

PEcAn Workflow

Aritra Dey

Load PEcAn packages

```
library("PEcAn.all")
```

Loading required package: PEcAn.DB

Loading required package: PEcAn.settings

Loading required package: PEcAn.MA

Loading required package: PEcAn.logger

Loading required package: PEcAn.utils

Loading required package: PEcAn.uncertainty

Loading required package: PEcAn.data.atmosphere

Loading required package: PEcAn.data.land

Loading required package: PEcAn.data.remote

Loading required package: PEcAn.assim.batch

Loading required package: PEcAn.emulator

Loading required package: PEcAn.priors

Loading required package: PEcAn.benchmark

Loading required package: PEcAn.remote

Loading required package: PEcAn.workflow

Load PEcAn settings files.

- If you have pecan.xml generated from web, place the pecan.xml file in your directory and copy its file path. Assign this path to the **settings_path** variable.

```
settings_path <- "~/pecan.xml"
settings <- PEcAn.settings::read.settings(settings_path)
```

Prepare and Validate Settings

PEcAn provides utilities to process and validate settings before execution, ensuring that all required fields are correctly configured.

```
settings <- PEcAn.settings::prepare.settings(settings)
```

Warning: <PostgreSQLConnection> uses an old dbplyr interface
i Please install a newer version of the package or contact the maintainer
This warning is displayed once every 8 hours.

Convert Settings for Ecosystem Model

Converts settings into the format required by the selected ecosystem model

```
settings <- PEcAn.workflow::do_conversions(settings)
```

Warning in postgresqlExecStatement(conn, statement, ...): RPostgreSQL warning:
unrecognized PostgreSQL field type geometry (id:20608) in column 17

```

      id site_id notes      created_at      updated_at start_date
1 1000011238    772    2017-03-23 18:05:51 2017-11-01 20:41:40 1998-01-01
  end_date      name parent_id user_id access_level raw
1 2016-12-31 AmerifluxLBL_site_0-772      NA      NA      4  NA
  format_id
1      5e+09
      id site_id notes      created_at      updated_at start_date
1 9.9e+10    772    2025-05-20 17:50:52 2025-05-20 17:50:52 2004-01-01
  end_date      name parent_id user_id access_level raw
1 2006-12-31 AmerifluxLBL_CF_site_0-772 1000011238      NA      4  NA
  format_id
1      33
      id site_id notes      created_at      updated_at start_date
1 9.9e+10    772    2025-05-20 17:51:11 2025-05-20 17:51:11 2004-01-01
  end_date      name parent_id user_id access_level
1 2006-12-31 AmerifluxLBL_CF_gapfill_site_0-772 9.9e+10      NA      4
  raw format_id
1 NA      33
$input.id
[1] 9.9e+10

$dbfile.id
[1] 9.9e+10

[1] "Convert to model format"
      id site_id notes      created_at      updated_at start_date
1 9.9e+10    772    2025-05-20 17:51:13 2025-05-20 17:51:13 2004-01-01
  end_date      name parent_id user_id access_level raw
1 2006-12-31 AmerifluxLBL_SIPNET_site_0-772 9.9e+10      NA      4  NA
  format_id
1      24

```

Trait and Meta Analysis

Retrieve trait data and generate probabilistic model parameter distributions.

```

# Retrieve trait data and prior distributions for the specified Plant Functional Types (PFTs)
settings <- PEcAn.workflow::runModule.get.trait.data(settings)
# Perform meta-analysis to derive probabilistic distributions for model parameters
PEcAn.MA::runModule.run.meta.analysis(settings)

```

```
# Save the updated settings, including the retrieved trait data, to an XML file
PEcAn.settings::write.settings(settings, outputfile = "pecan.TRAIT.xml")
```

```
[1] "/data/workflows/PEcAn_99000000049/pecan.TRAIT.xml"
```

Write Model Configuration Files

Generate model configuration files before simulation runs.

```
settings <-PEcAn.workflow::runModule.run.write.configs(settings)
```

Loading required package: PEcAn.SIPNET

```
PEcAn.settings::write.settings(settings, outputfile = "pecan.CONFIGS.xml")
```

```
[1] "/data/workflows/PEcAn_99000000049/pecan.CONFIGS.xml"
```

Run Model Simulations and Fetch Results

Start model simulations and retrieve output for analysis.

```
PEcAn.workflow::start_model_runs(settings)
```

```
|
|
|
|=====| 100%
```

```
runModule.get.results(settings)
```

Explore Available Variables and Dimensions in NetCDF File

```

library(ncdf4)
# Use settings$outdir to build base path
workflow_outdir <- settings$outdir # Correct path from pecan.xml <outdir>

# Default run_id for demo - you can replace this with your actual run ID
# Tip: You can find your run ID from the "run-model" chunk output during execution
run_id <- "99000000195"

#Select year to inspect
year <- 2006
#Build NetCDF file path
netcdf_file <- file.path(workflow_outdir, "out", run_id, paste0(year, ".nc"))

#Open NetCDF file and show variable and dimension names
nc <- nc_open(netcdf_file)
cat("Variables:\n")

```

Variables:

```
print(names(nc$var))
```

```

[1] "GPP" "NPP"
[3] "TotalResp" "AutoResp"
[5] "HeteroResp" "SoilResp"
[7] "NEE" "AbvGrndWood"
[9] "leaf_carbon_content" "TotLivBiom"
[11] "TotSoilCarb" "Qle"
[13] "Transp" "SoilMoist"
[15] "SoilMoistFrac" "SWE"
[17] "litter_carbon_content" "litter_mass_content_of_water"
[19] "LAI" "fine_root_carbon_content"
[21] "coarse_root_carbon_content" "GWBI"
[23] "AGB" "time_bounds"

```

```
cat("\n Dimensions:\n")
```

Dimensions:

```
print(names(nc$dim))
```

```
[1] "lon"          "lat"          "time"         "hist_interval"
```

```
nc_close(nc)
```

Plot NetCDF Variables Using Selected Data and Custom Parameters

```
# This chunk generates a NetCDF variable plot dynamically
# Make sure `settings` object is loaded from pecan.xml before running this
# Required user inputs (customize these):
# - year: choose a year to visualize
# - xvar, yvar: choose variables from the **Variables** printed in the previous chunk (`get`)
year      <- 2006
xvar      <- 'coarse_root_carbon_content' # Select valid variable from previous chunk output
yvar      <- 'LAI' # Select valid variable from previous chunk output
width     <- 800
height    <- 600
filename  <- "plot.png"

workflow_outdir <- settings$outdir
run_id          <- "99000000195" # Update this from "run-model" output
netcdf_file     <- file.path(workflow_outdir, "out", run_id, paste0(year, ".nc"))

PEcAn.visualization::plot_netcdf(
  datafile  = netcdf_file,
  yvar      = yvar,
  xvar      = xvar,
  width     = width,
  height    = height,
  filename  = filename,
  year      = year
)
```

```
Warning in png::readPNG(system.file("favicon.png", package =
"PEcAn.visualization")): libpng warning: iCCP: known incorrect sRGB profile
```

pdf
2

```
knitr::include_graphics(filename)
```

