Package: n2kanalysis (via r-universe)

August 10, 2024

Title Generic Functions to Analyse Data from the 'Natura 2000'
Monitoring

Version 0.3.2

Description All generic functions and classes for the analysis for the
'Natura 2000' monitoring. The classes contain all required data
and definitions to fit the model without the need to access
other sources. Potentially they might need access to one or
more parent objects. An aggregation object might for example
need the result of an imputation object. The actual definition
of the analysis, using these generic function and classes, is
defined in dedictated analysis R packages for every monitoring
scheme. For example 'abvanalysis' and 'watervogelanalysis'.

License GPL-3

URL https://doi.org/10.5281/zenodo.3576047

BugReports https://github.com/inbo/n2kanalysis/issues

Depends R (>= 4.2.0)

Imports assertthat (>= 0.2.1), aws.s3, digest (>= 0.6.23.2), dplyr,
fs, MASS, methods, multimput (>= 0.2.13), n2khelper (>= 0.5.0),
purrr, rlang, RODBC, tibble, tidyr (>= 0.4.0), yaml

Suggests INLA (>= 23.04.24), knitr, Matrix, parallel, rmarkdown, sn,
testthat (>= 2.0.1)

VignetteBuilder knitr

Remotes inbo/multimput, inbo/n2khelper

Additional_repositories https://inla.r-inla-download.org/R/stable,
https://inbo.r-universe.dev

Config/checklist/communities inbo

Config/checklist/keywords analysis, reproducible research, natura
2000, monitoring

Encoding UTF-8

Language en-GB

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.1

Collate 'n2k_parameter_class.R' 'n2k_anomaly_class.R'
'n2k_analysis_version_class.R' 'combine.R'
'n2k_contrast_class.R' 'n2k_analysis_metadata_class.R'
'n2k_result_class.R' 'combine_result.R' 'import_s3_classes.R'
'delete_model.R' 'display.R' 'n2k_model_class.R'
'n2k_inla_class.R' 'extract.R' 'fit_every_model.R'
'fit_model.R' 'fit_model_character.R' 'n2k_aggregate_class.R'
'fit_model_n2k_aggregate.R' 'n2k_composite_class.R'
'fit_model_n2k_composite.R' 'n2k_hurdle_imputed_class.R'
'fit_model_n2k_hurdle_imputed.R' 'fit_model_n2k_inla.R'
'n2k_inla_comparison_class.R' 'fit_model_n2k_inla_comparison.R'
'n2k_manifest_class.R' 'fit_model_n2k_manifest.R'
'n2k_model_imputed_class.R' 'fit_model_n2k_model_imputed.R'
'fit_model_s3_object.R' 'get_analysis_date.R'
'get_analysis_version.R' 'get_anomaly.R'
'get_anomaly_n2k_inla.R' 'get_anomaly_n2k_model.R' 'get_data.R'
'get_file_fingerprint.R' 'get_formula.R'
'get_location_group_id.R' 'get_model.R' 'get_model_parameter.R'
'get_model_parameter_n2k_aggregated.R'
'get_model_parameter_n2k_composite.R'
'get_model_parameter_n2k_hurdle_imputed.R'
'get_model_parameter_n2k_inla.R'
'get_model_parameter_n2k_inla_comparison.R'
'get_model_parameter_n2k_model_imputed.R' 'get_model_type.R'
'get_parents.R' 'get_result.R' 'get_result_character.R'
'get_result_n2k_inla.R' 'get_result_n2k_model.R'
'get_result_n2kmanifest.R' 'get_result_s3_object.R'
'get_scheme_id.R' 'get_seed.R' 'get_species_group_id.R'
'get_status_fingerprint.R' 'inla_inverse.R'
'manifest_yaml_to_bash.R' 'mark_obsolete_dataset.R'
'n2k_aggregated.R' 'n2k_composite.R' 'n2k_hurdle_imputed.R'
'n2k_import_class.R' 'n2k_import.R' 'n2k_inla.R'
'n2k_inla_comparison.R' 'n2k_manifest.R' 'n2k_model_imputed.R'
'parent_status.R' 'read_manifest.R' 'read_model.R'
'read_result.R' 'result_estimate.R'
'result_estimate_character.R' 'result_estimate_n2k_result.R'
'result_metadata.R' 'result_metadata_character.R'
'result_metadata_n2k_result.R'
'select_factor_count_strictly_positive.R'
'select_factor_treshold.R' 'select_observed_range.R'
'session_package.R' 'sha1.R' 'status.R' 'store_manifest.R'
'store_manifest_yaml.R' 'store_model.R' 'union.R'
'valid_object.R'

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

RemoteUrl https://github.com/inbo/n2kanalysis

RemoteRef HEAD

RemoteSha 07ef66200f9c102f4000c4285a492561d69b1917

Contents

---

  combine   *Combine multiple n2k objects*

Description

   Combine multiple n2k objects

Usage

```
combine(...)

## S4 method for signature 'n2kAnalysisVersion'
combine(...)

## S4 method for signature 'n2kParameter'
combine(...)

## S4 method for signature 'n2kAnomaly'
combine(...)

## S4 method for signature 'n2kResult'
combine(...)
```

Arguments

...	n2k objects

---

delete_model	*Delete an* n2kModel *object*

Description

Delete an n2kModel object

Usage

```
delete_model(x, base, project)

## S4 method for signature 'ANY,character'
delete_model(x, base, project)

## S4 method for signature 'ANY,s3_bucket'
delete_model(x, base, project)
```

Arguments

x	the file fingerprint of the n2kModel
base	the base location
project	will be a relative path within the base location

---

display	*Display a message.*

Description

This is a short cut for if(verbose) message(x).

