Package 'daRt'

October 5, 2019

Type Package

Title Read DART Model Outputs

Version 0.5.0

Author William T. J. Morrison

Maintainer William T. J. Morrison <willmorrison661@gmail.com>

Description For reading outputs from the Discrete Anisotropic Radiative Transfer (DART) model, formatted in a ``long" dplyr-ready format suitable for efficient analysis.

Github https://github.com/willmorrison1/daRt

License GPL-3

Encoding UTF-8

RoxygenNote 6.1.1

R topics documented:

accessors . . . 2
as.data.frame,SimulationData-method . . . 2
deleteFiles . . . 3
Directions-class . . . 3
getData . . . 3
getFiles . . . 4
Images-class . . . 4
imagesToDirectionsDF . . . 4
plotDirections . . . 5
RB3D-class . . . 5
rb3DtoNc . . . 6
removeRelief . . . 6
resourceUse . . . 7
sequenceParameters . . . 7
SimulationData-class . . . 8
SimulationFiles-class . . . 8
simulationFilter . . . 8
SimulationFilter-class . . . 9
versionInfo . . . 10

Index **11**

accessors *Access object information*

Description

Generic functions to access information from the objects with classes defined in this package

Usage

```
product(x)

simname(x)

files(x)

bands(x)

iters(x)

variables(x)

variablesRB3D(x)

typeNums(x)

imageTypes(x)

imageNos(x)
```

as.data.frame,SimulationData-method *as.data.frame*

Description

as.data.frame

Usage

```
## S4 method for signature 'SimulationData'
as.data.frame(x, as.tibble = TRUE)
```

Arguments

x	SimulationData.

`deleteFiles` *deleteFiles*

Description

DART input files can be very large. This function deletes those large files that are not required for post-processing of data in this package.

Usage

```
deleteFiles(x = "SimulationFiles", trianglesInput = "logical",
  maketOutput = "logical")
```

Arguments

`x`	SimulationFiles-class type object.
`trianglesInput`	remove "triangles" input files? (bool)
`maketOutput`	remove "maket.txt" output file? (bool)

Details

Delete potentially large input files

`Directions-class` *Directions data class*

Description

Directions data class that extends SimulationData-class class.

`getData` *Main function: get DART data*

Description

Main function to get data from DART simulation outputs in a friendly 'long' data format that is part of an object that extends a SimulationData-class type object

Usage

```
getData(x, sF, ...)
```

Arguments

`x`	simulation directory or directories (character) or SimulationFiles-class object
`sF`	SimulationFilter-class if x = `character`

getFiles *Get DART output filenames*

Description

Get DART output filenames

Usage

```
getFiles(x = "character", sF = "SimulationFilter")
```

Arguments

x	simulation directory or directories (character)
sF	SimulationFilter-class object
...	Optional arguments of: nCores: number of cores to use when loading data.

Images-class *Images data class*

Description

Image data class extends SimulationData-class class.

imagesToDirectionsDF *imagesToDirectionsDF*

Description

Convert an Images-class object to a Directions-class object

Usage

```
imagesToDirectionsDF(x, fun)
```

Arguments

x	Images-class object
fun	Function to apply across each image.

Details

Aggregate images to single values

plotDirections *plotDirections*

Description

Plot directions data as polar plot.

Usage

```
plotDirections(azimuth, zenith, value, azimuthOffsetVal = 0,
  outerRadius = max(zenith) + max(zenith) * 0.01, zenithLabPch = 20,
  zenithLabCol = "darkgrey", zenithLabCex = 1, brks = seq(min(value),
  max(value), length.out = 10), cols = c("dark grey",
  colorRampPalette(c("purple", "blue3", "yellow", "red"))(length(brks) -
  3), "firebrick4"), ...)
```

Arguments

azimuth	Numeric. Azimuth angle with DART conventions
zenith	Numeric. Zenith angle with DART conventions
value	Numeric. Values associated with the given azimuth and zenith angles
azimuthOffsetVal	Numeric. Scene offset (degrees) as shown in the DART GUI.
outerRadius	Numeric. Maximum radius (degrees) of polar plot
zenithLabPch	Numeric. Pch for zenith label.
zenithLabCol	Character. Colour for zenith label.
zenithLabCex	Numeric. Cex for zenith label.
brks	Numeric. Breaks for colour palette e.g. seq(0, 1, by = 0.1). Optional.
cols	Character. Colours for given breaks. Optional.
...	Additional options passed to points() when drawing directions points.

Examples

```
#Inputs are DART oriented directions (as seen in the DART files and \link{Directions-class})
plotDirections(azimuth = rep(225, 10),
               zenith = seq(0, 90, length.out = 10),
               value = 1:10)
#Output plot uses 'upward' directions from ground, where e.g.:
  0deg (270deg) azimuth faces north (west)
  0deg (90deg) zenith faces upward (horizon)
```

RB3D-class *RB3D class*

Description

RB3D (Radiative Budget 3D) class that extends SimulationData-class class.

