

Package: PEcAn.uncertainty (via r-universe)

October 16, 2026

Type Package

Title PEcAn Functions Used for Propagating and Partitioning
Uncertainties in Ecological Forecasts and Reanalysis

Version 1.9.0.9000

Author David LeBauer, Mike Dietze, Xiaohui Feng, Dan Wang, 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.

URL <https://pecanproject.github.io>

BugReports <https://github.com/PecanProject/pecan/issues>

Imports coda, dplyr, ggplot2, gridExtra, PEcAn.DB, PEcAn.emulator,
PEcAn.logger, PEcAn.priors, PEcAn.settings, PEcAn.utils, plyr
(>= 1.8.4), purrr, randtoolbox, rlang, sensitivity, tidyr

Suggests mockery, testthat (>= 1.0.2), withr

License BSD_3_clause + file LICENSE

Copyright Authors

LazyLoad yes

LazyData FALSE

Encoding UTF-8

RoxygenNote 7.3.3

X-schema.org-keywords ecological-forecasting, sensitivity-analysis,
PEcAn, uncertainty-propagation

Config/pak/sysreqs cmake make jags libicu-dev libuv1-dev libxml2-dev libnetcdf-dev libssl-
dev libudunits2-dev zlib1g-dev

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

Date/Publication 2026-10-16 08:17:39 UTC

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

RemoteRef HEAD

RemoteSha fdb9b89d6133f78473728f63b7c1b3deb4ee29ec

RemoteSubdir modules/uncertainty

Contents

compute_sobol_indices	3
ensemble.filename	3
ensemble.ts	5
flux.uncertainty	5
generate_joint_ensemble_design	6
generate_OAT_SA_design	7
get.change	9
get.coef.var	10
get.elasticity	10
get.ensemble.samples	11
get.gi.phii	12
get.parameter.samples	12
get.results	14
get.sensitivity	15
get_parameter_samples	16
input.ens.gen	18
kurtosis	19
load_pft_posteriors	20
plot_flux_uncertainty	21
plot_sensitivities	21
plot_sensitivity	22
plot_variance_decomposition	23
read.ameriflux.L2	24
read.ensemble.output	24
read.ensemble.ts	25
read.sa.output	26
run.ensemble.analysis	27
run.sensitivity.analysis	28
runModule.get.results	30
runModule.run.ensemble.analysis	30
runModule.run.sensitivity.analysis	31
sa.splinefun	31
sa_run_descriptions	32
sa_run_id_table	32
sa_run_samples	33
sd.var	34
sensitivity.analysis	34
sensitivity.filename	35

<i>compute_sobol_indices</i>	3
spline.ensemble	36
spline.truncate	37
variance.stats	38
write.ensemble.configs	38
Index	41

compute_sobol_indices	<i>Compute Sobol indices from a finished PEcAn run</i>
-----------------------	--

Description

Loads model outputs from a Sobol ensemble, calculates summary statistics for a chosen variable, feeds them to `sensitivity::tell()`, and returns the updated Sobol object.

Usage

```
compute_sobol_indices(outdir, sobol_obj, var = "GPP", stat_fun = mean)
```

Arguments

outdir	PEcAn run output directory that contains runs.txt
sobol_obj	object produced by <code>PEcAn.uncertainty::generate_joint_ensemble_design()</code>
var	Variable name to summarise (default "GPP").
stat_fun	Summary statistic applied to var default mean .

Value

sobol_obj .

ensemble.filename	<i>Generate ensemble filenames</i>
-------------------	------------------------------------

Description

Generates a vector of filenames to be used for PEcAn ensemble output files. All paths start from directory 'settings\$outdir', which will be created if it does not exist.

Usage

```
ensemble.filename(  
  settings,  
  prefix = "ensemble.samples",  
  suffix = "Rdata",  
  all.var.yr = TRUE,  
  ensemble.id = settings$ensemble$ensemble.id,  
  variable = settings$ensemble$variable,  
  start.year = settings$ensemble$start.year,  
  end.year = settings$ensemble$end.year  
)
```

Arguments

settings	list of PEcAn settings.
prefix	string to appear at the beginning of the filename
suffix	file extension: string to appear at the end of the filename
all.var.yr	logical: does ensemble include all vars and years? If FALSE, filename will include years and vars
ensemble.id	ensemble ID(s)
variable	variable(s) included in the ensemble.
start.year, end.year	first and last year simulated.

Details

Typically used by passing only a settings object, but all values can be overridden for manual use.

If only a single variable or a subset of years are needed, the generated filename will identify these in the form If all vars and years are included, set 'all.yr.var' to TRUE to get a filename of the form 'prefix.ensemble_id.suffix'. All elements are recycled vectorwise.

Value

a vector of filenames, each in the form '[settings\$outdir]/[prefix].[ensemble.ID].[variable].[start.year].[end.year][suffix]'.

Author(s)

Ryan Kelly

ensemble.ts	<i>Plots an ensemble time-series from PEcAn for the selected target variable</i>
-------------	--

Description

Plots an ensemble time-series from PEcAn for the selected target variable

Usage

```
ensemble.ts(ensemble.ts, observations = NULL, window = 1, ...)
```

Arguments

ensemble.ts	ensemble timeseries to be plotted, as returned by [read.ensemble.ts()]
observations	observed data to plot over timeseries predictions
window	number of timepoints to average across
...	additional arguments, currently ignored

Value

nothing, generates an ensemble time-series plot

Author(s)

Michael Dietze, Ryan Kelly

flux.uncertainty	<i>Calculate parameters for heteroskedastic flux uncertainty</i>
------------------	--

Description

Calculate parameters for heteroskedastic flux uncertainty

Usage

```
flux.uncertainty(  
  measurement,  
  QC = 0,  
  flags = TRUE,  
  bin.num = 10,  
  transform = identity,  
  minBin = 5,  
  ...  
)
```

Arguments

measurement	= flux time-series
QC	= quality control flag time series (0 = best)
flags	= additional flags on flux filtering of PAIRS (length = 1/2 that of the time series, TRUE = use).
bin.num	= number of bins (default = 10)
transform	= transformation of magnitude (default = identity)
minBin	= minimum number of points allowed in a bin
...	additional arguments, currently ignored

Value

return.list List of parameters from the fitted uncertainty model

Author(s)

Mike Dietze, Carl Davidson

generate_joint_ensemble_design

Generate joint ensemble design for parameter sampling

Description

Creates a joint ensemble design that maintains parameter correlations across all sites in a multi-site run. This function generates sample indices that are shared across sites to ensure consistent parameter sampling.

