)

**Imports** methods, zoo, snow

**Suggests** knitr, Rmpi, rmarkdown

**License** file LICENSE

**LazyData** true

**VignetteBuilder** knitr

**RoxygenNote** 6.0.1

**NeedsCompilation** no

**Author** Florian Hartig [aut],
Maurizio Bagnara [aut, cre],
Ramiro Silveyra Gonzalez [aut]

**Maintainer** Maurizio Bagnara <maurizio.bagnara@senckenberg.de>

## R topics documented:

## `Rlpj-package` *Overview of the functions in the Rlpj package*

### Description

The RLPJ package provides functions to run LPJ-GUESS within R. It also allows to parallelize the execution of LPJ-GUESS on personal laptops and on HPC.

The package should be particularly useful for users aiming at integrating LPJ outputs in R routines and parellizing the model.

Below is a list of the functions grouped by theme. See the vignette for more information and some examples (you can open it by running this command: `vignette('Rlpj')`)

### I. Run LPJ-Guess

- `runLPJ` To run LPJ-GUESS serial/parallel
- `setupLPJParallel` To create a parallel setup
- `exitMPI` To exit MPI clusters

### II. Visualize data

- `plotLPJData` To plot data from LPJData objects

### III. Utility functions

- `getTemplate` To obtain the in-package stored model templates
- `getParameterList` To obtain the parameter default values
- `getDesign` To obtain the default desgin
- `getTypeList` To obtain the default output model types
- `getRunInfo` To recover data or parameters from the runInfoDir

### IV. Other functions

- `callLPJ` To make a system call for LPJ-GUESS
- `getLPJData` To read and process LPJ-GUESS outputs
- `writeTemplate` To write LPJ-GUESS templates

### Author(s)

Except where indicated otherwise, the functions in this package were written by Ramiro Silveyra Gonzalez, Maurizio Bagnara and Florian Hartig

## callLPJ *A function to call the LPJ-GUESS modell*

### Description

This function does a system call, passing a template to the LPJ model.

### Usage

```
callLPJ(mainDir, runDir, template2, mode)
```

### Arguments

| | |
|---|---|
| `mainDir` | a character string indicating the path to LPJ-GUESS executable |
| `runDir` | a character string indicating the path to the runDirectory |
| `template2` | a character string providing the "specific" model template, e.g, global_cf.ins or global_cru.ins. Provide only the file name, not the path. If not provided, package templates will be used |
| `mode` | a character string indicating whether using cru or cf data |

### Value

none

### Note

Based an older code of Istem Fer, Uni Potsdam

### Author(s)

Ramiro Silveyra Gonzalez, Maurizio Bagnara, Florian Hartig

### Examples

```
## Not run:
callLPJ("/home/LPJrun", "/home/LPJrun/runDirectory1", "global_cru.ins")

## End(Not run)
```

## exitMPI *The function to close MPI connection on cluster.*

### Description

This function will close slaves and finalize mpi.

### Usage

```
exitMPI()
```

### Details

The exitMPI should be use when working on MPI clusters. It must be called at the end of you script. Be aware that when mpi is exited, it will be no longer possible to work on MPI clusters without relaunching R. The function is a wrapper of mpi.finalize from the Rmpi package. Check the package manual for futher advise on using mpi.finalize and mpi.quit.

### Author(s)

Ramiro Silveyra Gonzalez, Maurizio Bagnara, Florian Hartig

### See Also

https://cran.r-project.org/web/packages/Rmpi/Rmpi.pdf

### Examples

```
## Not run:
exitMPI()

## End(Not run)
```

## getDesign *A get design function*

### Description

This function returns the LPJ-GUESS default design for running the model

### Usage

```
getDesign(scale, list = F)
```

### Arguments

`scale` a character string indicating whether the parameters are for the global or europe template

`list` a character boolean to specify the returned format. If TRUE the parameter will be returned as a list, otherwise as a matrix.

### Value

a list or a matrix with design parameter names and their values

### Author(s)

Ramiro Silveyra Gonzalez, Maurizio Bagnara

### Examples

```
## Not run:
parameterList <- getParameterList("global")

## End(Not run)
```

## getLPJData *A function to process LPJ-GUESS outputs*

### Description

This function reads the ASCII outputs produced by LPJ-GUESS. It takes a list of output types (typeList) and finds them in the specified path. The data is stored in a data class object: LPJData. If processing is TRUE, the data will be stored as zoo time series. Ohterwise, as data frames.

