Package: PEcAn.benchmark (via r-universe)

November 4, 2024

Type Package

Title PEcAn Functions Used for Benchmarking

Version 1.7.3.9000

Author Michael Dietze, David LeBauer, Rob Kooper, Toni Viskari

Maintainer Mike Dietze <dietze@bu.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.

Imports dplyr, ggplot2, gridExtra, lubridate (>= 1.6.0), magrittr, ncdf4 (>= 1.15), PEcAn.DB, PEcAn.logger, PEcAn.settings, PEcAn.utils, reshape2, rlang, SimilarityMeasures, stringr, tidyselect, units, utils, grDevices, XML (>= 3.98-1.4), zoo

Suggests PEcAn.data.land, testthat (>= 2.0.0)

License BSD_3_clause + file LICENSE

Copyright Authors

LazyLoad yes

LazyData FALSE

Encoding UTF-8

RoxygenNote 7.3.2

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

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

RemoteRef HEAD

RemoteSha caad7b3f8386df43eaf44f9316459f66ffc69b0b

Contents

add_workflow_info *Add workflow specific info to settings list for benchmarking*

Description

Add workflow specific info to settings list for benchmarking

Usage

```
add_workflow_info(settings, bety)
```

Arguments

settings	settings or multisettings object
bety	connection to the database

Author(s)

Betsy Cowdery

align_by_first_observation *align_first_observation*

Description

align_first_observation

Usage

```
align_by_first_observation(observation_one, observation_two, custom_table)
```

Arguments

observation_one	a vector of plant functional types, or species. Provides species/pft names.
observation_two	another vector of plant functional types, or species. Provides the order.
custom_table	a table that either maps two pft's to one another or maps custom species codes to bety id codes. In the second case, must be passable to match_species_id.

Value

vector Returns a vector of PFT's/species from observation_one that matches the order of observation_two

Author(s)

Tempest McCabe

Examples

```
observation_one<-c("AMCA3","AMCA3","AMCA3","AMCA3")
observation_two<-c("a", "b", "a", "a")

table<-list()
table$plant_functional_type_one<- c("AMCA3","AMCA3","ARHY", "ARHY")
table$plant_functional_type_two<- c('a','a','b', 'b') # PFT groupings
table<-as.data.frame(table)

aligned <- align_by_first_observation(
  observation_one = observation_one,
  observation_two = observation_two,
  custom_table = table)

# aligned should be a vector '[1] "AMCA3" "ARHY"  "AMCA3" "AMCA3"'
```

---

align_data *Align timeseries data*

Description

Align timeseries data

Usage

```
align_data(model.calc, obvs.calc, var, align_method = "match_timestep")
```

Arguments

model.calc	data.frame
obvs.calc	data.frame
var	data.frame
align_method	name of function to use for alignment

Value

dat

Author(s)

Betsy Cowdery

align_data_to_data_pft *align_data_to_data_pft*

Description

align_data_to_data_pft

Usage

```
align_data_to_data_pft(
  con,
  observation_one,
  observation_two,
  custom_table = NULL,
  format_one,
  format_two,
  subset_is_ok = FALSE
)
```

Arguments

con	database connection
observation_one	a vector of plant functional types, or species
observation_two	another vector of plant functional types, or species
custom_table	a table that either maps two pft's to one another or maps custom species codes to bety id codes. In the second case, must be passable to match_species_id.
format_one	The output of query.format.vars() of observation one of the form output$vars$bety_names
format_two	The output of query.format.vars() of observation two of the form output$vars$bety_names
subset_is_ok	When aligning two species lists, this allows for alignment when species lists aren't identical. set to FALSE by default.

Details

Aligns vectors of Plant Fucntional Typed and species. Can align: - two vectors of plant functional types (pft's) if a custom map is provided - a list of species (usda, fia, or latin_name format) to a plant functional type - a list of species in a custom format, with a table mapping it to bety_species_id's

Will return a list of what was originally provided, bety_species_codes if possible, and an aligned output. Because some alignement is order-sensitive, alignment based on observation_one and observation_two are both provided.

Value

`list` containing the following columns:

`$original` Will spit back out original vectors pre-alignment

`$aligned$aligned_by_observation_one` Where possible, will return a vector of observation_one pft's/species in the order of observation_two

`species` Where possible, will return a vector of observation_two's pft's/species in the order of observation_one

`$bety_species_id` Where possible, will return the bety_species_id's for one or both observations

`$bety_species_intersection` Where possible, will return the intersection of two aligned lists of species. subset_is_ok must be set to TRUE.