Usage

```
display(verbose, message, linefeed = TRUE)
```

Arguments

verbose	A logical. When TRUE print the message. When FALSE do nothing.
message	a vector passed to [message()]().
linefeed	A logical. When TRUE append a newline character at the end of the message.

---

extract   *Extract the relevant coefficients*

Description

Extract the relevant coefficients

Usage

```
extract(extractor, object, base, project)

## S4 method for signature 'ANY,character'
extract(extractor, object, base, project)

## S4 method for signature 'ANY,n2kInla'
extract(extractor, object, base = NULL, project = NULL)
```

Arguments

extractor	the extractor function
object	the n2kModel object
base	the optional base location of the object
project	the optional subdirectory

Value

the relevant coefficients

---

fit_every_model   *Fit the model to the analysis files*

Description

Fit the model to the analysis files

Usage

```
fit_every_model(path, status, verbose = TRUE, n_cluster = 1, ...)
```

Arguments

path	The path containing the analysis files.
status	A vector with status levels to (re-)fit the model. Defaults to c("new", "waiting") when missing.
verbose	Show the name of the current analysis file on screen. Defaults to TRUE
n_cluster	The number of clusters to use.
...	Arguments passed to [fit_model()](fit_model)

---

fit_model	*Fit an* n2kModel *object*

Description

Fit an n2kModel object

Usage

```
fit_model(x, ...)

## S4 method for signature 'character'
fit_model(
  x,
  base,
  project,
  status = c("new", "waiting"),
  verbose = TRUE,
  ...,
  bucket
)

## S4 method for signature 'n2kAggregate'
fit_model(x, ...)

## S4 method for signature 'n2kComposite'
fit_model(x, base, project, status = "new", ...)

## S4 method for signature 'n2kHurdleImputed'
fit_model(x, base, project, status = c("new", "waiting"), ...)

## S4 method for signature 'n2kInla'
fit_model(
  x,
  status = "new",
  ...,
  timeout = NULL,
```

```
  seed = get_seed(x),
  num_threads = NULL,
  parallel_configs = TRUE
)

## S4 method for signature 'n2kInlaComparison'
fit_model(x, base, project, status = "new", verbose = TRUE, ...)

## S4 method for signature 'n2kManifest'
fit_model(
  x,
  base,
  project,
  status = c("new", "waiting"),
  verbose = TRUE,
  ...,
  local = tempfile("fit_model"),
  first = FALSE
)

## S4 method for signature 'n2kModelImputed'
fit_model(x, ...)

## S4 method for signature 's3_object'
fit_model(x, status = c("new", "waiting"), ...)
```

Arguments

x	the n2kModel
...	other arguments. See details
base	The root of a project. Can be either a directory on a file system or an AWS S3 bucket object. Extracted from bucket or x when missing.
project	The subdirectory of the project. Is relative the base. Extracted from x when missing.
status	A vector with status levels naming the levels which should be calculated. Defaults to "new".
verbose	A logical indicating if the function should display the name of the file and the status. Defaults to TRUE.
bucket	The name of the AWS S3 bucket. Only used when base is missing.
timeout	the optional number of second until the model will time out
seed	See the same argument in INLA::inla.qsample() for further information. In order to produce reproducible results, you ALSO need to make sure the RNG in R is in the same state, see the example in INLA::inla.posterior.sample(). When seed is non-zero, num_threads is forced to "1:1" and parallel_configs is set to FALSE, since parallel sampling would not produce a reproducible sequence of pseudo-random numbers.

num_threads	The number of threads to use in the format "A:B" defining the number threads in the outer (A) and inner (B) layer for nested parallelism. A "0" will be replaced intelligently. seed != 0 requires serial computations.
parallel_configs	
	Logical. If TRUE and not on Windows, then try to run each configuration in parallel (not Windows) using A threads (see num_threads), where each of them is using B:0 threads.
local	A local folder into which objects from an AWS S3 bucket are downloaded.
first	A logical. first = TRUE implies to fit only the first object in the manifest with matching status. first = FALSE implies to fit all objects in the manifest with matching status. Defaults to FALSE.

Details

- status: A vector with status levels naming the levels which should be recalculated. Defaults to c("new", "waiting").

- verbose: A logical indicating if the function should display the name of the file and the status. Defaults to TRUE.