Usage

```
generate_joint_ensemble_design(
  settings,
  ensemble_size,
  samples = NULL,
  sobol = FALSE
)
```

Arguments

- settings PEcAn settings object. This function directly uses:
- settings\$pfts - List of PFTs (extracts posterior.files)
 - settings\$ensemble\$samplingspace - Input sampling configuration
 - settings\$run\$inputs - Input paths for each input type

	When <code>samples</code> is not supplied, <code>load_pft_posteriors</code> additionally uses <code>settings\$database\$bet</code> and <code>settings\$host\$name</code> for the optional posterior lookup.
<code>ensemble_size</code>	Integer specifying the number of ensemble members. The <code>input_design</code> is generated once for the entire model run. You might want to recycle existing <code>ensemble_samples</code> when splitting larger runs into smaller jobs while keeping the same parameters.
<code>samples</code>	Optional pre-computed parameter samples. When supplied, these are used directly instead of loading posteriors and sampling. When <code>NULL</code> (default), samples are generated in memory via <code>load_pft_posteriors</code> and <code>get_parameter_samples</code> .
<code>sobol</code>	Logical. If <code>TRUE</code> , returns a <code>sensitivity::soboljansen</code> object for Sobol sensitivity analysis.

Details

Parameter samples are drawn in memory via `load_pft_posteriors` and `get_parameter_samples`, or reused when a `samples` bundle is supplied. Nothing is read from or written to `samples.Rdata`.

`load_pft_posteriors` finds and loads the parameter distributions.

In practice it: - uses `pft$posterior.files` directly when it is defined (an `Rdata` file with `post.distns` or `prior.distns`), - otherwise figures out an output directory from `pft$outdir` or, if needed, via `pft$posteriorid` in the database, - then looks in that directory for `post.distns.Rdata`, falling back to `prior.distns.Rdata`, - and, for MCMC posteriors, looks up `trait.mcmc*.Rdata` linked to the same `posteriorid` or a `trait.mcmc.Rdata` file in that directory.

Difference from `generate_OAT_SA_design`: This function samples inputs randomly or quasi-randomly, while `generate_OAT_SA_design` holds all non-parameter inputs constant to isolate parameter effects.

Value

If `sobol = FALSE`, a list with `design_matrix` (the design, one row per run, whose `param` column indexes the samples), `X` (the same matrix under its older name, kept so existing callers keep working), and `samples` (the parameter bundle those indices point into). If `sobol = TRUE`, a `sensitivity::soboljansen()` result object carrying the same `design_matrix` and `samples` entries, with `$X` set by **sensitivity** itself, plus the extra components its index calculations need.

```
generate_OAT_SA_design
```

Generate One-At-a-Time (OAT) input design for sensitivity analysis

Description

Creates an input design matrix for sensitivity analysis where non-parameter inputs (met, IC, soil, etc.) are held constant while parameters vary one-at-a-time across quantiles. This differs from ensemble design, where all inputs vary together.

Usage

```
generate_OAT_SA_design(settings, samples = NULL)
```

Arguments

settings	PEcAn settings object. See details for required elements.
samples	Optional pre-computed parameter samples (a list containing at least <code>sa.samples</code> , as returned by get_parameter_samples). When supplied these are used directly; when NULL (default) they are sampled in memory.

Details

Settings requirements

This function directly uses:

- `settings$pfts` - List of PFTs (extracts `posterior.files`)
- `settings$ensemble$samplingspace` - Input types to include in design
- `settings$sensitivity.analysis$quantiles` - SA quantiles, when sampling here rather than reusing a supplied bundle

When `samples = NULL`, [load_pft_posteriors](#) additionally uses `settings$database$bety` and `settings$host$name` for the optional posterior lookup.

OAT design logic For sensitivity analysis, we must isolate the effect of each parameter by holding all other inputs constant. The `param` column contains sequential indices (1, 2, 3, ...), and `sa_pft`, `sa_trait` and `sa_quantile` say what each run is: the first holds every parameter at its median, and each one after moves a single trait to one of its quantiles. All other columns (`met`, `ic`, `soil`, etc.) are set to 1, meaning the first input file is always used.

Where the samples come from Parameter samples are drawn in memory via [load_pft_posteriors](#) and [get_parameter_samples](#) (mirroring [generate_joint_ensemble_design](#)), or reused when a samples bundle is supplied. Nothing is read from or written to `samples.Rdata`. The design is built from the quantile-based `sa.samples` in that bundle.

Value

A list with `design_matrix`, a data.frame with one row per SA run: the `param` column holds sequential run indices, every input column is held at 1, and `sa_pft`, `sa_trait` and `sa_quantile` say what each run is. The first row is the median run, with those three NA, NA and "50"; every row after moves one trait of one PFT to one of its quantiles. Also `X`, the same matrix under its older name, and `samples`, the parameter bundle used.

Author(s)

Akash B V, Om Kapale

Examples

```
## Not run:
# Generate the SA design, sampling parameters in memory
sa_design <- generate_OAT_SA_design(settings)

# View the design matrix
print(sa_design$design_matrix)
#   param met ic soil
# 1      1      1      1      1 # Median run
# 2      2      1      1      1 # trait1 @ q=2.3%
# 3      3      1      1      1 # trait1 @ q=15.9%
# 4      4      1      1      1 # trait1 @ q=84.1%
# ...

# Reuse an established set of samples instead of drawing new ones
sa_design <- generate_OAT_SA_design(settings, samples = samples)

## End(Not run)
```

get.change

Get delta between sequential flux datapoints

Description

Get delta between sequential flux datapoints

Usage

```
get.change(measurement)
```

Arguments

measurement numeric vector

Value

Difference between consecutive measurements

Author(s)

Mike Dietze, Carl Davidson

get.coef.var	<i>Get coefficient of variance</i>
--------------	------------------------------------

Description

Given a set of numbers (a numeric vector), this returns the set's coefficient of variance.