Author(s)

Tempest McCabe

Examples

```
## Not run:

observation_one<-c("AMCA3","AMCA3","AMCA3","AMCA3")
observation_two<-c("a", "b", "a", "a")

table<-list()
table$plant_functional_type_one<- c("AMCA3","AMCA3","ARHY", "ARHY")
table$plant_functional_type_two<- c('a','a','b', 'b') # PFT groupings
table<-as.data.frame(table)

format_one<-"species_USDA_symbol"
format_two<-"plant_functional_type"

aligned <- align_data_to_data_pft(
 con = con,
 observation_one = observation_one, observation_two = observation_two,
 format_one = format_one, format_two = format_two,
 custom_table = table)

## End(Not run)
```

---

align_pft *Align vectors of Plant Functional Type and species.*

Description

Align vectors of Plant Functional Type and species.

Usage

```
align_pft(
  con,
  observation_one,
  observation_two,
  custom_table = NULL,
  format_one,
  format_two,
  subset_is_ok = FALSE,
  comparison_type = "data_to_data",
  ...
)
```

Arguments

con	database connection
observation_one	a vector of plant fucntional types, or species
observation_two	anouther vector of plant fucntional types, or species
custom_table	a table that either maps two pft's to one anouther or maps custom species codes to bety id codes. In the second case, must be passable to match_species_id.
format_one	The output of query.format.vars() of observation one of the form output$vars$bety_names
format_two	The output of query.format.vars() of observation two of the form output$vars$bety_names
subset_is_ok	When aligning two species lists, this allows for alignement when species lists aren't identical. set to FALSE by default.
comparison_type	one of "data_to_model", "data_to_data", or "model_to_model"
...	other arguments, currently ignored

Details

Can align: - two vectors of plant fucntional types (pft's) if a custom map is provided - a list of species (usda, fia, or latin_name format) to a plant fucntional type - a list of species in a custom format, with a table mapping it to bety_species_id's

Will return a list of what was originally provided, bety_speceis_codes if possible, and an aligned output. Becuase some alignement is order-sensitive, alignment based on observation_one and observation_two are both provided.

comparison_type can be one of the following:

data_to_data Will align lists of pfts and species. Must be assosiated with inputs.

data_to_model Not yet implemented

model_to_model Not yet implemented

Value

`list` containing the following columns:

`$original` Will spit back out original vectors pre-alignment

`$aligned$aligned_by_observation_one` Where possible, will return a vector of observation_one pft's/species in the order of observation_two

`species` Where possible, will return a vector of observation_two's pft's/species in the order of observation_one

`$bety_species_id` Where possible, will return the bety_species_id's for one or both observations

Author(s)

Tempest McCabe

Examples

```
## Not run:


#------------ A species to PFT alignment -----------
observation_one<-c("AMCA3","AMCA3","AMCA3","AMCA3")
observation_two<-c("a", "b", "a", "a") #

format_one<-"species_USDA_symbol"
format_two<-"plant_funtional_type"

table<-list()
table$plant_functional_type_one<- c("AMCA3","AMCA3","ARHY", "ARHY")
table$plant_functional_type_two<- c('a','a','b', 'b') # PFT groupings
table<-as.data.frame(table)


aligned<-align_pft(con = con, observation_one = observation_one, observation_two = observation_two,
format_one = format_one, format_two = format_two, custom_table = table)

## End(Not run)
```

---

bm_settings2pecan_settings

Move benchmarking settings back in to original pecan settings object

---

Description

Move benchmarking settings back in to original pecan settings object

Usage

```
bm_settings2pecan_settings(bm.settings)
```

Arguments

bm.settings	settings or multisettings object

Author(s)

Betsy Cowdery

---

calc_benchmark *Calculate benchmarking statistics*

Description

For each benchmark id, calculate metrics and update benchmarks_ensemble_scores

Usage

```
calc_benchmark(settings, bety, start_year = NA, end_year = NA)
```

Arguments

settings	settings object describing the run to calculate
bety	database connection
start_year, end_year	time range to read. If NA, these are taken from 'settings'

Author(s)

Betsy Cowdery

---

calc_metrics *calc_metrics*

Description

calc_metrics

Usage

```
calc_metrics(model.calc, obvs.calc, var, metrics, ensemble.id, bm_dir)
```

Arguments

model.calc	model data
obvs.calc	observational data
var	variables to be used
metrics	metrics to be used
ensemble.id	id of ensemble run
bm_dir	directory where benchmarking outputs will be saved

Author(s)

Betsy Cowdery

---

check_BRR *Check whether a run has been registered as a reference run in BETY*

Description

Check whether a run has been registered as a reference run in BETY

Usage

```
check_BRR(settings_xml, con)
```

Arguments

settings_xml	cleaned settings to be compared with BRR in the database
con	database connection

Author(s)

Betsy Cowdery

check_if_legal_table *check_if_legal_table*

Description

check_if_legal_table

Usage

```
check_if_legal_table(table, observation_one, observation_two)
```

Arguments

table	a table that either maps two pft's to one anouther or maps custom species codes to bety id codes. In the second case, must be passable to match_species_id.
observation_one	a vector of plant functional types, or species
observation_two	anouther vector of plant functional types, or species

