

Package ‘soReta’

September 12, 2026

Title Ready-to-Analyze Datasets from Camera Trap Data

Version 0.1.0

Description Functions to build datasets ready for statistical analysis from camera trap data: GLMM/GAMM on counts/RAI at various temporal levels, group size, occupancy, kernel/circular analysis of activity patterns, temporal interactions between species, hierarchical diel models, and classic capture-mark-recapture. Input data must be formatted in the style produced by 'camtrapR', the standard convention in the field.

License GPL (>= 3)

Encoding UTF-8

Depends R (>= 4.1.0)

Imports dplyr, lubridate, tidyr

Config/roxygen2/version 8.1.0

Suggests camtrapR, GLMMadaptive, knitr, rmarkdown, testthat (>= 3.0.0)

Config/testthat/edition 3

VignetteBuilder knitr

LazyData true

URL <https://github.com/OrlandoTomassini/soReta>

BugReports <https://github.com/OrlandoTomassini/soReta/issues>

NeedsCompilation no

Author Orlando Tomassini [aut, cre] (ORCID:
<<https://orcid.org/0000-0002-6963-0903>>)

Maintainer Orlando Tomassini <orlando.tomassini@gmail.com>

Repository CRAN

Date/Publication 2026-09-12 14:10:03 UTC

Contents

as_capture_strings	2
build_cmr_block	3
build_cmr_day	4
build_day_total	5
build_diel_binomial_block	6
build_diel_binomial_month	7
build_diel_binomial_period	7
build_diel_day	8
build_group_day_total	9
build_group_size_block	10
build_group_size_day	12
build_group_size_events	13
build_group_size_month	14
build_group_size_period	15
build_group_size_total	16
build_occupancy_block	18
build_occupancy_day	19
build_site_block	19
build_site_day	21
build_site_month	22
build_site_period	23
build_site_total	24
build_species_pair_interruptions	25
build_species_pair_intervals	26
camOp_soReta	27
camtraps_soReta	28
extract_radians	29
radians_to_env	30
recordTableIndividuals_soReta	31
recordTable_soReta	32
Index	33

as_capture_strings	<i>Convert an individual x occasion table (or a list per species) into CMR capture-history strings</i>
--------------------	--

Description

Convert an individual x occasion table (or a list per species) into CMR capture-history strings

Usage

```
as_capture_strings(ch, na_symbol = ".")
```

Arguments

ch	a data.frame produced by build_cmr_day()/build_cmr_block() (Individual column + occasion columns), OR the whole list (one element per species) if you have not disabled speciesCol upstream.
na_symbol	symbol to use for missing occasions (NA). Default "." (MARK/RMark convention).

Value

if ch is a data.frame: a character vector with names = Individual, values = capture-history string (e.g. "1.010"). If ch is a list: a named list per species, same structure per element.

build_cmr_block	<i>Build an N-day-block individual x block capture history (classic CMR)</i>
-----------------	--

Description

Build an N-day-block individual x block capture history (classic CMR)

Usage

```
build_cmr_block(
  recordTable,
  camOp,
  block_days,
  individualCol = "Individual",
  speciesCol = "Species",
  stationCol = "Station",
  dateTimeCol = "DateTimeOriginal",
  min_days = 1
)
```

Arguments

recordTable	data.frame produced by camtrapR::recordTableIndividual().
camOp	effort matrix produced by cameraOperation().
block_days	block length in days (positive integer).
individualCol	name of the individual ID column (default "Individual").
speciesCol	name of the species column (default "Species"). If present in recordTable, the result is split by species into a named list. Pass NULL to disable this (a single data.frame).
stationCol	name of the station column (default "Station").
dateTimeCol	name of the date/time column (default "DateTimeOriginal").

min_days minimum number of overall active station-days in the block (summed across all stations) for it to be treated as a genuine 0 rather than NA (default 1). Only a lower-bound check (≥ 0) is enforced: unlike `build_site_block()`'s per-station `min_days`, this one is a total across however many stations `camOp` contains, so no single fixed upper bound applies universally.