Usage

```
get.coef.var(set)
```

Arguments

set	numeric vector of trait values
-----	--------------------------------

Value

coefficient of variance

get.elasticity	<i>Generic function for the elasticity</i>
----------------	--

Description

Given the sensitivity, samples, and outputs for a single trait, return elasticity

Usage

```
get.elasticity(sensitivity, samples, outputs)
```

Arguments

sensitivity	univariate sensitivity of model to a parameter, can be calculated by get.sensitivity
samples	samples from trait distribution
outputs	model output from ensemble runs

Value

elasticity = normalized sensitivity

get.ensemble.samples *Get parameter values used in ensemble*

Description

Returns a matrix of randomly or quasi-randomly sampled trait values to be assigned to traits over several model runs. given the number of model runs and a list of sample distributions for traits The model run is indexed first by model run, then by trait

Usage

```
get.ensemble.samples(
  ensemble.size,
  pft.samples,
  env.samples,
  method = "random",
  param.names = NULL,
  ...
)
```

Arguments

ensemble.size	number of runs in model ensemble
pft.samples	random samples from parameter distribution, e.g. from a MCMC chain
env.samples	env samples
method	the method used to generate the ensemble samples. Random generators: uniform, uniform with latin hypercube permutation. Quasi-random generators: halton, sobol, torus. Random generation draws random variates whereas quasi-random generation is deterministic but well equidistributed. Default is uniform. For small ensemble size with relatively large parameter number (e.g ensemble size < 5 and # of traits > 5) use methods other than halton.
param.names	a list of parameter names that were fitted either by MA or PDA, important argument, if NULL parameters will be resampled independently
...	Other arguments passed on to the sampling method

Value

matrix of (quasi-)random samples from trait distributions

Author(s)

David LeBauer, Istem Fer

get.gi.phii	<i>Get $g_i(\phi_i)$</i>
-------------	-------------------------------------

Description

Calculate distribution of function of a variable

Usage

```
get.gi.phii(splinefuncs, trait.samples, maxn = NULL)
```

Arguments

splinefuncs	univariate spline functions created for each trait, e.g. by the sensitivity.analysis function.
trait.samples	n x m matrix (or list with m vectors of length n) of n parameter sets, each with a sample from m traits
maxn	maximum number of parameter sets to evaluate

Details

Transform trait parameter through trait-specific univariate model emulator spline function Given a matrix of parameter sets, calculates a matrix of model output by applying appropriate spline function to each parameter. Output is used with [plot_variance_decomposition](#) and [spline.ensemble](#)

Value

matrix of spline estimates of model output for each of n parameter sets

Author(s)

David LeBauer

get.parameter.samples	<i>Convert priors / MCMC samples to parameter sample chains</i>
-----------------------	---

Description

Loads posterior distributions and MCMC chain results from disk, generates parameter samples for ensemble and sensitivity analysis runs, and optionally saves results to samples.Rdata. This is the backward-compatible wrapper that delegates computation to [get_parameter_samples](#).

Usage

```
get.parameter.samples(
  settings,
  ensemble.size = 1,
  posterior.files = rep(NA, length(settings$pfts)),
  ens.sample.method = "uniform",
  outdir = settings$outdir
)
```

Arguments

settings	PEcAn settings object
ensemble.size	number of runs in model ensemble
posterior.files	list of filenames to read from
ens.sample.method	one of "halton", "sobol", "torus", "lhc", "uniform"
outdir	character path; directory to write samples.Rdata to for provenance. Defaults to settings\$outdir, preserving the legacy always-save behaviour for existing callers (none of which pass it). Pass outdir = NULL to skip the save entirely; the pure get_parameter_samples() never writes to disk regardless.

Details**Upstream contract (reads from each PFT's outdir):**

post.distns.Rdata **or** prior.distns.Rdata Posterior (or prior) distribution summaries produced by run.meta.analysis.pft. A data frame with columns distn, parama, paramb, n.

trait.mcmc.Rdata (Optional) MCMC chain samples from the meta-analysis. Named list of mcmc.list objects, one per trait. If present, samples are drawn from the chains directly; otherwise, independent samples are drawn from post.distns.

File-based side effects (saved to settings\$outdir):

samples.Rdata When outdir is non-NULL (the default), bundles 5 objects:

- trait.samples - Named list (PFT -> trait -> numeric vector of length iterations). Raw MCMC or prior-sampled values.
- sa.samples - Named list (PFT -> matrix[n_quantiles x n_traits]). Quantile-based samples for sensitivity analysis.
- ensemble.samples - Named list (PFT -> data frame[ensemble.size x n_traits]). Subsampled parameter sets for ensemble runs.
- env.samples - Currently empty list (reserved for environmental samples).
- runs.samples - Currently empty list (reserved for run metadata).

Downstream contract: samples.Rdata is loaded by run.write.configs (in PEcAn.workflow) to generate model configuration files. It is also loaded by [get.results](#) and [run.sensitivity.analysis](#) to retrieve sample metadata for post-processing. This implicit file-based coupling is a refactoring target.

Value

Named list with 5 elements: `trait.samples`, `sa.samples`, `ensemble.samples`, `runs.samples`, `env.samples`. Returned invisibly.

Author(s)

David LeBauer, Shawn Serbin, Istem Fer, Om Kapale

<code>get.results</code>	<i>Read model output and save parsed results for sensitivity and ensemble analyses</i>
--------------------------	--

Description

Loads sample metadata and run IDs from disk, reads model output for each run, and saves parsed results for downstream analysis by `run.sensitivity.analysis` and `run.ensemble.analysis`.

Usage

```
get.results(
  settings,
  sa.ensemble.id = NULL,
  ens.ensemble.id = NULL,
  variable = NULL,
  start.year = NULL,
  end.year = NULL
)
```

Arguments

<code>settings</code>	list, read from settings file (xml) using <code>PEcAn.settings::read.settings()</code>
<code>sa.ensemble.id</code> , <code>ens.ensemble.id</code>	ensemble IDs for the sensitivity analysis and ensemble analysis. If not provided, they are first looked up from <code>settings</code> , then if not found they are not used and the most recent set of results is read from <code>samples.Rdata</code> in directory <code>settings\$outdir</code>
<code>variable</code>	variables to retrieve, as vector of names or expressions
<code>start.year</code> , <code>end.year</code>	first and last years to retrieve

Details

Output is placed in model output directory (`settings$outdir`).

Upstream contract (reads from `settings$outdir`):

`sensitivity.samples.<id>.Rdata` **or** `samples.Rdata` Loaded to obtain `sa.run.ids`, `pft.names`, `trait.names`, and `sa.samples` for sensitivity analysis. Produced by `run.write.configs`.

ensemble.samples.<id>.Rdata **or** samples.Rdata Loaded to obtain ens.run.ids, pft.names, and trait.names for ensemble analysis. Produced by [run.write.configs](#).

File-based side effects (saved to settings\$outdir):

sensitivity.output.<var>.<years>.<id>.Rdata Contains sensitivity.output: a named list (PFT -> matrix[n_quantiles x n_traits]) of model output values corresponding to each SA run. One file per variable/year combination.

ensemble.output.<var>.<years>.<id>.Rdata Contains ensemble.output: a numeric vector of model output values, one per ensemble member. One file per variable/year combination.

