Package 'ED2ToolsMRJ'

August 20, 2018

Title Tools for working with the Ecosystem Demography model

Version 1.0

Description Tools for working with the Ecosystem Demography model

Depends R (>= 3.4.4)

License
Creative Commons Attribution-Noncommercial-No Derivative Works 4.0, for academic use only.

Encoding UTF-8

LazyData true

Maintainer 'Miriam Johnston' <mjohnston@g.harvard.edu>

RoxygenNote 6.0.1.9000

R topics documented:

DataFrametoXML 1
ED2Summary 2
ExtractChronDates 3
ExtractVariable 4
ListAvailVars 5
PlotVars_SameAxis 6
PlotVars_Stacked 7
XMLtoDataFrame 7

Index **9**

`DataFrametoXML` *Create an ED2 XML file with PFT parameters from an R dataframe*

Description

Create an ED2 XML file with PFT parameters from an R dataframe

Usage

```
DataFrametoXML(df, outpath)
```

Arguments

df	Full path to the dataframe file (character string) OR an R dataframe object
outpath	Where the output should be saved (full path, character string)

Value

ED2-compatible XML file with PFT parameters, saved in outpath

Examples

```
df<-XMLtoDataFrame(xml="/mnt/odyssey/moorcroftfs5/mjohnston/gitruns/run005/Tonzi.xml")
outpath="/home/miriam/Desktop/DFtoXMLtest.xml"
DataFrametoXML(df,outpath)
```

ED2Summary *ED2 Model Result Diagnosis*

Description

ED2 sanity check

Usage

```
ED2Summary(ED2INdir, analydir)
```

Arguments

ED2INdir	Directory containing ED2IN (character string, with terminal "/")
analydir	Directory containing the result h5 files (character string, usually ending in "analy", with terminal "/")

Value

Summary PDF document - an ED2 "sanity check" summarizing the plants and carbon/water/energy fluxes.

Important notes on this function

- ***This function will close & re-open your RStudio, deleting environment variables***
- This function always saves the PDF as:
  ED2INdir/ED2Summary.pdf
- This function has quite a few companion scripts, all in:
  ED2ToolsMRJ/ED2Summary_CompanionFxns

Examples

```
ED2INdir<-"/mnt/odyssey/moorcroftfs5/mjohnston/ED2_Ashehad/Case7.6/run048/"
analydir<-"/mnt/odyssey/moorcroftfs5/mjohnston/ED2_Ashehad/Case7.6/run048/analy/"
ED2Summary(ED2INdir,analydir)
```

ExtractChronDates *Extract Chron Dates*

Description

Used to extract 'chron' dates from filenames in a directory

Usage

```
ExtractChronDates(dir, tscale, perfile, first, last)
```

Arguments

dir	Directory in which to search for ED2 output files (character string)
tscale	Time scale for relevant files (single character; common options: "D","I")
perfile	Number of times desired per file (optional, numeric, use only sometimes for I tscale, default =1)
first	First file from which to extract (optional, character string, use with 'last' to extract between a set of files)
last	Last file from which to extract (optional, character string, use with 'first' to extract between a set of files)

Value

'chron' format dates

Examples

```
dir="/mnt/odyssey/moorcroftfs5/mjohnston/ED2_timeoutput/Case2/run004/analy/"
first="tonzi-D-2003-07-01-000000-g01.h5"
last="tonzi-D-2003-08-01-000000-g01.h5"

#Dates from daily files
dates <- ExtractChronDates(dir,tscale="D")

#Dates from hourly files where each file has 24 hours in it
dates <- ExtractChronDates(dir,tscale="I", perfile=24)

#Dates from daily files between 07/01/2003 and 08/01/2003 (inclusive)
dates <- ExtractChronDates(dir,tscale="D", first=first, last=last)
```

`ExtractVariable` *Extract Variables from ED2*

Description

Runs a bash script to extract ED2 variables into .rds objects.

Note: This function does NOT load those .rds objects into the R session; they may be found in folders on Odyssey as reported. Why bash? Bash is *much* faster than R at variable extraction! Also, multiple bash scripts may be sumitted to Odyssey in sequence, which means that your R session isn't tied up during extraction.

Usage

```
ExtractVariable(bashpath, vars, direc, tscale, extract_option, first = NULL,
  last = NULL, singlefile = NULL)
```

Arguments

`bashpath`	Path & filename to a bash script (on Odyssey) which submits variable extraction jobs. Suggest not to change from "~/code/VarExtractED2/runscript.sh"
`vars`	variables to extract (character string, see below for odd format)
`direc`	directory in which to find the .h5 files (character string, see below for odd format)
`tscale`	time scale for variable extraction (character string, see below for odd format)
`extract_option`	ALL, RANGE, or SINGLE, depending on whether to extract the var(s) from all files in direc, files w/i a range (inclusive), or a single file (character string, see below for odd format)
`first`	Read if extract_option = "RANGE", first file from which to extract (character string, see below for odd format)
`last`	Read if extract_option = "RANGE", last file from which to extract (character string, see below for odd format)
`singlefile`	Read if extract_option = "SINGLE", single file from which to extract (character string, see below for odd format)

Details

When you input bashpath, you're telling R where to find a wrapper bash script which does the following:

1. Sets the output directory for the to-be extracted vars & associated output files
2. Saves inputs as system variables
3. Notifies the user of choices (e.g. which vars, tscale, folders,etc)
4. Loops through the variables selected and calls a batch script to submit a job to Odyssey for each

The batch script that actually submits the jobs is: ~/code/VarExtractED2/batchsript.txt
The R script that actually extracts and saves the variables is: ~/code/VarExtractED2/ExtractVar_array.R
These 3 scripts must be in the same folder. If running into problems, check the hard-coding of paths.