Details

Checks if custom_table: 1. is formated correctly 2. is complete (has all of the species/pft's in both observations) 3. is condense-able (Could be represented as a hierachry)

Value

boolean

Author(s)

Tempest McCabe

check_if_list_of_pfts *check_if_list_of_pfts*

Description

Checks if format contains a variable named "plant_functional_type"

Usage

```
check_if_list_of_pfts(vars)
```

Arguments

vars	names to check

Value

boolean

Author(s)

Tempest McCabe

---

check_if_species_list *check_if_species_list*

Description

check_if_species_list

Usage

```
check_if_species_list(vars, custom_table = NULL)
```

Arguments

vars	format
custom_table	a table that either maps two pft's to one anouther or maps custom species codes to bety id codes. In the second case, must be passable to match_species_id.

Details

Checks if format contains a species list in a known format, or a declared custom format.

Value

boolean

Author(s)

Tempest McCabe

clean_settings_BRR *Cleans PEcAn settings file and prepares the settings to be saved in a reference run record in BETY*

Description

Cleans PEcAn settings file and prepares the settings to be saved in a reference run record in BETY

Usage

```
clean_settings_BRR(inputfile)
```

Arguments

inputfile	the PEcAn settings file to be used.

Author(s)

Betsy Cowdery

create_BRR *Create benchmark reference run and ensemble*

Description

For each benchmark id, calculate metrics and update benchmarks_ensemble_scores

Usage

```
create_BRR(ens_wf, con, user_id = "")
```

Arguments

ens_wf	table made from joining ensemble and workflow tables
con	database connection
user_id	Optional user id to use for this record in reference_runs table

Author(s)

Betsy Cowdery

define_benchmark *Benchmark Definition: Retrieve or Create Bety Benchmarking Records*

Description

Creates records for benchmarks, benchmarks_benchmarks_reference_runs, benchmarks_metrics

Usage

```
define_benchmark(settings, bety)
```

Arguments

settings	settings list
bety	database connection

Value

updated settings list

Author(s)

Betsy Cowdery

format_wide2long *Function to convert wide format to long format*

Description

Function to convert wide format to long format

Usage

```
format_wide2long(out, format, vars_used, time.row)
```

Arguments

out	wide format data
format	as returned by query.format.vars
vars_used	data frame mapping ‘input_name‘ to ‘bety_name‘
time.row	ignored; value in output is set from ‘format$vars$storage_type‘

Value

list of updated values

Author(s)

Istem Fer

`get_species_list_standard` *get_species_list_standard*

Description

Returns the format type for convience of use with match_species_id

Usage

```
get_species_list_standard(vars)
```

Arguments

`vars`	format to be matched

Value

`character` Returns "usda", "latin_name", "fia" or "custom"

Author(s)

Tempest McCabe

`load_csv` *load_csv*

Description

load_csv

Usage

```
load_csv(data.path, format, site, vars = NULL)
```

Arguments

`data.path`	character
`format`	list
`site`	list
`vars`	column names to return. If NULL, returns all columns

Author(s)

Betsy Cowdery

load_data *load data*

Description

Generic function to convert input files containing observational data to a common PEcAn format.

Usage

```
load_data(
  data.path,
  format,
  start_year = NA,
  end_year = NA,
  site = NA,
  vars.used.index = NULL,
  ...
)
```

Arguments

data.path	character
format	list
start_year	numeric
end_year	numeric
site	list
vars.used.index	which variables to use? If NULL, these are taken from 'format'
...	further arguments, currently ignored

Author(s)

Betsy Cowdery, Istem Fer, Joshua Mantooth

load_rds *load_rds*

Description

load_rds

Usage

```
load_rds(data.path, format, site, vars = NULL)
```

Arguments

data.path	character
format	list, not used, for compatibility
site	not used, for compatibility
vars	optional variable names to load. if NULL, returns all variables in file

Author(s)

Istem Fer

---

load_tab_separated_values

Load files with mime-type 'text/tab-separated-values'

Description

Load files with mime-type 'text/tab-separated-values'

Usage

```
load_tab_separated_values(data.path, format, site = NULL, vars = NULL)
```

Arguments

data.path	character
format	list
site	list
vars	variable names to load. If NULL, loads all columns

Author(s)

Betsy Cowdery, Mike Dietze

load_x_netcdf *Load from netCDF*

Description

Load from netCDF

Usage

```
load_x_netcdf(data.path, format, site, vars = NULL)
```

Arguments

data.path	character vector or list
format	list
site	list
vars	character

Author(s)

Istem Fer

match_timestep *Match time step*

Description

Match time step

Usage

```
match_timestep(date.coarse, date.fine, data.fine)
```

Arguments

date.coarse	numeric
date.fine	numeric
data.fine	matrix

Author(s)