Downstream contract: sensitivity.output.*.Rdata is loaded by [run.sensitivity.analysis](#). ensemble.output.*.Rdata is loaded by [run.ensemble.analysis](#). Both use these files to compute diagnostics and plots. This implicit file-based coupling is a refactoring target.

Value

Nothing (called for side effects). Saves sensitivity.output and/or ensemble.output .Rdata files to settings\$outdir.

Author(s)

David LeBauer, Shawn Serbin, Mike Dietze, Ryan Kelly

get.sensitivity	<i>Calculate the sensitivity of a function at the median</i>
-----------------	--

Description

This function evaluates the sensitivity of a model to a parameter. This is done by evaluating the first derivative of the univariate spline estimate of the model response at the parameter median.

Usage

```
get.sensitivity(trait.samples, sa.splinefun)
```

Arguments

trait.samples parameter values to evaluate at their median
sa.splinefun fitted spline function. Must take two arguments.

Value

numeric estimate of model sensitivity to parameter

`get_parameter_samples` *Generate parameter samples from priors and MCMC posteriors (pure function)*

Description

This is the pure computation core extracted from `get.parameter.samples()`. It takes pre-loaded R objects as input and returns all computed samples directly, with no file I/O and no database calls.

Usage

```
get_parameter_samples(
  pft_names,
  prior_distns_list,
  trait_mcmc_list = NULL,
  ensemble.size = 1,
  ens.sample.method = "uniform",
  sa_quantiles = NULL,
  do_ensemble = TRUE,
  independent = TRUE
)
```

Arguments

<code>pft_names</code>	character vector of PFT names
<code>prior_distns_list</code>	list of data frames (one per PFT), each with columns: <code>distn</code> , <code>parama</code> , <code>paramb</code> , <code>n</code> . Row names are trait names.
<code>trait_mcmc_list</code>	list of trait MCMC results (one per PFT). Each element should be a named list of <code>coda::mcmc.list</code> objects, or <code>NULL</code> if no MCMC results exist for that PFT.
<code>ensemble.size</code>	integer. Number of runs in model ensemble.
<code>ens.sample.method</code>	character. One of "halton", "sobol", "torus", "lhc", "uniform".
<code>sa_quantiles</code>	numeric vector of quantiles for sensitivity analysis, or <code>NULL</code> to skip SA sampling.
<code>do_ensemble</code>	logical. Whether to generate ensemble samples.
<code>independent</code>	logical. <code>TRUE</code> means parameters were fitted individually (standard MA) and can be resampled independently. <code>FALSE</code> means they were fitted together (e.g. PDA) and correlations should be preserved.

Details

For the original disk-based wrapper, see [get.parameter.samples](#).

Value

named list with five elements:

trait.samples Named list by PFT, then by trait. Each leaf is a numeric vector of parameter samples.

sa.samples Sensitivity analysis samples (list), or empty list if sa_quantiles is NULL.

ensemble.samples Ensemble samples (list), or empty list if do_ensemble is FALSE.

runs.samples Reserved for future use (empty list).

env.samples Environmental samples (empty list, populated by downstream code if needed).

Author(s)

David LeBauer, Shawn Serbin, Istem Fer, Om Kapale

See Also

[get.parameter.samples](#) for the backward-compatible wrapper that handles file I/O and database lookups.

Examples

```
## Not run:
priors <- data.frame(
  distn = "norm",
  parama = 20,
  paramb = 5,
  n = 50,
  row.names = "SLA",
  stringsAsFactors = FALSE
)

result <- get_parameter_samples(
  pft_names = "temperate.Hardwood",
  prior_distns_list = list(priors),
  ensemble.size = 100,
  do_ensemble = TRUE
)

str(result$trait.samples)
str(result$ensemble.samples)

## End(Not run)
```

input.ens.gen	<i>Generate an ensemble of samples from one input</i>
---------------	---

Description

Given an input containing paths to multiple files, this function samples from the given paths to make an ensemble of the requested size. If method is 'sampling' they are sampled with replacement from the full set; if method is 'looping' they are taken sequentially with the sequence repeating as needed.

Usage

```
input.ens.gen(
  settings,
  ensemble_size,
  input,
  method = "sampling",
  parent_ids = NULL,
  bad_parent_action = c("error", "resample")
)
```

Arguments

settings	list of PEcAn settings
ensemble_size	size of ensemble
input	name of input to sample, e.g. "met", "veg", "pss"
method	Method for sampling - For now looping or sampling with replacement is implemented
parent_ids	integer vector of indices to be used as-is
bad_parent_action	How to handle entries in 'parent_id' that are not valid indices into the paths given for this input. See details

Details

If 'parent_ids' is provided, it is used as the indices for the current input. Use this when you have multiple inputs whose files should be sampled as a coordinated set. For example if your soil moisture files are matched to the rainfall amounts in your weather files, generate the met ensemble first ('met_ids <- input.ens.gen(..., input = "met", parent_ids = NULL)') and then pass those as parents to the soil moisture ensemble: 'input.ens.gen(..., input = "soilmoist", parent_ids = met_ids)'.

If any indices in 'parent_ids' are larger than the length of the input being sampled (e.g. '53' when the input only has 25 paths), the result is determined by 'bad_parent_action': "error" stops execution and requests the user to fix their inputs, "resample" uses the valid parents as-is and replaces the invalid parents via method = "sampling". Note that method "resample" is provided for backwards compatibility; we strongly recommend validating parent-child alignment before calling 'input.ens.gen', ideally in the same process that generates your parent ids.

Value

A list with elements 'id' showing the order of sampling and 'samples' with the paths selected from those indices.

Examples

```
## Not run:
settings <- PEcAn.settings::read.settings("pecan.xml")
input.ens.gen(
  settings,
  ensemble_size = 50,
  input = "met",
  method = "sampling"
)

## End(Not run)
```

kurtosis

Calculates the excess kurtosis of a vector

Description

Note that this calculates the 'excess kurtosis', which is defined as kurtosis - 3. This statistic is used in the calculation of the standard deviation of sample variance in the function [sd.var](#).