---

get_analysis_date   *Get the analysis date of an* n2kModel *object*

Description

Get the analysis date of an n2kModel object

Usage

```
get_analysis_date(x)

## S4 method for signature 'n2kAnalysisMetadata'
get_analysis_date(x)
```

Arguments

x   the n2kModel object

Value

the analysis date of the object

---

get_analysis_version *Get an analysis version*

Description

Get an analysis version

Usage

```
get_analysis_version(version)

## S4 method for signature 'data.frame'
get_analysis_version(version)

## S4 method for signature 'n2kAnalysisMetadata'
get_analysis_version(version)

## S4 method for signature 'sessionInfo'
get_analysis_version(version)
```

Arguments

version   the object to extract the version

---

get_anomaly *Get the anomalies from a model*

Description

Get the anomalies from a model

Usage

```
get_anomaly(analysis, ...)

## S4 method for signature 'n2kInla'
get_anomaly(
  analysis,
  n = 20,
  expected_ratio = 5,
  expected_absent = 5,
  random_threshold = 1.05,
  verbose = TRUE,
  ...
)
```

```
## S4 method for signature 'n2kModel'
get_anomaly(analysis, verbose = TRUE, ...)
```

Arguments

analysis	The model to add
...	Extra options. See details.
n	the number of anomalies per category.
expected_ratio	Observations that have observed / fitted > expected_ratio or fitted / observed > expected_ratio are potential anomalies. Defaults to 5, which implies that observed values that are 5 times higher or lower than the fitted values are potential anomalies.
expected_absent	
	Zero observations where fitted > expected_absent are potential anomalies.
random_threshold	
	The minimal relative effect size of a random effect. Random effect with a smaller effect size will never be an anomaly. Defaults to 1.05 (5%).
verbose	Print extra information on the screen

---

get_data *Get the data of an* n2kModel *object*

Description

Get the data of an n2kModel object

Usage

```
get_data(x)

## S4 method for signature 'n2kInla'
get_data(x)
```

Arguments

x	the n2kModel object

Value

the data of the object

---

get_file_fingerprint   *Get the file fingerprint of an* n2kModel *object*

Description

Get the file fingerprint of an n2kModel object

Usage

```
get_file_fingerprint(x)

## S4 method for signature 'n2kAnalysisMetadata'
get_file_fingerprint(x)

## S4 method for signature 'n2kManifest'
get_file_fingerprint(x)
```

Arguments

x   the n2kModel object

Value

the file fingerprint of the object

---

get_formula   *Get the formula of an* n2kModel *object*

Description

Get the formula of an n2kModel object

Usage

```
get_formula(x)

## S4 method for signature 'n2kAnalysisMetadata'
get_formula(x)
```

Arguments

x   the n2kModel object

Value

the formula of the object

get_location_group_id   *Get the location group id of an* n2kModel *object*

Description

Get the location group id of an n2kModel object

Usage

```
get_location_group_id(x)

## S4 method for signature 'n2kAnalysisMetadata'
get_location_group_id(x)
```

Arguments

x   the n2kModel object

Value

the location group id of the object

get_model   *Get the model of a* n2kModel *object*

Description

Get the model of a n2kModel object

Usage

```
get_model(x)

## S4 method for signature 'n2kInla'
get_model(x)

## S4 method for signature 'character'
get_model(x)
```

Arguments

x   the n2kModel object

Value

the model of the object

---

get_model_parameter *Add the model parameters from a model*

Description

Add the model parameters from a model

Usage

```
get_model_parameter(analysis, ...)

## S4 method for signature 'n2kAggregate'
get_model_parameter(analysis, ...)

## S4 method for signature 'n2kComposite'
get_model_parameter(analysis, ...)

## S4 method for signature 'n2kHurdleImputed'
get_model_parameter(analysis, ...)

## S4 method for signature 'n2kInla'
get_model_parameter(analysis, verbose = TRUE, ...)

## S4 method for signature 'n2kInlaComparison'
get_model_parameter(analysis, ...)

## S4 method for signature 'n2kModelImputed'
get_model_parameter(analysis, ...)
```

Arguments

analysis	The model to add.
...	extra options
verbose	Print extra information on the screen

---

get_model_type *Get the model type of an* n2kModel *object*

Description

Get the model type of an n2kModel object

Usage

```
get_model_type(x)

## S4 method for signature 'n2kAnalysisMetadata'
get_model_type(x)
```

Arguments

x   the n2kModel object

Value

the model type of the object

---

get_parents   *Get the object of the parent analyses of an* n2kModel *object*

Description

Get the object of the parent analyses of an n2kModel object

Usage

```
get_parents(child, base, project)
```

Arguments

child   the child object

base   the base location to read the model

project   will be a relative path within the base location

---

get_result   *Add the results from an analysis*

Description

Add the results from an analysis

Usage

```
get_result(x, ...)

## S4 method for signature 'character'
get_result(x, n_cluster = 1, verbose = TRUE, ...)

## S4 method for signature 'n2kInla'
get_result(x, verbose = TRUE, ...)

## S4 method for signature 'n2kModel'
get_result(x, verbose = TRUE, ...)

## S4 method for signature 'n2kManifest'
get_result(x, ..., base, project, verbose = TRUE)

## S4 method for signature 's3_object'
get_result(x, ...)
```

Arguments

x	object with the current results
...	further arguments (see Details)
n_cluster	The number of clusters to run this function in parallel. Defaults to 1 (= no parallel computing).
verbose	Print extra information on the screen
base	the base location to read the model
project	will be a relative path within the base location