### Usage

```
getLPJData(x, typeList = NULL, runInfo = NULL, processing = FALSE)
```

### Arguments

`x` a character string indicating path to the output files

`typeList` a character vector with the outputs to be analyzed. Default value is all

`runInfo` a named list with the information of the LPJ run The runInfo it will be stored by the function as RData along with the processed outputs of the model (optional)

`processing` a boolean indicating whether output files will be turned into time series (default is FALSE)

### Details

To convert the outputs into zoo time series is only supported when running the model for one grid cell. For running LPJ-GUESS for several grid cells, please set processing to FALSE.

## Value

the processed data returned in a S4 Class: LPJData Class

## Note

Based on an older code of Joerg Steinkamp

## Author(s)

Ramiro Silveyra Gonzalez, Maurizio Bagnara, Florian Hartig

## See Also

LPJData, https://cran.r-project.org/web/packages/zoo/zoo.pdf

## Examples

```
## Not run:
LPJout <- getLPJData( typeList = c("aaet",  "cflux","lai", "nflux"),
          "~/path/to/output/files", runInfo = list(parameter1 = 0.5, grid = 1))

## End(Not run)
```

---

# getParameterList *A get paremeter list function*

---

## Description

This function returns the LPJ-GUESS parameter list required for the writeTemplate function. It also allows users to see the default values of the templates.

## Usage

```
getParameterList(scale, list = TRUE)
```

## Arguments

| | |
|---|---|
| scale | a character string indicating whether the parameters are for the global or europe template |
| list | a character boolean to specify the returned format. If TRUE the parameter will be returned as a list, otherwise as a matrix. |

## Value

a list or a matrix with parameter names and their values

### Author(s)

Ramiro Silveyra Gonzalez, Maurizio Bagnara

### Examples

```
## Not run:
parameterList <- getParameterList("global")

## End(Not run)
```

## getRunInfo    *A get runInfo data*

### Description

This function reads the runLPj outputs stored in the runInfoDir and returns as a list, with the same structure as the runLPJ outputs. Additionally, allows to retrieve the parameters from the runInfoDir files or the runLPJ outputs.

### Usage

```
getRunInfo(x, parameters = F)
```

### Arguments

`x` a character string indicating the absolute path to the runInfoDir folder or a R object produced by runLPJ

`parameters` a character boolean to specify whether to return the parameters instead of the LPJData objects

### Author(s)

Ramiro Silveyra Gonzalez

### Examples

```
## Not run:
# recover the data from the runInfo folder
result <- getRunInfo("/some/absolute/path/runInfo_2016_08_11_121507")

# recover only the paramaters from the runInfo folder
parameters <- getRunInfo("/some/absolute/path/runInfo_2016_08_11_121507", parameters =T)

# recover only the parameters from the result
parameters <- getRunInfo(result, parameters =T)


## End(Not run)
```

## getTemplate *A get template function*

### Description

This function returns the LPJ-GUESS templates.

### Usage

```
getTemplate(type, outputDir = NULL)
```

### Arguments

| | |
|---|---|
| `type` | a character string indicating the template name: global, global_cf, global_cru, europe, europe_cf, europe_cru |
| `outputDir` | a character string indicating path to the output directory (optional) |

### Value

a template object or template.ins file in the specified folder

### Author(s)

Ramiro Silveyra Gonzalez, Maurizio Bagnara, Florian Hartig

### Examples

```
## Not run:
template <- getTemplate("global.ins")
getTemplate("global.ins", "/home/LPJTemplates/")

## End(Not run)
```

## getTypeList *A get type list function*

### Description

This function returns the LPJ-GUESS output type list

### Usage

```
getTypeList()
```

### Value

a character vector with the default output types

### Author(s)

Ramiro Silveyra Gonzalez, Maurizio Bagnara

### Examples

```
## Not run:
typelist <- getTypeList()

## End(Not run)
```

## `LPJData-class` *A LPJData class object*

### Description

This is a S4 class object to store both the LPJ-GUESS outputs and the information that was passed to LPJ-GUESS.

### Value

a S4 class object with two slots

- runInfo contains a list with the information used to run the model such as templates and parameters
- dataTypes contains a list with the model outputs

### Author(s)

Ramiro Silveyra Gonzalez

## `LPJSetup-class` *A LPJSetup class object*

### Description

This is a S4 class object to store the configuration for parallel runs to be used by runLPJ.

