Package 'eddystore'

February 8, 2019

Title Parallel Processing Of Eddy Covariance Data On JASMIN.

Version 0.2

Description Parallel Processing Of Eddy Covariance Data On .

Depends R (>= 3.5.1),
dplyr,
stringr

License MIT

Encoding UTF-8

LazyData true

RoxygenNote 6.1.1

Suggests knitr, rmarkdown

VignetteBuilder knitr

Imports RCurl

R topics documented:

eddystore-package *Parallel processing of eddy covariance data on JASMIN.*

Description

eddystore allows you to set up and run processing jobs to calculate eddy covariance fluxes using eddypro on JASMIN.

Author(s)

Maintainer: Peter Levy <plevy@ceh.ac.uk>

adjustIntervals *Adjust Date Intervals*

Description

This function adjusts the output of makeDateIntervals to match boundaries between .eddypro project files.

Usage

```
adjustIntervals(stationID_proc, procID_proc, intervals,
 fname_df_project = "/gws/nopw/j04/eddystore/eddystore_projects/df_eddystore_projects.csv")
```

Arguments

stationID_proc	A character string identifying a station in the project data frame.
procID_proc	A character string identifying a processing setup in the project data frame.
intervals	A list of the start and end times of each interval, created by makeDateIntervals.
fname_df_project	Path to the eddystore project file table.

Value

An eddystore job object.

See Also

adjustIntervals for the adjusting this to match boundaries between processing files.

Examples

```
stationID_proc <- "EasterBush"
procID_proc <- "CO2_H2O"
nIntervals <- 4
startDate_period <- "2006-07-01 00:00"
endDate_period   <- "2007-12-31 23:30"
startDate_period <- as.POSIXct(strptime(startDate_period, "%Y-%m-%d %H:%M"), tz = "UTC")
endDate_period   <- as.POSIXct(strptime(endDate_period,   "%Y-%m-%d %H:%M"), tz = "UTC")
fname_df_project = "C:/Users/plevy/Documents/eddystore_projects/df_eddystore_projects.csv"
intervals <- makeDateIntervals(startDate_period, endDate_period, nIntervals)
myJob <- adjustIntervals(stationID_proc, procID_proc, intervals, fname_df_project = fname_df_project)
```

---

checkJobCompleted *Check Status of Job on LOTUS - Check if Completed*

Description

This function checks an eddypro processing job on LOTUS. For multi-processor jobs, it only checks the first-listed in the job array, may report "DONE" before all are finished.

Usage

```
checkJobCompleted(job_name)
```

Arguments

job_name	The job name identifier given in df_job_requests.

Value

job_status Logical, TRUE if job_status == "DONE"

---

checkJobFailed *Check Status of Job on LOTUS - Check if Failed*

Description

This function checks an eddypro processing job on LOTUS. For multi-processor jobs, it only checks the first-listed in the job array,

Usage

```
checkJobFailed(job_name)
```

Arguments

job_name	An eddystore job_name specified by the job request

Value

job_status Logical, TRUE if job_status == "EXIT" i.e. failed

checkJobRunning *Check Status of Job on LOTUS - Check if Still Running*

Description

This function checks an eddypro processing job on LOTUS. For multi-processor jobs, it only checks the first-listed in the job array,

Usage

```
checkJobRunning(job_name)
```

Arguments

job_name	An eddystore job_name specified by the job request

Value

job_status Logical, TRUE if job_status == "RUN" i.e. still running

convertProjectPath *Convert paths in Eddypro project files*

Description

Need to add biomet file path changes biom_file and biom_dir

Usage

```
convertProjectPath(eddyproProjectPathName, station_dir)
```

Arguments

eddyproProjectPathName	Path to the location of the uploaded project file on JASMIN.
station_dir	Path to the location of the station directory on JASMIN e.g. "/gws/nopw/j04/eddystore/stations/Easter

Details

This function changes all references to paths in an uploaded project file to the eddystore path names.

Value

eddyproProjectPathName_new File name for project file with eddystore paths. The side effect is to write this file.

Examples

```
eddyproProjectPathName <- "C:/Users/plevy/Documents/eddystore_projects/stations/EasterBush/projects/process
station_dir <- "C:/Users/plevy/Documents/eddystore_projects/stations/EasterBush"
convertProjectPath(eddyproProjectPathName, station_dir)
```

createJob *Create Processing Job*

Description

This function creates an eddypro processing job to be run on LOTUS.