Value

If `speciesCol` is present: named list, one `data.frame` per species. Otherwise: a single `data.frame`. Each `data.frame` has an `Individual` column and one column per block of the period covered by `camOp` ("YYYY-MM-DD–YYYY-MM-DD"), values 0/1/NA as described above.

<code>build_cmr_day</code>	<i>Build a daily individual x day capture history (classic CMR)</i>
----------------------------	---

Description

Build a daily individual x day capture history (classic CMR)

Usage

```
build_cmr_day(
  recordTable,
  camOp,
  individualCol = "Individual",
  speciesCol = "Species",
  stationCol = "Station",
  dateTimeCol = "DateTimeOriginal"
)
```

Arguments

<code>recordTable</code>	data.frame produced by <code>camtrapR::recordTableIndividual()</code> .
<code>camOp</code>	effort matrix produced by <code>cameraOperation()</code> .
<code>individualCol</code>	name of the individual ID column (default "Individual").
<code>speciesCol</code>	name of the species column (default "Species"). If present in <code>recordTable</code> , the result is split by species into a named list. Pass NULL to disable this (a single <code>data.frame</code>).
<code>stationCol</code>	name of the station column (default "Station").
<code>dateTimeCol</code>	name of the date/time column (default "DateTimeOriginal").

Value

If `speciesCol` is present: named list, one `data.frame` per species (`result[["SpeciesName"]]`). Otherwise: a single `data.frame`. Each `data.frame` has an `Individual` column and one column per truly active day covered by `camOp` ("YYYY-MM-DD"), values 0/1 only (see Details – no NA case at this daily grain).

build_day_total	<i>Build a dataset with 1 row per calendar day, aggregating across all stations</i>
-----------------	---

Description

Build a dataset with 1 row per calendar day, aggregating across all stations

Usage

```
build_day_total(
  recordTable,
  camOp,
  stationCol = "Station",
  speciesCol = "Species",
  dateTimeCol = "DateTimeOriginal",
  threshold_min = 30,
  independence_method = c("chain", "window"),
  require_uninterrupted = FALSE
)
```

Arguments

recordTable	camtrapR-style data.frame (e.g. output of recordTable()).
camOp	effort matrix produced by cameraOperation().
stationCol	name of the station column in recordTable (default "Station").
speciesCol	name of the species column in recordTable (default "Species").
dateTimeCol	name of the date/time column in recordTable (default "DateTimeOriginal").
threshold_min	independence threshold in minutes (default 30).
independence_method	how consecutive photos of the same species are collapsed into one independent event: "chain" (default) or "window". See build_site_block() docs for details.
require_uninterrupted	apply O'Brien et al. (2003)'s third independence criterion (default FALSE). See build_site_block() docs.

Value

data.frame with columns Date, n_stations_active (number of stations active that day), N_sp (number of species detected that day, across all stations), and for each species two columns: "<species>_N" (sum of independent events across all stations active that day) and "<species>_RAI" (= $N / n_stations_active * 100$, rounded to 2 decimals).

Note

Species names in the columns are cleaned (spaces and non-alphanumeric characters replaced by "_") via the internal .sanitize_species_names().

```
build_diel_binomial_block
```

Build a binomial site x N-day-block x time-bin x species dataset

Description

Build a binomial site x N-day-block x time-bin x species dataset

Usage

```
build_diel_binomial_block(
  recordTable,
  camOp,
  block_days,
  stationCol = "Station",
  speciesCol = "Species",
  dateTimeCol = "DateTimeOriginal",
  bin_hours = 1,
  min_days = NULL
)
```

Arguments

<code>recordTable</code>	camtrapR-style data.frame.
<code>camOp</code>	effort matrix produced by <code>cameraOperation()</code> .
<code>block_days</code>	block length in days (positive integer).
<code>stationCol, speciesCol, dateTimeCol</code>	names of the relevant columns.
<code>bin_hours</code>	time-bin width, in hours (default 1).
<code>min_days</code>	if specified (0 to <code>block_days</code>), keeps only site x block with <code>n_days_active</code> $\geq$ <code>min_days</code> . Default NULL.