Usage

```
kurtosis(x)
```

Arguments

x vector of values

Value

numeric value of kurtosis

Author(s)

David LeBauer

References

NIST/SEMATECH e-Handbook of Statistical Methods, <http://www.itl.nist.gov/div898/handbook/eda/section3/eda35b.htm>, 2011-06-20.

load_pft_posteriors	<i>Load PFT priors and posteriors from disk/database</i>
---------------------	--

Description

Resolves each PFT's output directory and reads its prior distributions and (where available) MCMC posterior chains into memory. This is the disk/BETY I/O step that used to live inside [get.parameter.samples](#); it is kept separate so that the sampling functions ([get.parameter.samples](#) and the design generators) can share one loader rather than each carrying their own copy.

Usage

```
load_pft_posteriors(settings, posterior.files = rep(NA, length(settings$pfts)))
```

Arguments

<code>settings</code>	PEcAn settings object. Uses <code>settings\$pfts</code> (name, outdir, posteriorid per PFT), <code>settings\$database\$bety</code> for the optional BETY lookup, and <code>settings\$host\$name</code> .
<code>posterior.files</code>	character vector of posterior filenames, one per PFT, NA where none is specified. Length must match <code>settings\$pfts</code> .

Details

Posterior type is detected by content, not filename, via [load.posteriors](#): MCMC chains take precedence over distribution summaries. When a PFT's posterior is a joint fit (e.g. from PDA), the returned independent flag is set to FALSE so downstream sampling preserves the parameter correlations.

Value

A list with four elements:

pft_names	character vector of PFT names ("NULL" where unnamed).
prior_distns_list	list of prior distribution data frames, one per PFT (NULL entry where none was found).
trait_mcmc_list	list of trait MCMC results, one per PFT (NULL entry where the PFT has no MCMC chains).
independent	logical, FALSE if any PFT carries a joint posterior, TRUE otherwise.

Author(s)

David LeBauer, Shawn Serbin, Istem Fer, Om Kapale

See Also

[get.parameter.samples](#) for the sampling step that consumes this output.

plot_flux_uncertainty *Plot fit for heteroskedastic flux uncertainty*

Description

Plot fit for heteroskedastic flux uncertainty

Usage

```
plot_flux_uncertainty(f, ...)
```

Arguments

f	output of flux.uncertainty functions
...	optional graphical paramters

Author(s)

Mike Dietze, Carl Davidson

plot_sensitivities *Plot functions and quantiles used in sensitivity analysis*

Description

Generates a plot using [plot_sensitivity](#) for multiple traits.

Usage

```
plot_sensitivities(  
  sensitivity.plot.inputs,  
  prior.sensitivity.plot.inputs = NULL,  
  ...  
)
```

Arguments

sensitivity.plot.inputs	
inputs	
prior.sensitivity.plot.inputs	
priors	
...	arguments passed to plot_sensitivity

Value

list of plots, one per trait

plot_sensitivity	<i>Sensitivity plot</i>
------------------	-------------------------

Description

Plot univariate response of model output to a trait parameter.

Usage

```
plot_sensitivity(
  sa.sample,
  sa.spline,
  trait,
  y.range = c(0, 50),
  median.i = 4,
  prior.sa.sample = NULL,
  prior.sa.spline = NULL,
  fontsize = list(title = 12, axis = 8),
  linesize = 1,
  dotsize = 2
)
```

Arguments

sa.sample	trait quantiles used in sensitivity analysis
sa.spline	spline function estimated from sensitivity analysis
trait	trait name for title
y.range	limits for y axis of plot
median.i	index of median value in sa.sample; median.i == which(as.numeric(rownames(sa.sample)) == 50)
prior.sa.sample	similar to sa.sample, but for prior distribution. If given, plots sensitivity for prior run
prior.sa.spline	similar to sa.spline, but for prior trait distribution.
fontsize	(optional) list with three arguments that can be set to vary the fontsize of the title, axis labels, and axis title in the sensitivity plots
linesize	passed to ggplot to set line thickness
dotsize	passed to ggplot to set point size

Details

Plots for a single trait; called by [plot_sensitivities](#) to plot sensitivity plots for multiple traits.

Value

object of class ggplot

plot_variance_decomposition

Variance Decomposition Plots

Description

Plots variance decomposition tryptich: CV, elasticity, variance

Usage

```
plot_variance_decomposition(
  plot.inputs,
  fontsize = list(title = 18, axis = 14),
  order_by = c("Variance", "Elasticity", "CV (%)", "rowname", "none")
)
```

Arguments

plot.inputs	Output from a sensitivity analysis. Output must be of the form given by sensitivity.results\$variance.decomposition.output in model output
fontsize	list specifying the font size of the titles and axes of the graph
order_by	Result column to sort by, or "none" to retain input order

Value

ggplot object

Author(s)

David LeBauer, Carl Davidson, Chris Black

Examples

```
x <- list(trait.labels = c('a', 'b', 'c'),
  coef.vars = c(a=1,b=0.5, c=0.1),
  elasticities = c(a=1,b=2,c=0.5),
  variances = c(a = 20, b=30, c = 10))
plot_variance_decomposition(x)
plot_variance_decomposition(x, order_by = "rowname")
```

read.ameriflux.L2	<i>Read Ameriflux L2 Data</i>
-------------------	-------------------------------

Description

Read Ameriflux L2 Data

Usage

```
read.ameriflux.L2(file.name, year)
```

Arguments

file.name	path to file
year	currently ignored

Value

Ameriflux L2 data read from file

Author(s)

Mike Dietze, Carl Davidson

read.ensemble.output	<i>Reads output from model ensemble</i>
----------------------	---

Description

Reads output for an ensemble of length specified by `ensemble.size` and bounded by `start.year` and `end.year`

Usage

```
read.ensemble.output(  
  ensemble.size,  
  pecandir,  
  outdir,  
  start.year,  
  end.year,  
  variable,  
  ens.run.ids = NULL  
)
```

Arguments

ensemble.size	the number of ensemble members run
pecandir	specifies where pecan writes its configuration files
outdir	directory with model output to use in ensemble analysis
start.year	first year to include in ensemble analysis
end.year	last year to include in ensemble analysis
variable	target variables for ensemble analysis
ens.run.ids	dataframe. Must contain a column named "id" giving the run IDs to be read. If NULL, will attempt to read IDs from a file named "samples.Rdata" in pecandir

Value

a list of ensemble model output

Author(s)

Ryan Kelly, David LeBauer, Rob Kooper

read.ensemble.ts	<i>Reads ensemble time-series from PEcAn for the selected target variable</i>
------------------	---

Description

Reads ensemble time-series from PEcAn for the selected target variable

Usage

```
read.ensemble.ts(
  settings,
  ensemble.id = NULL,
  variable = NULL,
  start.year = NULL,
  end.year = NULL
)
```