Istem Fer

`mean_over_larger_timestep` *Calculate benchmarking statistics*

Description

Calculate benchmarking statistics

Usage

```
mean_over_larger_timestep(date.coarse, date.fine, data.fine)
```

Arguments

`date.coarse`	numeric
`date.fine`	numeric
`data.fine`	data.frame

Author(s)

Betsy Cowdery, Michael Dietze

`metric_AME` *Absolute Maximum Error*

Description

Absolute Maximum Error

Usage

```
metric_AME(dat, ...)
```

Arguments

`dat`	dataframe
`...`	ignored

Author(s)

Betsy Cowdery

metric_cor *Correlation Coefficient*

Description

Correlation Coefficient

Usage

```
metric_cor(dat, ...)
```

Arguments

dat	dataframe
...	ignored

Author(s)

Mike Dietze

metric_Frechet *Frechet Distance*

Description

Frechet Distance

Usage

```
metric_Frechet(metric_dat, ...)
```

Arguments

metric_dat	dataframe
...	ignored

Author(s)

Betsy Cowdery

metric_lmDiag_plot *Linear Regression Diagnostic Plot*

Description

Linear Regression Diagnostic Plot

Usage

```
metric_lmDiag_plot(metric_dat, var, filename = NA, draw.plot = FALSE)
```

Arguments

metric_dat	data.frame
var	ignored
filename	path to save plot, or NA to not save
draw.plot	logical: return plot object?

Author(s)

Betsy Cowdery

metric_MAE *Mean Absolute Error*

Description

Mean Absolute Error

Usage

```
metric_MAE(dat, ...)
```

Arguments

dat	dataframe
...	ignored

Author(s)

Betsy Cowdery

metric_MSE *Mean Square Error*

Description

Mean Square Error

Usage

```
metric_MSE(dat, ...)
```

Arguments

dat	dataframe
...	ignored

Author(s)

Betsy Cowdery

metric_PPMC *Pearson Product Moment Correlation*

Description

Pearson Product Moment Correlation

Usage

```
metric_PPMC(metric_dat, ...)
```

Arguments

metric_dat	dataframe
...	ignored

Author(s)

Betsy Cowdery

metric_R2 *Coefficient of Determination (R2)*

Description

Coefficient of Determination (R2)

Usage

```
metric_R2(metric_dat, ...)
```

Arguments

metric_dat	dataframe
...	ignored

Author(s)

Betsy Cowdery

metric_RAE *Relative Absolute Error*

Description

Relative Absolute Error

Usage

```
metric_RAE(metric_dat, ...)
```

Arguments

metric_dat	dataframe
...	ignored

Author(s)

Betsy Cowdery

metric_residual_plot *Residual Plot*

Description

Residual Plot

Usage

```
metric_residual_plot(
  metric_dat,
  var,
  filename = NA,
  draw.plot = is.na(filename)
)
```

Arguments

metric_dat	dataframe to plot, with at least columns 'time', 'model', 'obvs'
var	variable name, used as plot title
filename	path to save plot, or NA to not save
draw.plot	logical: Return the plot object?

Author(s)

Betsy Cowdery

metric_RMSE *Root Mean Square Error*

Description

Root Mean Square Error

Usage

```
metric_RMSE(dat, ...)
```

Arguments

dat	dataframe
...	ignored

Author(s)

Betsy Cowdery

metric_run *Model Run Check*

Description

Model Run Check

Usage

```
metric_run(settings)
```

Arguments

settings	list

Author(s)

Betsy Cowdery

metric_scatter_plot *Scatter Plot*

Description

Scatter Plot

Usage

```
metric_scatter_plot(
  metric_dat,
  var,
  filename = NA,
  draw.plot = is.na(filename)
)
```

Arguments

metric_dat	dataframe to plot, with at least columns ‘model‘ and ‘obvs‘
var	ignored
filename	path to save plot, or NA to not save
draw.plot	logical: Return the plot object?

Author(s)

Betsy Cowdery

metric_timeseries_plot *Timeseries Plot*

Description

Timeseries Plot

Usage

```
metric_timeseries_plot(
  metric_dat,
  var,
  filename = NA,
  draw.plot = is.na(filename)
)
```

Arguments

metric_dat	dataframe to plot, with at least columns 'time', 'model', 'obvs'
var	variable name, used as plot title
filename	path to save plot, or NA to not save
draw.plot	logical: Return the plot object?

Author(s)

Betsy Cowdery

read_settings_BRR *Read settings from database using reference run id*

Description

For each benchmark entry in a (multi)settings object, get run settings using reference run id and add to the settings object

Usage

```
read_settings_BRR(settings)
```

Arguments

settings	settings or multisettings object

Author(s)

Betsy Cowdery

Index