Value

data.frame with `Station`, `block_start`, `block_end`, `n_days_active`, `Time`, `Species`, `success`, `failure`, `Site_Block`.

```
build_diel_binomial_month
```

Build a binomial site x month x time-bin x species dataset

Description

Build a binomial site x month x time-bin x species dataset

Usage

```
build_diel_binomial_month(
  recordTable,
  camOp,
  stationCol = "Station",
  speciesCol = "Species",
  dateTimeCol = "DateTimeOriginal",
  bin_hours = 1,
  min_days = NULL
)
```

Arguments

<code>recordTable</code>	camtrapR-style data.frame.
<code>camOp</code>	effort matrix produced by <code>cameraOperation()</code> .
<code>stationCol</code> , <code>speciesCol</code> , <code>dateTimeCol</code>	names of the relevant columns.
<code>bin_hours</code>	time-bin width, in hours (default 1).
<code>min_days</code>	if specified (0 to 31), keeps only site x month with <code>n_days_active</code> $\geq$ <code>min_days</code> . Default NULL.

Value

data.frame with `Station`, `sampling_event` ("YYYY-MM"), `n_days_active`, `Time`, `Species`, `success`, `failure`, `Site_Event`.

```
build_diel_binomial_period
```

Build a binomial site x period/season x time-bin x species dataset

Description

Build a binomial site x period/season x time-bin x species dataset

Usage

```
build_diel_binomial_period(
  recordTable,
  camOp,
  period_names,
  period_starts,
  stationCol = "Station",
  speciesCol = "Species",
  dateTimeCol = "DateTimeOriginal",
  bin_hours = 1,
  min_days = NULL
)
```

Arguments

recordTable	camtrapR-style data.frame.
camOp	effort matrix produced by cameraOperation().
period_names	vector of period labels, e.g. c("winter", "spring", "summer", "autumn").
period_starts	vector of "DD/MM/YYYY" dates, same length as period_names – start of each period, recurring every year.
stationCol, speciesCol, dateTimeCol	names of the relevant columns.
bin_hours	time-bin width, in hours (default 1).
min_days	if specified, keeps only site x period with n_days_active >= min_days. Default NULL. Only a lower-bound check (>= 0) is enforced: unlike the monthly version's fixed 31-day ceiling, a period's length is entirely user-defined.

Value

data.frame with Station, period_name, year, period_start ("DD/MM"), period_end ("DD/MM"), n_days_active, Time, Species, success, failure, Site_Period.

build_diel_day	<i>Build a non-aggregated, day-by-day site x day x time-bin x species dataset</i>
----------------	---

Description

Build a non-aggregated, day-by-day site x day x time-bin x species dataset

Usage

```
build_diel_day(
  recordTable,
  camOp,
  stationCol = "Station",
  speciesCol = "Species",
  dateTimeCol = "DateTimeOriginal",
  bin_hours = 1,
  tz = "UTC"
)
```

Arguments

recordTable	camtrapR-style data.frame (e.g. output of recordTable()).
camOp	effort matrix produced by cameraOperation().
stationCol	name of the station column (default "Station").
speciesCol	name of the species column (default "Species").
dateTimeCol	name of the date/time column (default "DateTimeOriginal").
bin_hours	time-bin width, in hours (default 1). Must divide 24 exactly (e.g. 1, 2, 3, 4, 6, 0.5, 0.25).
tz	time zone of the stations, used to build the "date" column (default "UTC").