---

get_scheme_id   *Get the* scheme_id *of an* n2kModel *object*

Description

Get the scheme_id of an n2kModel object

Usage

```
get_scheme_id(x)

## S4 method for signature 'n2kAnalysisMetadata'
get_scheme_id(x)
```

Arguments

x	the n2kModel object

Value

the `scheme_id` of the object

---

get_seed *Get the seed of an* n2kModel *object*

Description

Get the seed of an n2kModel object

Usage

```
get_seed(x)

## S4 method for signature 'n2kAnalysisMetadata'
get_seed(x)
```

Arguments

x   the n2kModel object

Value

the seed of the object

---

get_species_group_id *Get the species group id of an* n2kModel *object*

Description

Get the species group id of an n2kModel object

Usage

```
get_species_group_id(x)

## S4 method for signature 'n2kAnalysisMetadata'
get_species_group_id(x)
```

Arguments

x   the n2kModel object

Value

the species group id of the object

---

get_status_fingerprint

Get the status fingerprint of an n2kModel *object*

Description

Get the status fingerprint of an n2kModel object

Usage

```
get_status_fingerprint(x)

## S4 method for signature 'n2kAnalysisMetadata'
get_status_fingerprint(x)
```

Arguments

x   the n2kModel object

Value

the status fingerprint of the object

---

inla_inverse   *Calculate the mean and 95\ marginal*

Description

This is useful for calculating the variance of a random effect when the marginal gives the precision of the random effect.

Usage

```
inla_inverse(marginal)
```

Arguments

marginal   The INLA marginal

---

manifest_yaml_to_bash *Convert a manifest yaml file into a bash script*

Description

Convert a manifest yaml file into a bash script

Usage

```
manifest_yaml_to_bash(
  base,
  project,
  hash,
  shutdown = FALSE,
  split = 1,
  status = c("new", "waiting"),
  limit = FALSE,
  timeout = 4
)

## S4 method for signature 's3_bucket'
manifest_yaml_to_bash(
  base,
  project,
  hash,
  shutdown = FALSE,
  split = 1,
  status = c("new", "waiting"),
  limit = FALSE,
  timeout = 4
)

## S4 method for signature 'character'
manifest_yaml_to_bash(
  base,
  project,
  hash,
  shutdown = FALSE,
  split = 1,
  status = c("new", "waiting"),
  limit = FALSE
)
```

Arguments

base	the base location to store the manifest
project	will be a relative path within the base location

hash	Fingerprint of the manifest `yaml`file.
shutdown	Append a shutdown command at the end of the script. Defaults to `FALSE`.
split	Number of scripts over which to splits the analyses. Default to 1.
status	A vector with status levels naming the levels which should be calculated. Defaults to `c("new", "waiting")`.
limit	Limit bandwidth and CPU usage. Defaults to `FALSE`.
timeout	number of hours to time out the docker container. Defaults to 4.

---

mark_obsolete_dataset   *Find and mark obsolete datasets*

Description

A dataset is obsolete when a newer version is available

Usage

```
mark_obsolete_dataset(channel)
```

Arguments

channel	An open ODBC channel

---

n2kAggregate-class   *The* n2kAggregate *class*

Description

It holds analysis data based on an aggregated imputation

Slots

RawImputed A `rawImputed` object with multiple imputations.

Function The function to apply on each group.

Filter The object to pass to the `filter` argument of multimput::aggregate_impute().

Join The object to pass to the `join` argument of multimput::aggregate_impute().

AggregatedImputed An `aggregatedImputed` object with multiple imputations.

---

n2kAnalysisMetadata-class
>   *The* n2kAnalysisMetadata *class*

Description

The n2kAnalysisMetadata class

---

n2kAnalysisVersion-class
>   *The* n2kAnalysisVersion *class*

Description

The n2kAnalysisVersion class

---

n2kAnomaly-class   *The* n2kAnomaly *class*

Description

The n2kAnomaly class

---

n2kComposite-class   *The* n2kComposite *class*

Description

Calculate composite indices from multiple analysis

Slots

Extractor A function to extract the relevant parameters from the model.

Parameter A data.frame with the relevant parameter estimates of each parent analysis.

Index The composite index based on the parameters.

---

n2kContrast-class   *The* n2kContrast *class*

Description

The n2kContrast class

---

n2kHurdleImputed-class

                              The n2kHurdleImputed *class*

Description

It holds the model2 of hurdle imputed data.

Slots

Presence A rawImputed object with the presence component.

Count A rawImputed object with the count component.

Hurdle A aggregatedImputed object with the resulting hurdle object.

---

n2kImport-class   *The* n2kImport *class*

Description

It hold the imported datasets

Slots

Dataset A data.frame with the datasets.

---

n2kInla-class   *The* n2kInla *class*

Description

It hold analysis data based on an INLA Poisson model

Slots

Data A data.frame with the data.

LinearCombination An optional matrix with the linear combinations.

ReplicateName An optional list with names of replicates.

Model Either NULL or the resulting INLA model.

Family The family of the INLA model.

Control A named list with options passed to the arguments of INLA::inla().

ImputationSize The number of multiple imputations. Defaults to 0, indication no multiple imputation.

Minimum An optional string containing the name of the variable in Data holding the minimal values for imputation.