Arguments

settings	PEcAn settings object
ensemble.id	database ID, taken from settings if not specified
variable	variable name to process. Required
start.year, end.year	taken from settings if not specified

Value

list

Author(s)

Michael Dietze, Ryan Kelly

read.sa.output	<i>Reads output of sensitivity analysis runs</i>
----------------	--

Description

Reads output of sensitivity analysis runs

Usage

```
read.sa.output(
  traits,
  quantiles,
  pecandir,
  outdir,
  pft.name = "",
  start.year,
  end.year,
  variable,
  sa.run.ids = NULL,
  per.pft = FALSE
)
```

Arguments

traits	model parameters included in the sensitivity analysis
quantiles	quantiles selected for sensitivity analysis
pecandir	specifies where pecan writes its configuration files
outdir	directory with model output to use in sensitivity analysis
pft.name	name of PFT used in sensitivity analysis (Optional)
start.year	first year to include in sensitivity analysis
end.year	last year to include in sensitivity analysis
variable	variables to be read from model output
sa.run.ids	run ids of this site's sensitivity runs, as built by sa_run_id_table . Needed when several sites share 'runs_manifest.csv'; if NULL, every run in the manifest is used
per.pft	flag to determine whether we want SA on pft-specific variables

Value

dataframe with one col per quantile analysed and one row per trait, each cell is a list of AGB over time

Author(s)

Ryan Kelly, David LeBauer, Rob Kooper, Mike Dietze, Istem Fer, Akash B V

run.ensemble.analysis *Run ensemble analysis on finished model runs*

Description

Loads parsed ensemble model output from disk, computes summary statistics (mean, median, quantiles), and generates diagnostic plots (histogram, boxplot, and optionally time series).

Usage

```
run.ensemble.analysis(
  settings,
  plot.timeseries = NA,
  ensemble.id = NULL,
  variable = NULL,
  start.year = NULL,
  end.year = NULL,
  ...
)
```

Arguments

settings	PEcAn settings object
plot.timeseries	if TRUE plots a modeled timeseries of target variable(s) with CIs
ensemble.id	database ID, taken from settings if not specified
variable	variable name to process, taken from settings if not specified
start.year, end.year	taken from settings if not specified
...	additional arguments passed to ensemble.ts

Details

Upstream contract (reads from settings\$outdir):

ensemble.output.<var>.<years>.<id>.Rdata Produced by [get.results](#). Contains ensemble.output:
a numeric vector of model output values, one per ensemble member.

File-based side effects (saved to settings\$outdir):

ensemble.analysis.<var>.<years>.<id>.pdf Histogram and boxplot of ensemble output distribution.

ensemble.ts.<var>.<years>.<id>.pdf (Optional) Time series plot with confidence intervals. Generated when plot.timeseries is not NA.

ensemble.ts.analysis.<var>.<years>.<id>.Rdata (Optional) Contains ensemble.ts.analysis: time series summary statistics. Saved when plot.timeseries is not NA.

Note: This is a terminal step — nothing downstream loads these files programmatically. The results are consumed by visualization or user inspection.

Value

Nothing (called for side effects). Creates ensemble plots as PDF files and optionally saves time series analysis results.

Author(s)

David LeBauer, Shawn Serbin, Ryan Kelly

```
run.sensitivity.analysis
```

Run sensitivity analysis on finished model runs

Description

Loads parameter samples and parsed model output from disk, computes sensitivity coefficients and variance decomposition for each PFT/trait, and saves results and diagnostic plots.

Usage

```
run.sensitivity.analysis(
  settings,
  plot = TRUE,
  ensemble.id = NULL,
  variable = NULL,
  start.year = NULL,
  end.year = NULL,
  pfts = NULL,
  ...
)
```

Arguments

settings	a PEcAn settings object
plot	logical. Option to generate sensitivity analysis and variance decomposition plots (plot = TRUE) or to turn these plots off (plot = FALSE).
ensemble.id	ensemble ID
variable	which variable(s) to do sensitivity analysis for. Defaults to all specified in settings
start.year	defaults to what is specified in settings
end.year	defaults to what is specified in settings
pfts	a vector of PFT names found in settings to run sensitivity analysis on
...	currently unused

Details**Upstream contract (reads from settings\$outdir):**

`samples.Rdata` Produced by [get.parameter.samples](#). Loaded to obtain `trait.samples`, `trait.names`, `pft.names`, and `sa.run.ids`.

`sensitivity.samples.<id>.Rdata` Produced by [run.write.configs](#). If present, overrides `sa.run.ids` and `sa.samples` from `samples.Rdata`.

`sensitivity.output.<var>.<years>.<id>.Rdata` Produced by [get.results](#). Contains `sensitivity.output`: a named list (PFT -> matrix) of model outputs for each SA run.

File-based side effects (saved to settings\$outdir):

`sensitivity.results.<var>.<years>.<id>.Rdata` Contains `sensitivity.results`: a named list (PFT -> list) with elements `sensitivity.output` (partial derivatives) and `variance.decomposition.output` (elasticities and partial variances).

`sensitivityanalysis.<pft>.<var>.<years>.<id>.pdf` Sensitivity analysis diagnostic plots (one per PFT). Generated when `plot = TRUE`.

`variance.decomposition.<pft>.<var>.<years>.<id>.pdf` Variance decomposition "popsicle" plots (one per PFT). Generated when `plot = TRUE`.

Note: This is a terminal step in the workflow — nothing downstream loads `sensitivity.results.*.Rdata` programmatically. The results are consumed by visualization or user inspection.

Value

Nothing (called for side effects). Saves `sensitivity.results` as `sensitivity.results.*.Rdata` and optional PDF plots.

Author(s)

David LeBauer, Shawn Serbin, Ryan Kelly

Examples

```
## Not run:
library(PeAn.settings)
library(PeAn.uncertainty)
settings <- read.settings("path/to/pecan.xml")
run.sensitivity.analysis(settings)

## End(Not run)
```

runModule.get.results *Apply get.results to each of a list of settings*

Description

Apply get.results to each of a list of settings

Usage

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

Arguments

settings a PEcAn Settings or MultiSettings object

See Also

get.results

runModule.run.ensemble.analysis
Run ensemble analyses across all sites in settings

Description

Run ensemble analyses across all sites in settings

Usage

```
runModule.run.ensemble.analysis(settings, ...)
```

Arguments

settings PEcAn settings object
... arguments passed on to run.ensemble.analysis

```
runModule.run.sensitivity.analysis
```

Run sensitivity analysis for every Settings in a MultiSettings

Description

Caution: Not yet working for multisite settings. It will `_run_` the analysis for all sites, but each site will overwrite the result from the previous one.