Value

.rds files in folders on Odyssey, as reported by the code output.

TO USE THIS FUNCTION

1. Sign in to Odyssey
2. Open an R terminal
3. Import the ED2TooldMRJ package into R on Odyssey or just copy this ExtractVar function into your workspace (easier)
4. Define inputs for the function (bashpath, vars, etc.)
5. Run on Odyssey; don't assign output to a variable.

Examples

```
#THIS IS A BIZARRE BUT REQUIRED FORMAT: double quotes WITHIN single quotes
bashpath<- "~/code/VarExtractED2/runscript.sh"
vars <- '"LEAF_TEMP" "LAI"'
direc<- '"/n/moorcroftfs5/mjohnston/ED2_timeoutput/Case2/run004/analy"'
tscale<-'"D"'
extract_option<- '"ALL"'
first<- '"tonzi-D-2006-07-29-000000-g01.h5"'
last<-  '"tonzi-D-2006-08-02-000000-g01.h5"'
singlefile<- '"tonzi-D-2006-10-29-000000-g01.h5"'

#DO NOT assign this function to an R object
ExtractVariable(bashpath,vars,direc,tscale,extract_option,first=NULL,last=NULL,singlefile=NULL)
```

`ListAvailVars` *List the ED2 variables that are available for extraction*

Description

List the ED2 variables that are available for extraction

Usage

```
ListAvailVars(direc, tscale, varsnippets)
```

Arguments

direc	Directory where there are ED2 .h5 output files (character string)
tscale	Time scale for relevant files (character vector, may have multiple entries)
varsnippets	Text which the variable name should match (optional, character vector; without varsnippets all var names in tscale will print)

Value

Names of variables available by tscale

Examples

```
direc<-"/n/moorcroftfs5/mjohnston/ED2_timeoutput/Case2/run004/analy/"

#Outputs all variable names in "D" or "I" files with the text "TEM" or "NEP"
ListAvailVars(direc=direc,tscale=c("D","I"),varsnippets=c("TEM","NEP"))

#Outputs all variable names in "D" files
#' ListAvailVars(direc=direc,tscale="D"))
```

PlotVars_SameAxis *Plot Output Variables on the Same Axis*

Description

Given dates & 1-demensional variables with the same units, plot these on the same axis

Usage

```
PlotVars_SameAxis(dates, vars, names, units)
```

Arguments

dates	'chron' dates, e.g. as extracted by ExtractChronDates function
vars	list of (1 dimensional) variables that match dates - likely NOT raw output from an extraction (list of character strings)
names	vector of names that will go on the legend (vector of character strings)
units	Of the variables (string, optional)

Value

line plot with multiple data series

Examples

```
#Dates from daily files
PlotVars_SameAxis(chrondates,vars=list("L1","L2","L3"),names=c("leaftemp1","leaftemp2","leafttemp3"),units="K"
```

PlotVars_Stacked *Plot Variables on Separate Plots, align by date*

Description

Given dates & 1-demensional variables with the same units, plot these on the separate axes

Usage

```
PlotVars_Stacked(dates, vars, names)
```

Arguments

dates	'chron' dates, e.g. as extracted by ExtractChronDates function
vars	list of (1 dimensional) variables that match dates - likely NOT raw output from an extraction (list of character strings)
names	vector of names that will go on the legend (vector of character strings)

Value

stacked line plots

Examples

```
#Dates from daily files
PlotVars_Stacked(chrondates,vars=list("AGB","LEAF_TEMP","LAI"),names=c("biomass","leaftemp","lai"))
```

XMLtoDataFrame *Extract PFT parameters from an ED2 XML*

Description

Extract PFT parameters from an ED2 XML

Usage

```
XMLtoDataFrame(xml)
```

Arguments

xml	Full path to the XML file (character string)

Value

R Dataframe where column = PFT, row.names = parameter names and values are parameter values.

Examples

```
xml<-"/mnt/odyssey/moorcroftfs5/mjohnston/gitruns/run006/Tonzi.xml"
df<-XMLtoDataFrame(xml)
#Note: warning "incomplete final line found on 'xmlfile.xml'" seems not to affect results.
```

Index

DataFrametoXML, 1

ED2Summary, 2
ExtractChronDates, 3
ExtractVariable, 4

ListAvailVars, 5

PlotVars_SameAxis, 6
PlotVars_Stacked, 7

XMLtoDataFrame, 7