Value

data.frame with columns Station, Date (Date class, real day, always a truly active day for that station), date (POSIXct class, local noon of Date in the tz time zone. Time (start of the time bin, 0-24), Species, detected (0/1).

build_group_day_total	<i>Build a dataset with 1 row per calendar day, with group size statistics across all stations</i>
-----------------------	--

Description

Build a dataset with 1 row per calendar day, with group size statistics across all stations

Usage

```
build_group_day_total(
  recordTable,
  camOp,
  countCol,
  stationCol = "Station",
  speciesCol = "Species",
  dateTimeCol = "DateTimeOriginal",
```

```

threshold_min = 30,
independence_method = c("chain", "window"),
require_uninterrupted = FALSE
)

```

Arguments

recordTable	camtrapR-style data.frame (e.g. output of recordTable()).
camOp	effort matrix produced by cameraOperation().
countCol	name of the column in recordTable holding the number of animals per photo (e.g. "N_individuals").
stationCol	name of the station column in recordTable (default "Station").
speciesCol	name of the species column in recordTable (default "Species").
dateTimeCol	name of the date/time column in recordTable (default "DateTimeOriginal").
threshold_min	independence threshold in minutes (default 30).
independence_method	how consecutive photos of the same species are collapsed into one independent event: "chain" (default) or "window". See build_site_block() docs for details.
require_uninterrupted	apply O'Brien et al. (2003)'s third independence criterion (default FALSE). See build_site_block() docs.

Value

data.frame with columns Date, n_stations_active (number of stations active that day), N_sp (number of species detected that day, across all stations), and for each species four columns: "<species>_mean_group_size", "<species>_max_group_size", "<species>_sum_group_size", "<species>_RAI_individuals" (= sum_group_size / n_stations_active * 100, rounded to 2 decimals). NA (not 0) if the species had no events, at any station, that day.

Note

Species names in the columns are cleaned (spaces and non-alphanumeric characters replaced by "_") via the internal .sanitize_species_names().

build_group_size_block

Build a site x N-day-block dataset with group size statistics per species

Description

Build a site x N-day-block dataset with group size statistics per species

Usage

```
build_group_size_block(
  recordTable,
  camOp,
  countCol,
  block_days,
  stationCol = "Station",
  speciesCol = "Species",
  dateTimeCol = "DateTimeOriginal",
  threshold_min = 30,
  min_days = NULL,
  independence_method = c("chain", "window"),
  require_uninterrupted = FALSE
)
```

Arguments

recordTable	camtrapR-style data.frame (e.g. output of recordTable()).
camOp	effort matrix produced by cameraOperation().
countCol	name of the column in recordTable holding the number of animals per photo (e.g. "N_individuals").
block_days	block length in days (positive integer).
stationCol	name of the station column in recordTable (default "Station").
speciesCol	name of the species column in recordTable (default "Species").
dateTimeCol	name of the date/time column in recordTable (default "DateTimeOriginal").
threshold_min	independence threshold in minutes (default 30).
min_days	if specified (0 to block_days), keeps only rows with n_days_active >= min_days. Default NULL = no filter.
independence_method	how consecutive photos of the same species are collapsed into one independent event: "chain" (default) or "window". See build_site_block() docs for details.
require_uninterrupted	apply O'Brien et al. (2003)'s third independence criterion (default FALSE). See build_site_block() docs.

Value

data.frame with columns Station, block_start, block_end, n_days_active, N_sp (number of species with at least one event in the block), and for each species four columns: "<species>_mean_group_size", "<species>_max_group_size", "<species>_sum_group_size", and "<species>_RAI_individuals" (= sum_group_size / n_days_active * 100, rounded to 2 decimals). NA (not 0) if the species had no events in that block.