Usage

```
createJob(stationID_proc, procID_proc, startDate_period, endDate_period,
  nProcessors,
 fname_df_project = "/gws/nopw/j04/eddystore/eddystore_projects/df_eddystore_projects.csv",
 binpath = "/gws/nopw/j04/eddystore/eddypro-engine_6.2.0/eddypro-engine/bin/linux/eddypro_rp",
  switch_OS = "-s linux", eddystore_path = "/gws/nopw/j04/eddystore",
  job_name = "eddytest", user_email = "plevy@ceh.ac.uk")
```

Arguments

stationID_proc	A character string identifying a station in the project data frame.
procID_proc	A character string identifying a processing setup in the project data frame.
startDate_period	A character string for the time the run was started, used to identify the relevant files.
endDate_period	A character string for the time the run was started, used to identify the relevant files.
nProcessors	An integer number of processors to use, equal to the number of time intervals to split the processing into.
fname_df_project	Path to the eddystore project file table.
binpath	Path to the eddy_rp raw data processing binary file.
switch_OS	Switch betweeen linux and windows versions.
eddystore_path	The path to eddystore.
job_name	The job name identifier given in df_job_requests.
user_email	E-mail address to send notifications to.

Value

An eddystore job object.

See Also

adjustIntervals for the adjusting this to match boundaries between project files.

Examples

```
stationID_proc <- "EasterBush"
procID_proc <- "CO2_H2O"
nProcessors <- 4
startDate_period <- "2006-07-01 00:00"
endDate_period   <- "2007-12-31 23:30"
startDate_period <- as.POSIXct(strptime(startDate_period, "%Y-%m-%d %H:%M"), tz = "UTC")
endDate_period   <- as.POSIXct(strptime(endDate_period,   "%Y-%m-%d %H:%M"), tz = "UTC")
myJob <- createJob(stationID_proc, procID_proc, startDate_period, endDate_period, nProcessors,
   fname_df_project = "C:/Users/plevy/Documents/eddystore_projects/df_eddystore_projects.csv",
    eddystore_path = "N:/0Peter/curr/ECsystem/eddystore",
    job_name = "createJob_example", user_email = "plevy@ceh.ac.uk")
```

`get_essential_output_df`

Collate essential output files using job name

Description

Collate essential output files using job name

Usage

```
get_essential_output_df(job_name, station_dir)
```

Arguments

`job_name`	The job name identifier given in df_job_requests.
`station_dir`	The path to the station directory on eddystore.

Value

A data frame of the concatenated essential output files from each job.

`get_essential_output_df_byTime`

Collate essential output files using time only

Description

Note that curently "EasterBush" is hard-coded in the path - need to pass path argument

Usage

```
get_essential_output_df_byTime(job_startTime)
```

Arguments

`job_startTime`	Start time of the eddystore job made with the runJob function

Value

A data frame of the concatenated essential output files from each job.

`makeDateIntervals` *Make Date Intervals*

Description

This function creates a number of equally sized time intervals between the specified start and end times.

Usage

```
makeDateIntervals(startDate_period, endDate_period, nIntervals)
```

Arguments

`startDate_period`	A character string for the time the run was started, used to identify the relevant files.
`endDate_period`	A character string for the time the run was started, used to identify the relevant files.
`nIntervals`	An integer number of intervals to use to split the processing into.

Value

An object with vectors of start and end dates for n intervals .

See Also

`adjustIntervals` for the adjusting this to match boundaries between .eddypro project files.

Examples

```
nIntervals = 4
startDate_period <- "2006-07-01 00:00"
endDate_period   <- "2007-12-31 23:30"
startDate_period <- as.POSIXct(strptime(startDate_period, "%Y-%m-%d %H:%M"), tz = "UTC")
endDate_period   <- as.POSIXct(strptime(endDate_period,   "%Y-%m-%d %H:%M"), tz = "UTC")
myJob <- makeDateIntervals(startDate_period, endDate_period, nIntervals)
```

`read_full_output` *Collate full output files - but not yet working for collating multiple job files*

Description

"full output" files are harder to read - header is on line 2, data on line 4: use readr::read_csv instead: This function collates all the full output files. This function collates all the full output files.

Usage

```
read_full_output(fname)
```

Arguments

fname	A character string for the time the run was started, used to identify the relevant files.

Value

A data frame of the concatenated full output files from each job.

---

runJob *Run Processing Job*

Description

This function runs an eddypro processing job on LOTUS.

Usage

```
runJob(job)
```

Arguments

job	An eddystore job object made with the createJob function

Value

An eddystore job object.