### Value

a S4 class object with five slots

- clusterType a character string indicating the type of cluster to be created (MPI o SOCK)
- numCores a integer specifying number of cores of the cluster
- mainDir a character string indicating the path to the directory where the model link and the templates are located the directory structure for the outputs
- runDir a character string indicating the path to the directory where the needed files for model run are located
- outDir a character string indicating the path to the directory where the model outputs will be saved

### Author(s)

Ramiro Silveyra Gonzalez

## plotLPJData *A plot function for LPJData objects*

### Description

This function reads data from a LPJData object and plots the variables against time. If the save.plots is set to TRUE, plots are saved in the output folder.

### Usage

```
plotLPJData(x, typeList = NULL, outDir = NULL, save.plots = FALSE,
  prefix = "")
```

### Arguments

`x` a LPJData object.

`typeList` a character vector with the outputs to be plotted

`outDir` a character string indicating the folder where the plots will be saved, if save.plot set to TRUE

`save.plots` a boolean indicating whether the plots are saved in the outDir. Plots will be saved as pdf

`prefix` a character string specifying the prefix to be added to the plots files. Only relevant if saving plots is TRUE

### Value

plots for data types included in typeList. The grid cells will be plotted independently

### Author(s)

Ramiro Silveyra Gonzalez, Maurizio Bagnara, Florian Hartig

### See Also

https://cran.r-project.org/web/packages/zoo/zoo.pdf

### Examples

```
## Not run:
plotLPJData(data, typeList = c("aaet", "lai"),
 outDir = "/runDir/outDir", save.plots = FALSE)

## End(Not run)
```

runLPJ *The function to run the LPJ-GUESS in parallel*

### Description

This function allows to run LPJ-GUESS the model serial or parallel and returns the model outputs as an R object, which is also stored as RData.

### Usage

```
runLPJ(x, settings, typeList = NULL, parameterList = NULL)
```

### Arguments

`x` either a LPJSetup object created with the setupLPJParallel function or a character string indicating the path to the directory where the model link and template are located, and in which the function will create the directory structure for the outputs

`settings` additional parameters

- gridList a character string providing the name of the text file with the grids to be included in the model, e.g, gridlist.txt. It must be in the mainDir. Provide only the file name, not the path
- mode a character string indicating whether using cru or cf data
- scale a character string indicating whether the model runs global or for europe
- mode a character string indicating whether using cru or cf data
- file.co2 a character string providing the absolute path to the C02 input file
- file.cru a character string providing the absolute path to the cru input file
- file.cru.misc a character string providing the absolute path to the cru misc input file
- file.ndep a character string providing the absolute path to the nitrogen deposition input file
- file.temp a character string providing the absolute path to the temperature input file
- file.prec a character string providing the absolute path to the precipitation input file
- file.insol a character string providing the absolute path to the insolation input file
- file.wetdays a character string providing the absolute path to the wetdays input file
- file.minTemp a character string providing the absolute path to the minimum temperature input file
- file.maxTemp a character string providing the absolute path to the maximum temperature input file

- variable.temp a character string providing the variable name of the temperature input file
- variable.prec a character string providing the variable name of the precipitacion input file
- variable.insol a character string providing the variable name of the insolation input file
- variable.wetdays a character string providing the variable name of the wetdays input file
- variable.minTemp a character string providing the variable name of the minimum temperature input file
- variable.maxTemp a character string providing the variable name of the maximum temperature input file
- template1 character string providing the general model template, e.g, global.ins. It must be in the mainDir. Provide only the file name, not the path. If not provided, package templates will be used
- template2 a character string providing the "specific" model template, e.g, global_cf.ins or global_cru.ins. It must be in the mainDir. Provide only the file name, not the path. If not provided, package templates will be used
- plot.data a boolean indicating whether the ouput data will be plotted (default FALSE)
- save.plots a boolean indicating whether the plots will be saved (default FALSE)
- processing a boolean indicating whether output files will be turned into zoo time series (default FALSE). This is only supported when running the model for one grid cell. For several grid cells, please set processing to FALSE
- parallel a character string providing the parallel strategy. If grids, it will parallelize grids. If parameters, it will parallelize parameters. If both, it will parallelize both grids and parameters. If auto, it will decided the strategy based on the provided parameterList and gridList. Default value is auto
- delete a boolean indicating whether output files should be deleted after processing (default TRUE). Saved plots will not be deleted
- save a boolean indicating whether function outputs should be saved as RData into an output directory named (runInfoDir_DATE). Default is TRUE
- runID an integer after which the output directory will be named (default empty). If parallel TRUE, ID is ignored and defined by setupLPJParallel
- design a named list containing the general parameters for LPJ-GUESS. Seefunction `getDesign` for default values and examples

`typeList` a character vector with the outputs to be analyzed. Default value is all outputs

`parameterList` either a named list containing the parameters to be calibrated or a matrix. If running in parallel, parameter list should be either a list of of list or a matrix where each row is a parameter combination and the column names should be named after the parameters.