Note

Species names in the columns are cleaned (spaces and non-alphanumeric characters replaced by "_") via the internal .sanitize_species_names().

build_group_size_day *Build a site x day dataset with group size statistics per species*

Description

Build a site x day dataset with group size statistics per species

Usage

```
build_group_size_day(
  recordTable,
  camOp,
  countCol,
  stationCol = "Station",
  speciesCol = "Species",
  dateTimeCol = "DateTimeOriginal",
  threshold_min = 30,
  independence_method = c("chain", "window"),
  require_uninterrupted = FALSE
)
```

Arguments

recordTable	camtrapR-style data.frame (e.g. output of recordTable()).
camOp	effort matrix produced by cameraOperation().
countCol	name of the column in recordTable holding the number of animals per photo (e.g. "N_individuals").
stationCol	name of the station column in recordTable (default "Station").
speciesCol	name of the species column in recordTable (default "Species").
dateTimeCol	name of the date/time column in recordTable (default "DateTimeOriginal").
threshold_min	independence threshold in minutes (default 30).
independence_method	how consecutive photos of the same species are collapsed into one independent event: "chain" (default) or "window". See build_site_block() docs for details.
require_uninterrupted	apply O'Brien et al. (2003)'s third independence criterion (default FALSE). See build_site_block() docs.

Value

data.frame with columns Station, Date, N_sp (number of species with at least one event that day), and for each species present in recordTable three columns: "<species>_mean_group_size", "<species>_max_group_size", "<species>_sum_group_size" – computed across that species' independent events on that day. NA (not 0) if the species had no events that day. Includes only the days on which the station was active in camOp.

Note

Species names in the columns are cleaned (spaces and non-alphanumeric characters replaced by "_") via the internal `.sanitize_species_names()`.

`build_group_size_events`

Build a dataset with 1 row per independent event, including group size

Description

Build a dataset with 1 row per independent event, including group size

Usage

```
build_group_size_events(
  recordTable,
  camOp,
  countCol,
  stationCol = "Station",
  speciesCol = "Species",
  dateTimeCol = "DateTimeOriginal",
  threshold_min = 30,
  independence_method = c("chain", "window"),
  require_uninterrupted = FALSE
)
```

Arguments

<code>recordTable</code>	camtrapR-style data.frame (e.g. output of <code>recordTable()</code>).
<code>camOp</code>	effort matrix produced by <code>cameraOperation()</code> .
<code>countCol</code>	name of the column in <code>recordTable</code> holding the number of animals per photo (e.g. "N_individuals").
<code>stationCol</code>	name of the station column in <code>recordTable</code> (default "Station").
<code>speciesCol</code>	name of the species column in <code>recordTable</code> (default "Species").
<code>dateTimeCol</code>	name of the date/time column in <code>recordTable</code> (default "DateTimeOriginal").
<code>threshold_min</code>	independence threshold in minutes (default 30).
<code>independence_method</code>	how consecutive photos of the same species are collapsed into one independent event: "chain" (default) or "window". See <code>build_site_block()</code> docs for details.
<code>require_uninterrupted</code>	apply O'Brien et al. (2003)'s third independence criterion (default FALSE). See <code>build_site_block()</code> docs.

Value

data.frame with columns Station, Species, DateTime (start of the event's bout), Date (the calendar day of DateTime), Time ("HH:MM:SS", the time of day of DateTime), group_size (the MAXIMUM value of countCol across all photos in that bout – never the sum, to avoid double-counting the same individuals photographed more than once within one passage). Includes only events that fell on a truly active day for that station.

build_group_size_month

Build a site x month dataset with group size statistics per species

Description

Build a site x month dataset with group size statistics per species

Usage

```
build_group_size_month(
  recordTable,
  camOp,
  countCol,
  stationCol = "Station",
  speciesCol = "Species",
  dateTimeCol = "DateTimeOriginal",
  threshold_min = 30,
  min_days = NULL,
  independence_method = c("chain", "window"),
  require_uninterrupted = FALSE
)
```