Usage

```
runModule.run.sensitivity.analysis(settings, ...)
```

Arguments

<code>settings</code>	PEcan settings object
<code>...</code>	additional arguments passed on to <code>'run.sensitivity.analysis'</code>

<code>sa.splinefun</code>	<i>Spline estimate of univariate relationship between parameter value and model output</i>
---------------------------	--

Description

Creates a spline function using the `splinefun` function that estimates univariate response of parameter input to model output

Usage

```
sa.splinefun(quantiles.input, quantiles.output)
```

Arguments

<code>quantiles.input</code>	passed to <code>'x'</code> argument of <code>'stats::splinefun'</code>
<code>quantiles.output</code>	passed to <code>'y'</code> argument of <code>'stats::splinefun'</code>

Value

function

sa_run_descriptions	<i>Describe each run of a sensitivity analysis design</i>
---------------------	---

Description

The writers need three things about each run: what to call it, how to record it in the database, and what to put in the run manifest that the sensitivity analysis post-processing reads back. All three follow from the design labels, so they are built here rather than as a side effect of writing configs.

Usage

```
sa_run_descriptions(design_matrix, site_id, pft_names)
```

Arguments

design_matrix	The design from generate_OAT_SA_design .
site_id	Site id, used as the last part of each run id.
pft_names	Names of the PFTs in the run, used to describe the median run, which moves nothing.

Value

A data.frame with one row per design row: id, paramlist, and the pft_name, trait, quantile and type the manifest carries. This is what [write_ensemble_configs](#) takes as run_descriptions.

sa_run_id_table	<i>Index a sensitivity analysis run's ids by PFT, quantile and trait</i>
-----------------	--

Description

The sensitivity analysis post-processing looks runs up by which trait sits at which quantile, rather than by run order. This builds that lookup from the design and the ids the writer returned, so it no longer has to be produced as a side effect of writing the configs.

Usage

```
sa_run_id_table(design_matrix, run_ids)
```

Arguments

design_matrix	The design from generate_OAT_SA_design .
run_ids	Run ids, one per design row, in design order.

Value

A named list with one data.frame per PFT, quantiles as rownames and traits as columns, holding the run id for each combination. The median run fills the median row of every trait, since one run covers them all. This is the `sa.run.ids` object saved in `sensitivity.samples.<id>.Rdata`.

sa_run_samples

Expand SA quantile samples into one parameter set per run

Description

[get_parameter_samples](#) returns sensitivity analysis samples as one table per PFT, quantiles down the rows and traits across the columns. The config writers want the opposite shape: one row per model run, holding the parameter values that run uses. This converts between the two, using the labels the design carries.

Usage

```
sa_run_samples(sa_samples, design_matrix)
```

Arguments

sa_samples	Sensitivity analysis samples, a named list with one data.frame per PFT, quantiles as rownames and traits as columns, as returned by get_parameter_samples .
design_matrix	The design from generate_OAT_SA_design , carrying sa_pft, sa_trait and sa_quantile.

Details

Every run starts from each PFT's median values. A run that the design labels as moving a trait then has that one value replaced with the trait's value at the labelled quantile, which is what makes a one-at-a-time design separable. The first row, the median run, keeps every value at its median.

Entries that are not PFTs, such as env, are carried at their medians for every run, matching what `write.sa.configs` passes today.

Value

A named list with one data.frame per entry of `sa_samples`, each with one row per design row and one column per trait. This is the shape [write_ensemble_configs](#) takes as `ensemble.samples`.

See Also

[generate_OAT_SA_design](#) for the labels this reads.

sd.var

Calculates the standard deviation of the variance estimate

Description

Uses the equation $\sigma^4 \left(\frac{2}{n-1} + \frac{\kappa}{n} \right)$

Usage

```
sd.var(x)
```

Arguments

x sample

Value

estimate of standard deviation of the sample variance

Author(s)

David LeBauer

References

Mood, Graybill, Boes 1974 'Introduction to the Theory of Statistics' 3rd ed. p 229; Casella and Berger 'Statistical Inference' p 364 ex. 7.45; 'Reference for Var(s^2)' CrossValidated <http://stats.stackexchange.com/q/29905/1381>, 'Calculating required sample size, precision of variance estimate' CrossValidated <http://stats.stackexchange.com/q/7004/1381>, 'Variance of Sample Variance?' Mathematics - Stack Exchange <http://math.stackexchange.com/q/72975/3733>

sensitivity.analysis *Performs univariate sensitivity analysis and variance decomposition*

Description

This function estimates the univariate responses of a model to a parameter for a set of traits, calculates the model sensitivity at the median, and performs a variance decomposition. This function results in a set of ensitivity plots (one per variable) and plot_variance_decomposition.

Usage

```
sensitivity.analysis(trait.samples, sa.samples, sa.output, outdir)
```

Arguments

trait.samples	list of vectors, one per trait, representing samples of the trait value, with length equal to the mcmc chain length. Samples are taken from either the prior distribution or meta-analysis results
sa.samples	data.frame with one column per trait and one row for the set of quantiles used in sensitivity analysis.
sa.output	list of data.frames, similar to sa.samples, except cells contain the results of a model run with that trait x quantile combination and all other traits held at their median value
outdir	directory to which plots are written

Value

results of sensitivity analysis

Author(s)

David LeBauer

Examples

```
## Not run:
sensitivity.analysis(
  trait.samples[[pft$name]],
  sa.samples[[pft$name]],
  sa.agb[[pft$name]],
  pft$outdir
)

## End(Not run)
```

sensitivity.filename	<i>Generate sensitivity analysis filenames</i>
----------------------	--

Description

Generate sensitivity analysis filenames

Usage

```
sensitivity.filename(
  settings,
  prefix = "sensitivity.samples",
  suffix = "Rdata",
  all.var.yr = TRUE,
  pft = NULL,
  ensemble.id = settings$sensitivity.analysis$ensemble.id,
```

```

    variable = settings$sensitivity.analysis$variable,
    start.year = settings$sensitivity.analysis$start.year,
    end.year = settings$sensitivity.analysis$end.year
  )

```

Arguments

settings	list of PEcAn settings.
prefix	string to appear at the beginning of the filename
suffix	file extension: string to appear at the end of the filename
all.var.yr	logical: does ensemble include all vars and years? If FALSE, filename will include years and vars
pft	name of PFT used for analysis. If NULL, assumes all PFTs in run are used and does not add them to the filename
ensemble.id	ensemble ID(s)
variable	variable(s) included in the ensemble.
start.year, end.year	first and last year simulated.