RawImputed A rawImputed object with multiple imputations.

Extra A data.frame with extra data to add to the imputations. This data is not used in the imputation model. It must contain the same variables as the original data.

---

n2kInlaComparison-class

                          The n2kInlaComparison *class*

Description

Compare multiple models using the WAIC criterion

Slots

WAIC A data.frame with WAIC values per model.

---

n2kManifest-class   *The* n2kManifest *class*

Description

The n2kManifest class

---

n2kModel-class   *The* n2kModel *class*

Description

A virtual superclass to contain the analysis models

---

n2kModelImputed-class   *The* n2kModelImputed *class*

Description

It holds the model of aggregated imputed data

Slots

Function The object to pass to the model.fun argument of multimput::model_impute().

Package A vector of package names which must be loaded to run the function.

ModelArgs The object to pass to the model.args argument of multimput::model_impute().

PrepareModelArgs An optional list containing a single function that will be applied to the object. The result of the function will be appended to the ModelsArgs.

Extractor The object to pass to the extractor argument of multimput::model_impute().

ExtractorArgs The object to pass to the extractor.args argument of multimput::model_impute().

Filter The object to pass to the filter argument of multimput::model_impute().

Mutate The object to pass to the mutate argument of multimput::model_impute().

AggregatedImputed An aggregatedImputed object with multiple imputations.

Results The data.frame with the results of multimput::model_impute().

---

n2kParameter-class   *The* n2kParameter *class*

Description

The n2kParameter class

---

n2kResult-class   *The* n2kResult *class*

Description

The n2kResult class

---

n2k_aggregate   *Create an* n2kAggregate *object*

Description

A new n2kAggregate model.

Usage

```
n2k_aggregate(...)

## S4 method for signature 'ANY'
n2k_aggregate(...)
```

Arguments

...   other arguments. See below

Details

- scheme_id: a string holding the id of the scheme.

- species_group_id: a string identifying the species group.

- location_group_id: a string identifying the location group.

- model_type: a string identifying the type of model to fit to the data.

- first_imported_year: Oldest year considered in the data.

- last_imported_year: Most recent year considered in the data.

- duration: The width of the moving window. Defaults to the last_imported_year – first_imported_year + 1.

- last_analysed_year: Most recent year in the window. Defaults to last_imported_year.

- analysis_date: A POSIXct date indicating the date that the dataset was imported.

- seed: A single integer used as a seed for all calculations. A random seed will be inserted when missing.

- fun: The function to apply when aggregating.

- filter: An optional list for the filter argument of multimput::aggregate_impute().

- join: An optional list for the join argument of multimput::aggregate_impute().

n2k_composite   *Create an* n2kComposite *object*

Description

A new n2kComposite model is created when parent is a data.frame.

Usage

```
n2k_composite(parent_status, ...)

## S4 method for signature 'data.frame'
n2k_composite(
  parent_status,
  status = "waiting",
  result_datasource_id,
  scheme_id,
  formula,
  species_group_id,
  location_group_id,
  model_type,
  first_imported_year,
  last_imported_year,
  duration,
  last_analysed_year,
  analysis_date,
  extractor,
  ...,
  seed
)
```

Arguments

parent_status   A data.frame with columns parent_analysis (the file fingerprint of the par-
                 ent), parentstatus_fingerprint (the status fingerprint of the parent), and
                 parent_status (the status of the parent).

...   other arguments

status   A single character indicating the status of the model. Defaults to "waiting".

result_datasource_id
                 A string identifying the data source.

scheme_id   A single integer holding the id of the scheme.

formula   A single character identifying the comparison.

species_group_id
                 A string identifying the species group.

```
location_group_id
```
A string identifying the location group.

`model_type`   The type of the models. Must start with `"inla comparison:"`.

```
first_imported_year
```
Oldest year considered in the data.

```
last_imported_year
```
Most recent year considered in the data.

`duration`   The width of the moving window. Defaults to the `last_imported_year` - `first_imported_year` + 1.

```
last_analysed_year
```
Most recent year in the window. Defaults to `last_imported_year`.

`analysis_date` A `POSIXct` date indicating the date that the dataset was imported.

`extractor`   a function to extract the relevant parameters from the model

`seed`   A single integer used as a seed for all calculations. A random seed will be inserted when missing.

---

n2k_hurdle_imputed   *Create an* n2kHurdleImputed *object*

Description

A new n2kHurdleImputed model.

Usage

```
n2k_hurdle_imputed(presence, count, verbose = FALSE)

## S4 method for signature 'n2kInla'
n2k_hurdle_imputed(presence, count, verbose = FALSE)
```

Arguments

`presence`   the n2kInla object for the presence model.

`count`   the n2kInla object for the count model.

`verbose`   display the location group ID and species group ID. Defaults to `FALSE`.

n2k_import   *Create an* n2kImport *object*

Description

A new n2kImport model.

Usage

```
n2k_import(...)

## S4 method for signature 'ANY'
n2k_import(...)
```

Arguments

...   other arguments. See below

Details

- scheme_id: a string holding the id of the scheme.

- species_group_id: a string identifying the species group.

- location_group_id: a string identifying the location group.