Arguments

recordTable	camtrapR-style data.frame (e.g. output of recordTable()).
camOp	effort matrix produced by cameraOperation().
countCol	name of the column in recordTable holding the number of animals per photo (e.g. "N_individuals").
stationCol	name of the station column in recordTable (default "Station").
speciesCol	name of the species column in recordTable (default "Species").
dateTimeCol	name of the date/time column in recordTable (default "DateTimeOriginal").
threshold_min	independence threshold in minutes (default 30).
min_days	if specified (0 to 31), keeps only rows with n_days_active greater than or equal to min_days. Default NULL = no filter. NB: the possible maximum varies month by month (see the n_days_in_month column) – 31 is only the theoretical upper bound.

independence_method

how consecutive photos of the same species are collapsed into one independent event: "chain" (default) or "window". See build_site_block() docs for details.

require_uninterrupted

apply O'Brien et al. (2003)'s third independence criterion (default FALSE). See build_site_block() docs.

Value

data.frame with columns Station, year, month, month_start, month_end, n_days_active, n_days_in_month, N_sp (number of species with at least one event that month), and for each species four columns: "<species>_mean_group_size", "<species>_max_group_size", "<species>_sum_group_size", "<species>_RAI_individual" (= sum_group_size / n_days_active * 100, rounded to 2 decimals). NA (not 0) if the species had no events that month.

Note

Species names in the columns are cleaned (spaces and non-alphanumeric characters replaced by "_") via the internal .sanitize_species_names().

build_group_size_period

Build a site x period/season dataset with group size statistics per species

Description

Build a site x period/season dataset with group size statistics per species

Usage

```
build_group_size_period(
  recordTable,
  camOp,
  countCol,
  period_names,
  period_starts,
  stationCol = "Station",
  speciesCol = "Species",
  dateTimeCol = "DateTimeOriginal",
  threshold_min = 30,
  independence_method = c("chain", "window"),
  require_uninterrupted = FALSE
)
```

Arguments

recordTable	camtrapR-style data.frame (e.g. output of recordTable()).
camOp	effort matrix produced by cameraOperation().
countCol	name of the column in recordTable holding the number of animals per photo (e.g. "N_individuals").
period_names	vector of period labels, e.g. c("winter", "spring", "summer", "autumn").
period_starts	vector of "DD/MM/YYYY" dates, same length as period_names – the start date of each period, recurring every year.
stationCol	name of the station column in recordTable (default "Station").
speciesCol	name of the species column in recordTable (default "Species").
dateTimeCol	name of the date/time column in recordTable (default "DateTimeOriginal").
threshold_min	independence threshold in minutes (default 30).
independence_method	how consecutive photos of the same species are collapsed into one independent event: "chain" (default) or "window". See build_site_block() docs for details.
require_uninterrupted	apply O'Brien et al. (2003)'s third independence criterion (default FALSE). See build_site_block() docs.

Value

data.frame with columns Station, period_name, year, period_start ("DD/MM", no year), period_end ("DD/MM", no year), period_length_days, n_days_active, N_sp (number of species with at least one event in that period), and for each species four columns: "<species>_mean_group_size", "<species>_max_group_size", "<species>_sum_group_size", "<species>_RAI_individuals" (= sum_group_size / n_days_active * 100, rounded to 2 decimals). NA (not 0) if the species had no events in that period.

Note

Species names in the columns are cleaned (spaces and non-alphanumeric characters replaced by "_") via the internal .sanitize_species_names().

build_group_size_total

Build a dataset with 1 row per site, with group size statistics over the entire activity period

Description

Build a dataset with 1 row per site, with group size statistics over the entire activity period

Usage

```
build_group_size_total(
  recordTable,
  camOp,
  countCol,
  stationCol = "Station",
  speciesCol = "Species",
  dateTimeCol = "DateTimeOriginal",
  threshold_min = 30,
  independence_method = c("chain", "window"),
  require_uninterrupted = FALSE
)
```

Arguments

recordTable	camtrapR-style data.frame (e.g. output of recordTable()).
camOp	effort matrix produced by cameraOperation().
countCol	name of the column in recordTable holding the number of animals per photo (e.g. "N_individuals").
stationCol	name of the station column in recordTable (default "Station").
speciesCol	name of the species column in recordTable (default "Species").
dateTimeCol	name of the date/time column in recordTable (default "DateTimeOriginal").
threshold_min	independence threshold in minutes (default 30).
independence_method	how consecutive photos of the same species are collapsed into one independent event: "chain" (default) or "window". See build_site_block() docs for details.
require_uninterrupted	apply O'Brien et al. (2003)'s third independence criterion (default FALSE). See build_site_block() docs.