## Details

The runLPJ in parallel assumes the existence of a folder the model templates for LPJ-GUESS (optional) and link to the model executable. Running the LPJ-GUESS in parallel involves two steps. First, to create a parallel setup (setupLPJParallel), and second, to actually run the model (runLPJ). The parallelization requires the package *snow* for SOCK clusters or the package *Rmpi* for MPI clusters.

## Value

an object of class LPJData. The LPJData object will be automatically stored as RData in a folder in the mainDir. The folder will be named as runInfo plus the date in format

## Warning

When using MPI clusters, please call the function exitMPI before terminating your R session.

## Model templates

The provided templates can be either the ones provided by the package or a self edited templates. The function assumes a specific coding for writing the parameters values. For this reason, we recommend to use the package templates. If using self edited templates, please take the package templates as a reference (getTemplate)

## Author(s)

Ramiro Silveyra Gonzalez, Maurizio Bagnara, Florian Hartig

## See Also

https://cran.r-project.org/web/packages/Rmpi/Rmpi.pdf, https://cran.r-project.org/web/packages/snow/snow.pdf, setupLPJParallel, exitMPI, LPJData, LPJSetup

## Examples

```
## Not run:

file.co2<-"/some/absolute/path/crudata/co2_1901-2013.txt"
file.cru <- "/some/absolute/path/crudata/cru_1901_2006.bin"
file.cru.misc <- "/some/absolute/path/crudata/cru_1901_2006misc.bin"
file.ndep <- "/some/absolute/path/crudata/GlobalNitrogenDeposition.bin"
file.temp <- "/some/absolute/path/cfdata/temp.nc"
file.prec <- "/some/absolute/path/cfdata/prec.nc"
file.insol <- "/some/absolute/path/cfdata/rad.nc"

mainDir <- "/some/absolute/path/mainDir"
list.files(mainDir)
[1] "guess" or "guesscmd.exe"  # link to the model executable
[2] "gridlist.txt"      # list of gridcells
[3] "global.ins"        # template1 (optional)
[4] "global_cru.ins"    # template2 (optional)
```

```
# General settings
settings <- list (gridList = gridList,mode = "cf", scale = "global",
                file.co2 = file.co2, file.cru = file.cru,
                file.cru.misc = file.cru.misc, file.ndep = file.ndep,
                file.temp = file.temp, file.prec = file.prec,
                file.insol = file.insol, delete = F)


# Single  Run
 result <-  runLPJ(mainDir, settings= settings)
 result
     class              : LPJData
     LPJ template 1     : global.ins
     LPJ template 2     : global_cf.ins
     grid cells         : 99  Somewhere
     run directory      : /some/absolute/path/mainDir/runDirectory
     LPJ model outputs  : 39 outputs
     aaet agpp aiso amon anpp cflux clitter cmass cpool cton_leaf dens
     firert fpc speciesheight lai maet mevap mgpp mintercep miso mlai mmon
     mnee mnpp mpet mra mrh mrunoff mwcont_lower mwcont_upper nflux ngases
     nlitter nmass npool nsources nuptake runoff vmaxnlim


#  Parallel Run
# Create some paramaters to test the model.
# Number of runs is proportional to number of parameter being testet
parameterDefault <- list (run_emax = NULL)

# Test 6 different values for emax.
par <- seq(1,5, len = 6)
# Create the list object with the parameters
parameterList <- vector("list", length(par))
for (i in 1:length(par)) {
   parameterDefault$run_emax <- par[i]
   parameterList[[i]] <- parameterDefault
 }

# Call setupParallel
mySetup  <- setupLPJParallel(3, "SOCK", "cf",
                             mainDir = "/some/absolute/path/mainDir")

# Call runLPJ
result <- runLPJ(mySetup, settings= settings, parameterList = parameterList)
str(result,1)
   List of 6
   $ :Formal class 'LPJData' [package "Rlpj"] with 2 slots
   $ :Formal class 'LPJData' [package "Rlpj"] with 2 slots
   $ :Formal class 'LPJData' [package "Rlpj"] with 2 slots
   $ :Formal class 'LPJData' [package "Rlpj"] with 2 slots
   $ :Formal class 'LPJData' [package "Rlpj"] with 2 slots
   $ :Formal class 'LPJData' [package "Rlpj"] with 2 slots
```

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

## setupLPJParallel *The function to create a setup for parallel runs of the LPJ-GUESS*

### Description

This function first creates a setup for running the LPJ-GUESS in parallel based on the provided input parameters. The function assumes a specific initial configuration. A folder (mainDir) containing the model templates for LPJ-GUESS (optional) and link to the model executable.