- model_type: a string identifying the type of model to fit to the data.

- first_imported_year: Oldest year considered in the data.

- last_imported_year: Most recent year considered in the data.

- duration: The width of the moving window. Defaults to the last_imported_year - first_imported_year + 1.

- last_analysed_year: Most recent year in the window. Defaults to last_imported_year.

- analysis_date: A POSIXct date indicating the date that the dataset was imported.

- seed: A single integer used as a seed for all calculations. A random seed will be inserted when missing.

- formula: a string holding the model formula.

- dataset: A data.frame with filename, fingerprint and import_date.

n2k_inla	*Create an* n2kInla *object*

Description

A new n2kInla model is created when data is a data.frame.

In case data is an n2kInla object, then only the model and status are updated. All other slots are unaffected.

Usage

```
n2k_inla(data, ..., model_fit)

## S4 method for signature 'data.frame,ANY'
n2k_inla(
  data,
  status = "new",
  result_datasource_id,
  scheme_id,
  family = "poisson",
  formula,
  species_group_id,
  location_group_id,
  model_type,
  first_imported_year,
  last_imported_year,
  duration,
  last_analysed_year,
  analysis_date,
  lin_comb = NULL,
  minimum = "",
  imputation_size,
  parent = character(0),
  seed,
  replicate_name = list(),
  control = list(),
  parent_status = "converged",
  parent_statusfingerprint,
  extra,
  ...,
  model_fit
)

## S4 method for signature 'n2kInla,inla'
n2k_inla(data, status, raw_imputed = NULL, ..., model_fit)
```

Arguments

`data`	a `data.frame` with the data to analyse
`...`	other arguments. See below
`model_fit`	The fitted model
`status`	A single character indicating the status of the model. Defaults to `"waiting"`.
`result_datasource_id`	A string identifying the data source.
`scheme_id`	A single integer holding the id of the scheme.
`family`	the family to use in the INLA model.
`formula`	A single character identifying the comparison.
`species_group_id`	A string identifying the species group.
`location_group_id`	A string identifying the location group.
`model_type`	The type of the models. Must start with `"inla comparison:"`.
`first_imported_year`	Oldest year considered in the data.
`last_imported_year`	Most recent year considered in the data.
`duration`	The width of the moving window. Defaults to the `last_imported_year` − `first_imported_year` + 1.
`last_analysed_year`	Most recent year in the window. Defaults to `last_imported_year`.
`analysis_date`	A `POSIXct` date indicating the date that the dataset was imported.
`lin_comb`	A model matrix to calculate linear combinations.
`minimum`	The name of the variable which holds the minimum counts. Only relevant in case of multiple imputation.
`imputation_size`	The required number of imputations defaults to 0.
`parent`	The file fingerprint of the optional parent analysis.
`seed`	A single integer used as a seed for all calculations. A random seed will be inserted when missing.
`replicate_name`	A list with the names of replicates. Defaults to an empty list. Used in case of `f(X, ..., replicate = Z)`. Should be a named list like e.g. `list(X = c("a", "b", "c"))`.
`control`	A named list passed to [`INLA::inla()`](#) when fitting the model.
`parent_status`	The status of the parent analysis.
`parent_statusfingerprint`	The status fingerprint of the parent analysis.
`extra`	a `data.frame` with extra observations not used in the model. They will be added in subsequent analyses.
`raw_imputed`	the optional `rawImputed` object

---

n2k_inla_comparison   *Create an* n2kInlaComparison *object*

Description

A new n2kInlaComparison model is created when parent is a character.

Usage

```
n2k_inla_comparison(parent_status, ...)

## S4 method for signature 'data.frame'
n2k_inla_comparison(
  parent_status,
  status = "waiting",
  result_datasource_id,
  scheme_id,
  formula,
  species_group_id,
  location_group_id,
  model_type,
  first_imported_year,
  last_imported_year,
  duration,
  last_analysed_year,
  analysis_date,
  ...,
  seed
)
```

Arguments

parent_status   A data.frame with columns parent_analysis (the file fingerprint of the parent), parentstatus_fingerprint (the status fingerprint of the parent), and parent_status (the status of the parent).

...   other arguments

status   A single character indicating the status of the model. Defaults to "waiting".

result_datasource_id
                A string identifying the data source.

scheme_id   A single integer holding the id of the scheme.

formula   A single character identifying the comparison.

species_group_id
                A string identifying the species group.

location_group_id
                A string identifying the location group.

model_type	The type of the models. Must start with `"inla comparison:"`.
first_imported_year	
	Oldest year considered in the data.
last_imported_year	
	Most recent year considered in the data.
duration	The width of the moving window. Defaults to the `last_imported_year` - `first_imported_year` + 1.
last_analysed_year	
	Most recent year in the window. Defaults to `last_imported_year`.
analysis_date	A `POSIXct` date indicating the date that the dataset was imported.
seed	A single integer used as a seed for all calculations. A random seed will be inserted when missing.

---

n2k_manifest   *Create an* n2kManifest *object*

Description

A new n2kManifest object is created when manifest is a data.frame.