Value

data.frame with columns Station, n_days_active, N_sp (number of species with at least one event over the entire period), and for each species four columns: "<species>_mean_group_size", "<species>_max_group_size", "<species>_sum_group_size", "<species>_RAI_individuals" (= sum_group_size / n_days_active * 100, rounded to 2 decimals). NA (not 0) if the species was never detected at that station.

Note

Species names in the columns are cleaned (spaces and non-alphanumeric characters replaced by "_") via the internal .sanitize_species_names().

build_occupancy_block *Build, for each species, a site x N-day-block detection history*

Description

Build, for each species, a site x N-day-block detection history

Usage

```
build_occupancy_block(
  recordTable,
  camOp,
  block_days,
  stationCol = "Station",
  speciesCol = "Species",
  dateTimeCol = "DateTimeOriginal",
  min_days = 1
)
```

Arguments

recordTable	camtrapR-style data.frame (e.g. output of recordTable()).
camOp	effort matrix produced by cameraOperation().
block_days	block length in days (positive integer). 1 = daily (equivalent to build_occupancy_day), 7 = weekly, etc.
stationCol	name of the station column in recordTable (default "Station").
speciesCol	name of the species column in recordTable (default "Species").
dateTimeCol	name of the date/time column in recordTable (default "DateTimeOriginal").
min_days	minimum number of active days in the block for it to be treated as a genuine 0 rather than NA (default 1).

Value

named list, one data.frame per species. Each data.frame has a Station column and one column per block of the period covered by camOp (column name = "YYYY-MM-DD-YYYY-MM-DD", block start and end), values 0/1/NA as described above.

build_occupancy_day	<i>Build, for each species, a daily site x day detection history</i>
---------------------	--

Description

Build, for each species, a daily site x day detection history

Usage

```
build_occupancy_day(
  recordTable,
  camOp,
  stationCol = "Station",
  speciesCol = "Species",
  dateTimeCol = "DateTimeOriginal"
)
```

Arguments

recordTable	camtrapR-style data.frame (e.g. output of recordTable()).
camOp	effort matrix produced by cameraOperation().
stationCol	name of the station column in recordTable (default "Station").
speciesCol	name of the species column in recordTable (default "Species").
dateTimeCol	name of the date/time column in recordTable (default "DateTimeOriginal").

Value

named list, one data.frame per species. Each data.frame has a Station column and one column per day of the period covered by camOp (column name = date "YYYY-MM-DD"), values 0/1/NA as described above.

build_site_block	<i>Build a site x N-day-block dataset with independent-event counts per species</i>
------------------	---

Description

Build a site x N-day-block dataset with independent-event counts per species

Usage

```

build_site_block(
  recordTable,
  camOp,
  block_days,
  stationCol = "Station",
  speciesCol = "Species",
  dateTimeCol = "DateTimeOriginal",
  threshold_min = 30,
  min_days = NULL,
  independence_method = c("chain", "window"),
  require_uninterrupted = FALSE
)