See Also

adjustIntervals for the adjusting this to match boundaries between project files.

Examples

```
stationID_proc <- "EasterBush"
procID_proc <- "CO2_H2O"
nIntervals <- 4
startDate_period <- "2006-07-01 00:00"
endDate_period   <- "2007-12-31 23:30"
startDate_period <- as.POSIXct(strptime(startDate_period, "%Y-%m-%d %H:%M"), tz = "UTC")
endDate_period   <- as.POSIXct(strptime(endDate_period,   "%Y-%m-%d %H:%M"), tz = "UTC")
intervals <- makeDateIntervals(startDate_period, endDate_period, nIntervals)
fname_df_project = "C:/Users/plevy/Documents/eddystore_projects/df_eddystore_projects.csv"
myJob <- adjustIntervals(stationID_proc, procID_proc, intervals, fname_df_project = fname_df_project)
myJob <- writeProjectFiles(myJob)
myJob <- writeJobFile(myJob, binpath = "N:/0Peter/curr/ECsystem/eddypro",
                             switch_OS = "-s linux",
                             eddystore_path = "N:/0Peter/curr/ECsystem/eddystore")
myJob <- runJob(myJob)
```

writeJobFile *Write LOTUS batch job files for all intervals*

Description

This function writes a LOTUS batch job file for each of the intervals specified.

Usage

```
writeJobFile(job,
 binpath = "/gws/nopw/j04/eddystore/eddypro-engine_6.2.0/eddypro-engine/bin/linux/eddypro_rp",
  switch_OS = "-s linux", eddystore_path = "/gws/nopw/j04/eddystore",
  user_email = "plevy@ceh.ac.uk")
```

Arguments

job	An eddystore job object made with the writeProjectFiles function.
binpath	Path to the eddy_rp raw data processing binary file.
switch_OS	Switch betweeen linux and windows versions.
eddystore_path	The path to eddystore.
user_email	E-mail address to send notifications to.

Value

An eddystore job object.

See Also

adjustIntervals for the adjusting this to match boundaries between processing files.

Examples

```
stationID_proc <- "EasterBush"
procID_proc <- "CO2_H2O"
nIntervals <- 4
startDate_period <- "2006-07-01 00:00"
endDate_period   <- "2007-12-31 23:30"
startDate_period <- as.POSIXct(strptime(startDate_period, "%Y-%m-%d %H:%M"), tz = "UTC")
endDate_period   <- as.POSIXct(strptime(endDate_period,   "%Y-%m-%d %H:%M"), tz = "UTC")
intervals <- makeDateIntervals(startDate_period, endDate_period, nIntervals)
fname_df_project = "C:/Users/plevy/Documents/eddystore_projects/df_eddystore_projects.csv"
myJob <- adjustIntervals(stationID_proc, procID_proc, intervals, fname_df_project = fname_df_project)
myJob <- writeProjectFiles(myJob)
myJob <- writeJobFile(myJob, binpath = "N:/0Peter/curr/ECsystem/eddypro",
                             switch_OS = "-s linux",
                             eddystore_path = "N:/0Peter/curr/ECsystem/eddystore")
```

writeProjectFiles *Write Eddypro Project files for all intervals*

Description

This function writes an eddypro project file for each of the intervals specified.

Usage

```
writeProjectFiles(job, job_name = "eddytest")
```

Arguments

job	A list of the start and end times of each interval, created by makeDateIntervals.
job_name	The job name identifier given in df_job_requests.

Value

An eddystore job object.

See Also

adjustIntervals for the adjusting this to match boundaries between processing files.

Examples

```
stationID_proc <- "EasterBush"
procID_proc <- "CO2_H2O"
nIntervals <- 4
startDate_period <- "2006-07-01 00:00"
endDate_period   <- "2007-12-31 23:30"
startDate_period <- as.POSIXct(strptime(startDate_period, "%Y-%m-%d %H:%M"), tz = "UTC")
endDate_period   <- as.POSIXct(strptime(endDate_period,   "%Y-%m-%d %H:%M"), tz = "UTC")
intervals <- makeDateIntervals(startDate_period, endDate_period, nIntervals)
fname_df_project = "C:/Users/plevy/Documents/eddystore_projects/df_eddystore_projects.csv"
myJob <- adjustIntervals(stationID_proc, procID_proc, intervals, fname_df_project = fname_df_project)
myJob <- writeProjectFiles(myJob)
```

Index