Usage

```
n2k_manifest(manifest)

## S4 method for signature 'data.frame'
n2k_manifest(manifest)
```

Arguments

manifest   a data.frame with the manifest content

---

n2k_model_imputed   *Create an* n2kModelImputed *object*

Description

A new n2kModelImputed model.

Usage

```
n2k_model_imputed(...)

## S4 method for signature 'ANY'
n2k_model_imputed(...)
```

Arguments

...	other arguments. See below

Details

- scheme_id: a string holding the id of the scheme.
- species_group_id: a string identifying the species group.
- location_group_id: a string identifying the location group.
- model_type: a string identifying the type of model to fit to the data.
- first_imported_year: Oldest year considered in the data.
- last_imported_year: Most recent year considered in the data.
- duration: The width of the moving window. Defaults to the last_imported_year – first_imported_year + 1.
- last_analysed_year: Most recent year in the window. Defaults to last_imported_year.
- analysis_date: A POSIXct date indicating the date that the dataset was imported.
- seed: A single integer used as a seed for all calculations. A random seed will be inserted when missing.

- model_fun: The model_fun argument of multimput::model_impute().
- package: A character vector of package names which must be loaded for model_fun.
- model_args: An optional list for the model_args argument of multimput::model_impute().
- extractor: An optional list for the extractor argument of multimput::model_impute().
- extractor_args: An optional list for the extractor_args argument of multimput::model_impute().
- filter: An optional list for the filter argument of multimput::model_impute().
- mutate: An optional list for the 'mutate" argument of multimput::model_impute().

---

parent_status	*Get the parent status of an* n2kModel

Description

Get the parent status of an n2kModel

Usage

```
parent_status(x)

## S4 method for signature 'n2kAnalysisMetadata'
parent_status(x)
```

Arguments

x	the n2kModel object

Value

the parent status of the object

---

parent_status<- *Overwrite the status of an* n2kAnalysisMetadata

---

Description

Overwrite the status of an n2kAnalysisMetadata

Usage

```
parent_status(x) <- value

## S4 replacement method for signature 'n2kComposite'
parent_status(x) <- value
```

Arguments

x	the n2kAnalysisMetadata object
value	the new values for the status

---

read_manifest *Read a* n2kManifest *object*

Description

Read a n2kManifest object

Usage

```
read_manifest(base, project, hash)

## S4 method for signature 'character'
read_manifest(base, project, hash)

## S4 method for signature 's3_bucket'
read_manifest(base, project, hash)
```

Arguments

base	The base location to read the manifest.
project	Will be a relative path within the base location.
hash	Optional the sha1 of the manifest. This can be abbreviated to to first unique characters. The function will return an error in case of multiple matches. If missing, then most recent manifest will be returned.

---

read_model   *Read an* n2kModel *object*

Description

Read an n2kModel object

Usage

```
read_model(x, base, project)

## S4 method for signature 'ANY,character'
read_model(x, base, project)

## S4 method for signature 'ANY,s3_bucket'
read_model(x, base, project)

## S4 method for signature 'ANY,ANY'
read_model(x, base, project)
```

Arguments

x	the file fingerprint of the n2kModel
base	the base location to read the model
project	will be a relative path within the base location

---

read_result   *Read an* n2kResult *object*

Description

Read an n2kResult object

Usage

```
read_result(x, base, project)

## S4 method for signature 'ANY,character'
read_result(x, base, project)

## S4 method for signature 'ANY,s3_bucket'
read_result(x, base, project)

## S4 method for signature 'ANY,ANY'
read_result(x, base, project)
```

Arguments

x	the file fingerprint of the n2kResult
base	the base location to read the results
project	will be a relative path within the base location

---

result_estimate	*Aggregate all results estimates in a single dataframe*

Description

Aggregate all results estimates in a single dataframe

Usage

```
result_estimate(x, ...)

## S4 method for signature 'character'
result_estimate(x, ..., base, project)

## S4 method for signature 'n2kResult'
result_estimate(x, ...)
```

Arguments

x	object with the current results
...	further arguments (see Details)
base	the base location to read the results
project	will be a relative path within the base location

---

result_metadata	*Aggregate all results meta data in a single dataframe*

Description

Aggregate all results meta data in a single dataframe

Usage

```
result_metadata(x, ...)

## S4 method for signature 'character'
result_metadata(x, ..., base, project)

## S4 method for signature 'n2kResult'
result_metadata(x, ...)
```

Arguments

x	object with the current results
...	further arguments (see Details)
base	the base location to read the results
project	will be a relative path within the base location

---

select_factor_count_strictly_positive

Select data based on the number of presences per category

Description

Presences have Count > 0.

Usage

```
select_factor_count_strictly_positive(
  observation,
  variable,
  threshold,
  relative = FALSE,
  dimension = 1
)
```

Arguments

observation	the data.frame with observations
variable	the name of the factor
threshold	the minimal threshold
relative	When FALSE the threshold is the number of non-zero observations. When TRUE the threshold is the proportion of non-zero observations. Defaults to FALSE.
dimension	Indicates which element of variable is used for the final aggregation.