Details

Generally uses values in settings, but can be overwritten for manual uses

Value

a filename

Author(s)

Ryan Kelly

spline.ensemble	<i>Spline Ensemble</i>
-----------------	------------------------

Description

Estimate model output based on univariate splines

Usage

```
spline.ensemble(gi.phii, median)
```

Arguments

gi.phii	matrix given as output from get.gi.phii
median	median value around which variance will be calculated

Details

Accepts output from `get.gi.phii` (the matrix $g(\phi_i)$) and produces spline estimate of $f(\phi)$ for use in estimating closure term associated with spline approximation

Author(s)

David LeBauer

spline.truncate	<i>Truncate spline at zero if..</i>
-----------------	-------------------------------------

Description

Truncate spline at zero if $P[x < 0] < \text{pnorm}(-3)$ `pnorm(-3)` chosen as default value for min quantile because this is the default low end of range for the sensitivity analysis. This parameter could be determined based on minimum value in `settings$sensitivity.analysis$quantiles`

Usage

```
spline.truncate(x, min.quantile = stats::pnorm(-3))
```

Arguments

x	vector
min.quantile	threshold quantile for testing lower bound on variable

Value

either x or a vector with values < 0 converted to zero

Author(s)

David LeBauer

Examples

```
set.seed(0)
x <- c(rgamma(998,1,1), rnorm(10))
min(x) # -0.5238
min(PEcAn.uncertainty::spline.truncate(x))
```

variance.stats	<i>variance statistics</i>
----------------	----------------------------

Description

Variance and SD(variance)

Usage

```
variance.stats(x)
```

Arguments

x numeric vector

Details

calculates variance and sd of variance using the var from base R and [sd.var](#) from PEcAn.

Value

list with variance and sd of variance

Author(s)

David LeBauer

write.ensemble.configs	<i>Write ensemble config files</i>
------------------------	------------------------------------

Description

Writes config files for use in meta-analysis and returns a list of run ids. Given a pft.xml object, a list of lists as supplied by get.sa.samples, a name to distinguish the output files, and the directory to place the files.

Usage

```
write.ensemble.configs(
  input_design,
  ensemble.size,
  defaults,
  ensemble.samples,
  settings,
  model,
```

```

    clean = FALSE,
    write.to.db = TRUE,
    restart = NULL,
    rename = FALSE,
    run_descriptions = NULL,
    ensemble.id = NULL
  )

```

Arguments

input_design	design matrix describing sampled inputs (see 'run.write.configs()'). Columns named after 'settings\$run\$inputs' tags give 1-based indices into each input's 'path' list and rows follow run order. Requires 'nrow(input_design) >= ensemble.size'; extra rows are ignored.
ensemble.size	size of ensemble
defaults	pft
ensemble.samples	list of lists supplied by get.ensemble.samples
settings	list of PEcAn settings
model	name of model to be run, e.g. "ED2" or "SIPNET"
clean	remove old output first?
write.to.db	logical: Record this run in BETY?
restart	In case this is a continuation of an old simulation. Needs to be a list with name tags of runid, inputs, new.params (parameters), new.state (initial condition), ensemble.id (ensemble id), start.time and stop.time. See Details.
rename	Decide if we want to rename previous output files, for example convert from sipnet.out to sipnet.2020-07-16.out.
run_descriptions	optional data.frame, one row per design row, with id, paramlist and the pft_name, trait, quantile and type the run manifest carries. Lets a caller that already knows what each run is describe them, instead of these being generated as ensemble members.
ensemble.id	optional id for this set of runs, used instead of the one in settings\$ensemble\$id.

Details

The restart functionality is developed using model specific functions by calling write_restart.modelname function. First, you need to make sure that this function is already exist for your desired model. See here <https://pecanproject.github.io/pecan-documentation/latest/pecan-models.html>. new state is a dataframe with a different column for each state variable. The number of the rows in this dataframe needs to be the same as the ensemble size. The state variables that you can use for setting up initial conditions are model specific. Check the documentation of the write_restart.<modelname> function for the model you are using. The units for the state variables need to be in the PEcAn standard units which can be found in [standard_vars](#). new.params also has similar structure to ensemble.samples which is sent as an argument.

Value

list, containing \$runs = data frame of runids, \$ensemble.id = the ensemble ID for these runs, and \$samples with ids and samples used for each tag. Also writes sensitivity analysis configuration files as a side effect

Author(s)

David LeBauer, Carl Davidson, Hamze Dokoochaki

Index

`compute_sobol_indices`, 3

`ensemble.filename`, 3
`ensemble.ts`, 5, 27

`flux.uncertainty`, 5

`generate_joint_ensemble_design`, 6, 8
`generate_OAT_SA_design`, 7, 32, 33
`get.change`, 9
`get.coef.var`, 10
`get.elasticity`, 10
`get.ensemble.samples`, 11, 39
`get.gi.phii`, 12, 36
`get.parameter.samples`, 12, 16, 17, 20, 29
`get.results`, 13, 14, 27, 29
`get.sensitivity`, 10, 15
`get_parameter_samples`, 7, 8, 12, 16, 20, 33

`input.ens.gen`, 18

`kurtosis`, 19

`load.posteriors`, 20
`load_pft_posteriors`, 7, 8, 20

`plot_flux_uncertainty`, 21
`plot_sensitivities`, 21, 22
`plot_sensitivity`, 21, 22
`plot_variance_decomposition`, 12, 23

`read.ameriflux.L2`, 24
`read.ensemble.output`, 24
`read.ensemble.ts`, 25
`read.sa.output`, 26
`run.ensemble.analysis`, 14, 15, 27
`run.sensitivity.analysis`, 13–15, 28
`run.write.configs`, 14, 15, 29
`runModule.get.results`, 30
`runModule.run.ensemble.analysis`, 30
`runModule.run.sensitivity.analysis`, 31

`sa.splinefun`, 31
`sa_run_descriptions`, 32
`sa_run_id_table`, 26, 32
`sa_run_samples`, 33
`sd.var`, 19, 34, 38
`sensitivity.analysis`, 12, 34
`sensitivity.filename`, 35
`spline.ensemble`, 12, 36
`spline.truncate`, 37
`standard_vars`, 39

`variance.stats`, 38

`write.ensemble.configs`, 32, 33, 38