### Usage

```
setupLPJParallel(numCores, clusterType, mainDir)
```

### Arguments

| | |
|---|---|
| numCores | a integer specifying number of cores of the cluster |
| clusterType | a character string indicating the type of cluster to be created. If running on personal computer, type should be SOCK. If running on HPC, type should be MPI |
| mainDir | a character string indicating the path to the directory where the template and the model link are located, and in which the function will create the directory structure for the outputs |

### Value

an LPJSetup object

### Author(s)

Ramiro Silveyra Gonzalez, Maurizio Bagnara, Florian Hartig

### See Also

https://cran.r-project.org/web/packages/Rmpi/Rmpi.pdf, https://cran.r-project.org/web/packages/snow/snow.pdf

### Examples

```
## Not run:
mainDir <- "/some/absolute/path/mainDir"
list.files(mainDir)
[1] "guess" or "guesscmd.exe"  # link to the model executable
[2] "gridlist.txt"      # list of gridcells
[3] "global.ins"        # template1 (optional)
[4] "global_cru.ins"    # template2 (optional)
```

```
mySetup <- setupLPJParallel(numCores= 3, clusterType = "SOCK", mainDir=mainDir)
mySetup
     class              : LPJSetup
     cluster type       : SOCK
     number of cores    : 3
     output directories :
     /some/absolute/path/mainDir/runDirectory1
     /some/absolute/path/mainDir/runDirectory2
     /some/absolute/path/mainDir/runDirectory3

## End(Not run)
```

## writeTemplate *A writing template function for LPJ*

### Description

This function reads a template, and replaces parameters with the provides parameters list. If no parameter values are provided, the function will set them to the default values. The function assumes that a copy of the template is already placed in the run directory.

### Usage

```
writeTemplate(template1, parameterList, runDir, check = "serial")
```

### Arguments

| | |
|---|---|
| `template1` | a character string providing the general model template, e.g, global.ins. Provide only the file name, not the path |
| `parameterList` | a named list containing the parameters to be calibrated |
| `runDir` | a character string indicating path to the run directory |
| `check` | a character string indicating how to check the provided parameterList. Default value is serial. Other possible value is parallel. |

### Details

If check is serial, it will return the complete and checked parameterList. If parallel, it would only check the provided parameters. Please only use serial, other options are handled internally in the parallelization.

### Value

none

## Warning

The provided template can be either one provided by the package or a self edited template. The function assumes a specific coding for writing the parameters values. For this reason, we recommend to use the package templates. If using self edited templates, please take the package templates as a reference.

## Note

Based an older code of Istem Fer, Uni Potsdam

## Author(s)

Ramiro Silveyra Gonzalez, Maurizio Bagnara, Florian Hartig,

## See Also

getTemplate, getParameterList

## Examples

```
## Not run:
writeTemplate("global.ins", list(run_lamda_max = 0.5, run_emax= 5),
              "/home/lpjRun/runDirectory1")

## End(Not run)
```

# [,LPJData,character-method

*Extract parts of LPJData*

## Description

Extract parts of LPJData

Extract parts of LPJSetup

## Usage

```
## S4 method for signature 'LPJData,character'
x[i, j, ..., drop = TRUE]

## S4 method for signature 'LPJSetup,character'
x[i, j, ..., drop = TRUE]
```

### Arguments

`x` an `LPJData` or an `LPJSetup` object

`i` a character string indicating the slot name of an LPJData class object, an LPJ-GUESS output or a character string indicating the slot name of an LPJSetup class object

`j` a character string indicating a sublevel of any slot (only needed if i is provided)

`...` additional arguments (none implemented)

`drop` unused

---

## `[<-,LPJData,character-method` *Replace parts of LPJData*

---

### Description

Replace parts of LPJData

### Usage

```
## S4 replacement method for signature 'LPJData,character'
x[i, j, ...] <- value
```

### Arguments

`x` an `LPJData` object

`i` a character string indicating the LPJ-GUESS output

`j` a character string indicating a sublevel of the dataType slot (only needed if i is provided)

`...` additional arguments (none implemented)

`value` any value to create or replace and existing

# Index