Examples

```
observation <- data.frame(
  Count = c(4, 4, 4, 4, 3, 3, 3, 0, 2, 2, 0, 0),
  LocationID = rep(1:3, each = 4),
  Year = rep(c(1, 1, 1, 1, 2, 2), 2)
)
# Select the locations with at least 3 prescenses
select_factor_count_strictly_positive(
  observation,
  variable = "LocationID",
  threshold = 3
```

```
)
# Select those locations in which the species is present in at least 2 years
select_factor_count_strictly_positive(
  observation, variable = c("LocationID", "Year"), threshold = 2
)
# Select those years in which the species is present in at least 2 locations
select_factor_count_strictly_positive(
  observation, variable = c("LocationID", "Year"),
  threshold = 2,
  dimension = 2
)
```

---

select_factor_threshold

Select the observations based on the average of a factor

Description

The negative binomial average of the `Count` variable is calculated for each level of `variable`. Only the levels which are equal or larger than `threshold` times the maximal average (in the original scale) are retained.

Usage

```
select_factor_threshold(observation, variable, threshold)
```

Arguments

observation   the `data.frame` with observations

variable   the name of the `factor`

threshold   the minimal threshold

Examples

```
observation <- data.frame(
  Count = c(100, 101, 50, 51, 1, 0, 0, 0),
  LocationID = factor(rep(1:4, each = 2))
)
select_factor_threshold(observation, "LocationID", threshold = 0.05)
```

---

select_observed_range   *Limit the observation to the range in which the species is present*

Description

Limit the observation to the range in which the species is present

Usage

```
select_observed_range(observation, variable)
```

Arguments

observation   the `data.frame` with observations

variable   the name of the `factor`

Examples

```
observation <- data.frame(
  Count = c(0, 0, 100, 101, 0, 51, 1, 0, 0, 0),
  Year = 1:10
)
select_observed_range(observation, "Year")
```

---

session_package   *Convert a* sessionInfo() *to a data.frame of packages*

Description

Convert a `sessionInfo()` to a data.frame of packages

Usage

```
session_package(session)

## S4 method for signature 'sessionInfo'
session_package(session)
```

Arguments

session   The output of `sessionInfo()`

Value

a data.frame with the packages of a `sessionInfo()`

---

status   *Get the status of an* n2kModel

Description

Get the status of an n2kModel

Usage

```
status(x)

## S4 method for signature 'n2kAnalysisMetadata'
status(x)

## S4 method for signature 'character'
status(x)
```

Arguments

x   the n2kModel object

Value

the status of the object

---

status<-   *Overwrite the status of an* n2kAnalysisMetadata

Description

Overwrite the status of an n2kAnalysisMetadata

Usage

```
status(x) <- value

## S4 replacement method for signature 'n2kInla'
status(x) <- value

## S4 replacement method for signature 'n2kComposite'
status(x) <- value

## S4 replacement method for signature 'n2kInlaComparison'
status(x) <- value

## S4 replacement method for signature 'n2kAggregate'
```

```
status(x) <- value

## S4 replacement method for signature 'n2kModelImputed'
status(x) <- value

## S4 replacement method for signature 'n2kHurdleImputed'
status(x) <- value
```

Arguments

x	the n2kAnalysisMetadata object
value	the new values for the status

---

store_manifest *Store an* n2kManifest *object*

Description

Store an n2kManifest object

Usage

```
store_manifest(x, base, project)

## S4 method for signature 'ANY,character'
store_manifest(x, base, project)

## S4 method for signature 'ANY,s3_bucket'
store_manifest(x, base, project)
```

Arguments

x	the n2kManifest
base	the base location to store the manifest
project	will be a relative path within the base location

---

store_manifest_yaml   *Store a Docker configuration*

Description

Store a Docker configuration

Usage

```
store_manifest_yaml(x, base, project, docker, dependencies)

## S4 method for signature 'ANY,s3_bucket'
store_manifest_yaml(x, base, project, docker, dependencies)

## S4 method for signature 'ANY,character'
store_manifest_yaml(x, base, project, docker, dependencies)
```

Arguments

x	the n2kManifest
base	the base location to store the manifest
project	will be a relative path within the base location
docker	the docker image to use
dependencies	extra GitHub packages to install

---

store_model   *Store an* n2kModel *object*

Description

Store an n2kModel object

Usage

```
store_model(x, base, project, overwrite = TRUE, validate = TRUE)

## S4 method for signature 'ANY,character'
store_model(x, base, project, overwrite = TRUE, validate = TRUE)

## S4 method for signature 'ANY,s3_bucket'
store_model(x, base, project, overwrite = TRUE, validate = TRUE)
```

Arguments

x	The n2kModel.
base	The base location to store the model.
project	Will be a relative path within the base location.
overwrite	Should an existing object be overwritten? Defaults to TRUE.
validate	Check that the object is valid before storing it. Defaults to TRUE.

---

union	*Make a union of* AnalysisVersions

Description

Make a union of AnalysisVersions

Usage

```
union(...)
```

Arguments

...	the n2kAnalysisVersions to union

---

validObject,character-method
Get the validity of objects in files

Description

Get the validity of objects in files

Usage

```
## S4 method for signature 'character'
validObject(object, test = FALSE, complete = FALSE)
```

Arguments

object	a single filename, a single path or a vector with filenames.
test	See methods::validObject()
complete	See methods::validObject()

Index