`rb3DtoNc` *rb3DtoNc*

Description

DART radiative budget .bin files can be very large. This function replaces all .bin files with .nc files, which can be compressed and are faster to read.

Usage

```
rb3DtoNc(x = "SimulationFiles", ...)
```

Arguments

`x`	SimulationFiles-class type object.
`ncCompressionFactor`	Compression factor (0 - 9) for writing ncdf files (see ncdf4 package)

Details

Convert radiative budget .bin to .nc

`removeRelief` *removeRelief*

Description

Remove underlying orography from a RB3D-class dataset using a digital elevation model (DEM) of class `RasterLayer` that is georeferenced to RB3D-class.

Usage

```
removeRelief(x = "RB3D", DEM = "RasterLayer")
```

Arguments

`x`	RB3D-class type object.
`DSM`	`RasterLayer` type object with height above ground level (m) and - preferably - a finer horizontal reoslution than that of the radiative budget cells in x. The center of the DSM must be georeferenced to the center of the radiarive budget data in x. The DSM can have a larger extent than x.

Details

Remove underlying orography

resourceUse *ResourceUse*

Description

Return a data frame with information on the resource use for a SimulationFiles-class type object

Usage

```
resourceUse(x = "SimulationFiles")
```

Arguments

x	SimulationFiles-class type object

Details

Return resource use

sequenceParameters *sequenceParameters*

Description

return a data frame. A row describes the parameters (parametre*) for a simulation (simName).

Usage

```
sequenceParameters(x)
```

Arguments

SimulationFiles-class	or SimulationData-class class object

Details

Get data frame of all sequence parameters

SimulationData-class *Generic SimulationData class*

Description

Generic SimulationData class that extends to data classes for specific DART products

Slots

`data` data.frame.

See Also

Images-class Directions-class RB3D-class

SimulationFiles-class *SimulationFiles class*

Description

An S4 class to represent the files within a simulation or simulations. Created using the `getFiles` method. Specific files within the class are modified by the object with class SimulationFilter-class

Usage

```
simdir(x)
```

Slots

`simulationFilter` contains SimulationFilter-class object

`files` a data.frame, with each row describing the file

`sequenceInfoList` a list, with each list element showing the variable permutation(s) within this specific simulation sequence.

simulationFilter *Create SimulationFilter class*

Description

Function for creating the SimulationFilter class

Usage

```
simulationFilter(product = "character", ...)
```

Arguments

product	One of: 'directions', 'rb3D', 'images'.
...	Optional arguments of: 'bands', 'variables', 'iterations', 'variablesRB3D', 'typeNums', 'imageTypes', 'imageNos'. See SimulationFilter-class for full description.

See Also

SimulationFilter-class

SimulationFilter-class

SimulationFilter class.

Description

SimulationFilter class.

Usage

```
product(x) <- value

iters(x) <- value

bands(x) <- value

variablesRB3D(x) <- value

variables(x) <- value

typeNums(x) <- value

imageTypes(x) <- value

imageNos(x) <- value
```

Slots

`bands` character e.g. "BAND0".
`variables` character e.g. "BRF".
`iters` character e.g. "ITERX".
`variablesRB3D` character e.g. "Irradiance".
`typeNums` character e.g. "2_Ground".
`imageTypes` character e.g. "ima".
`imageNos` numeric.
`product` character e.g. "directions".

See Also

simulationFilter

versionInfo *versionInfo*

Description

Get the version used for the given simulation data

Usage

```
versionInfo(x)
```

Arguments

x	SimulationFiles-class object

Details

Simulation version info

Index

accessors, 2
as.data.frame,SimulationData-method, 2

bands *(accessors)*, 2
bands<- *(SimulationFilter-class)*, 9

deleteFiles, 3
Directions-class, 3, *8*

files *(accessors)*, 2

getData, 3
getFiles, 4, *8*

imageNos *(accessors)*, 2
imageNos<- *(SimulationFilter-class)*, 9
Images-class, *4*, 4, *8*
imagesToDirectionsDF, 4
imageTypes *(accessors)*, 2
imageTypes<- *(SimulationFilter-class)*, 9
iters *(accessors)*, 2
iters<- *(SimulationFilter-class)*, 9

plotDirections, 5
product<- *(SimulationFilter-class)*, 9

RB3D-class, 5, *6*, *8*
rb3DtoNc, 6
removeRelief, 6
resourceUse, 7

sequenceParameters, 7
simdir *(SimulationFiles-class)*, 8
simname *(accessors)*, 2
SimulationData-class, *3–5*, *7*, 8
SimulationFiles-class, *3*, *6*, *7*, 8, *10*
SimulationFilter, *8*
simulationFilter, 8, *9*
SimulationFilter-class, *3*, *4*, *8*, *9*, 9

typeNums *(accessors)*, 2
typeNums<- *(SimulationFilter-class)*, 9

variables *(accessors)*, 2
variables<- *(SimulationFilter-class)*, 9
variablesRB3D *(accessors)*, 2
variablesRB3D<- *(SimulationFilter-class)*, 9
versionInfo, 10