Package: PEcAn.workflow (via r-universe)

June 27, 2024

Type Package

Title PEcAn Functions Used for Ecological Forecasts and Reanalysis

Version 1.7.2

Date 2021-10-04

Author David LeBauer, Mike Dietze, Xiaohui Feng, Dan Wang, Mike Dietze, Carl Davidson, Rob Kooper, Shawn Serbin

Maintainer David LeBauer <dlebauer@illinois.edu>

Description The Predictive Ecosystem Carbon Analyzer (PEcAn) is a scientific workflow management tool that is designed to simplify the management of model parameterization, execution, and analysis. The goal of PEcAn is to streamline the interaction between data and models, and to improve the efficacy of scientific investigation. This package provides workhorse functions that can be used to run the major steps of a PEcAn analysis.

License BSD_3_clause + file LICENSE

Imports dplyr, PEcAn.data.atmosphere, PEcAn.data.land, PEcAn.DB, PEcAn.logger, PEcAn.remote, PEcAn.settings, PEcAn.uncertainty, PEcAn.utils, purrr (>= 0.2.3), XML

Suggests mockery, testthat, withr

Copyright Authors

Encoding UTF-8

RoxygenNote 7.3.1

Repository https://pecanproject.r-universe.dev

RemoteUrl https://github.com/PecanProject/pecan

RemoteRef HEAD

RemoteSha d5c7bffdf233077968945a182c11240b5d76e42d

Contents

`create_execute_test_xml` *Create a PEcAn XML file and use it to run a PEcAn workflow*

Description

Create a PEcAn XML file and use it to run a PEcAn workflow

Usage

```
create_execute_test_xml(
  model_id,
  met,
  site_id,
  start_date,
  end_date,
  dbfiles_folder,
  user_id,
  output_folder = "batch_test_output",
  pecan_path = getwd(),
  pft = NULL,
  ensemble_size = 1,
  sensitivity_variable = "NPP",
  sensitivity = FALSE,
  db_bety_username = NULL,
  db_bety_password = NULL,
  db_bety_hostname = NULL,
  db_bety_port = NULL,
  db_bety_driver = "Postgres"
)
```

Arguments

`model_id`	(numeric) Model ID (from ‘models‘ table)
`met`	(character) Name of meteorology input source (e.g. ‘"CRUNCEP"‘)
`site_id`	(numeric) Site ID (from ‘sites‘ table)

`start_date` (character or date) Run start date

`end_date` (character or date) Run end date

`dbfiles_folder` (character) Path to ‘dbfiles‘ directory

`user_id` (numeric) User ID to associate with the workflow

`output_folder` (character) Path to root directory for storing outputs. Default = ‘"batch_test_output"‘

`pecan_path` (character) Path to PEcAn source code. Default is current working directory.

`pft` (character) Name of PFT to run. If ‘NULL‘ (default), use the first PFT in BETY associated with the model.

`ensemble_size` (numeric) Number of ensembles to run. Default = 1.

`sensitivity_variable`
(character) Variable for performing sensitivity analysis. Default = ‘"NPP"‘

`sensitivity` (logical) Whether or not to perform a sensitivity analysis (default = ‘FALSE‘)

`db_bety_username`, `db_bety_password`, `db_bety_hostname`, `db_bety_port`
(character) BETY database connection options. Default values for all of these are pulled from ‘<pecan_path>/web/config.php‘.

`db_bety_driver` (character) BETY database connection driver (default = ‘"Postgres"‘)

Value

A list with two entries: * ‘sys‘: Exit value returned by the workflow (0 for sucess). * ‘outdir‘: Path where the workflow results are saved

Author(s)

Alexey Shiklomanov, Tony Gardella

`do_conversions` *do_conversions*

Description

Input conversion workflow

Usage

```
do_conversions(
  settings,
  overwrite.met = FALSE,
  overwrite.fia = FALSE,
  overwrite.ic = FALSE
)
```

Arguments

settings PEcAn settings list

overwrite.met, overwrite.fia, overwrite.ic
logical

Author(s)

Ryan Kelly, Rob Kooper, Betsy Cowdery, Istem Fer

---

model_specific_tags *model_specific_tags*

Description

model_specific_tags

Usage

```
model_specific_tags(settings, model.info)
```

Arguments

settings	pecan xml settings
model.info	model info extracted from bety

Value

updated settings list

---

run.write.configs *Write model-specific run scripts and configuration files*

Description

Generates run scripts and configuration files for all analyses specified in the provided settings. Most of the heavy lifting is done by the write.config.* function for your specific ecosystem model (e.g. write.config.ED2, write.config.SIPNET).

Usage

```
run.write.configs(
  settings,
  write = TRUE,
  ens.sample.method = "uniform",
  posterior.files = rep(NA, length(settings$pfts)),
  overwrite = TRUE
)
```

Arguments

`settings`	a PEcAn settings list
`write`	should the runs be written to the database?
`ens.sample.method`	how to sample the ensemble members('halton' sequence or 'uniform' random)
`posterior.files`	Filenames for posteriors for drawing samples for ensemble and sensitivity analysis (e.g. post.distns.Rdata, or prior.distns.Rdata)
`overwrite`	logical: Replace output files that already exist?

Details

The default value for `posterior.files` is NA, in which case the most recent posterior or prior (in that order) for the workflow is used. When specified, `posterior.files` should be a vector of filenames with one entry for each PFT. Specify filenames with no path; PFT outdirs will be appended. This forces use of only files within this workflow, to avoid confusion.

Value

an updated settings list, which includes ensemble IDs for SA and ensemble analysis

Author(s)

David LeBauer, Shawn Serbin, Ryan Kelly, Mike Dietze

`runModule.get.trait.data`

Get trait data for all PFTs in a settings list

Description

Get trait data for all PFTs in a settings list

Usage

```
runModule.get.trait.data(settings)
```

Arguments

`settings`	PEcAn configuration list. Must have class ‘Settings‘ or ‘MultiSettings‘

runModule.run.write.configs *Generate model-specific run configuration files for one or more PEcAn runs*

Description

Generate model-specific run configuration files for one or more PEcAn runs

Usage

```
runModule.run.write.configs(settings, overwrite = TRUE)
```

Arguments

`settings`	a PEcAn Settings or MultiSettings object
`overwrite`	logical: Replace config files if they already exist?

Value

A modified settings object, invisibly

start_model_runs *Start selected ecosystem model runs within PEcAn workflow*

Description

Start selected ecosystem model runs within PEcAn workflow

Usage

```
start_model_runs(settings, write = TRUE, stop.on.error = TRUE)

runModule_start_model_runs(settings, stop.on.error = TRUE)
```

Arguments

`settings`	pecan settings object
`write`	(logical) Whether or not to write to the database. Default TRUE.
`stop.on.error`	Throw error if _any_ of the runs fails. Default TRUE.

Functions

- `runModule_start_model_runs()`: A lightweight wrapper around ‘start_model_runs‘ that takes ‘write‘ from ‘settings‘ instead of as a separate argument.

Author(s)

Shawn Serbin, Rob Kooper, David LeBauer, Alexey Shiklomanov

Examples

```
## Not run:
  start_model_runs(settings)

## End(Not run)
```

Index