```

Arguments

<code>recordTable</code>	camtrapR-style data.frame (e.g. output of <code>recordTable()</code>).
<code>camOp</code>	effort matrix produced by <code>cameraOperation()</code> .
<code>block_days</code>	block length in days (positive integer). 1 = daily, 7 = weekly, 8, 30, etc.
<code>stationCol</code>	name of the station column in <code>recordTable</code> (default "Station").
<code>speciesCol</code>	name of the species column in <code>recordTable</code> (default "Species").
<code>dateTimeCol</code>	name of the date/time column in <code>recordTable</code> (default "DateTimeOriginal").
<code>threshold_min</code>	independence threshold in minutes (default 30).
<code>min_days</code>	if specified (0 to <code>block_days</code>), keeps only rows with <code>n_days_active</code> $\geq$ <code>min_days</code> . Default NULL = no filter.
<code>independence_method</code>	how consecutive photos of the same species are collapsed into one independent event: "chain" (default, compares each photo to the previous one) or "window" (compares each photo to the start of the current bout; a bout can never last longer than <code>threshold_min</code>). See <code>.classify_events()</code> internal docs for the reasoning; "window" has no verified precedent in the camera-trapping literature.
<code>require_uninterrupted</code>	if TRUE, applies O'Brien et al. (2003)'s third independence criterion: two photos of the same species are only subject to the time threshold if no photo of ANY other species falls chronologically between them at that station; otherwise they are always independent. Default FALSE (matches every verified implementation checked in the literature).

Value

data.frame with columns `Station`, `block_start`, `block_end`, `mid_day` (day halfway between `block_start` and `block_end`, handy for joining external daily covariates), `n_days_active`, `N_sp` (number of species with at least one event in the block), and for each species present in `recordTable` two columns: "`<species>_N`" (independent-event count) and "`<species>_RAI`" (Relative Abundance Index = $\text{count} / \text{n_days_active} * 100$, rounded to 2 decimals).

Note

Species names in the columns are cleaned (spaces and non-alphanumeric characters replaced by "_") via the internal `.sanitize_species_names()`.

build_site_day	<i>Build a site x day dataset with independent-event counts per species</i>
----------------	---

Description

Build a site x day dataset with independent-event counts per species

Usage

```
build_site_day(
  recordTable,
  camOp,
  stationCol = "Station",
  speciesCol = "Species",
  dateTimeCol = "DateTimeOriginal",
  threshold_min = 30,
  independence_method = c("chain", "window"),
  require_uninterrupted = FALSE
)
```

Arguments

recordTable	camtrapR-style data.frame (e.g. output of <code>recordTable()</code>).
camOp	effort matrix produced by <code>cameraOperation()</code> .
stationCol	name of the station column in recordTable (default "Station").
speciesCol	name of the species column in recordTable (default "Species").
dateTimeCol	name of the date/time column in recordTable (default "DateTimeOriginal").
threshold_min	independence threshold in minutes (default 30).
independence_method	how consecutive photos of the same species are collapsed into one independent event: "chain" (default) or "window". See <code>build_site_block()</code> docs for details.
require_uninterrupted	apply O'Brien et al. (2003)'s third independence criterion (default FALSE). See <code>build_site_block()</code> docs.

Value

data.frame with columns Station, Date, N_sp (number of species with at least one event that day), and one "<species>_N" column (independent-event count) per species present in recordTable. No "<species>_RAI" columns (redundant at the daily level – see above). Includes only the days on which the station was active in camOp.

build_site_month	<i>Build a site x month dataset with independent-event counts per species</i>
------------------	---

Description

Build a site x month dataset with independent-event counts per species

Usage

```
build_site_month(
  recordTable,
  camOp,
  stationCol = "Station",
  speciesCol = "Species",
  dateTimeCol = "DateTimeOriginal",
  threshold_min = 30,
  min_days = NULL,
  independence_method = c("chain", "window"),
  require_uninterrupted = FALSE
)
```

Arguments

recordTable	camtrapR-style data.frame (e.g. output of recordTable()).
camOp	effort matrix produced by cameraOperation().
stationCol	name of the station column in recordTable (default "Station").
speciesCol	name of the species column in recordTable (default "Species").
dateTimeCol	name of the date/time column in recordTable (default "DateTimeOriginal").
threshold_min	independence threshold in minutes (default 30).
min_days	if specified (0 to 31), keeps only rows with n_days_active greater than or equal to min_days. Default NULL = no filter. NB: the possible maximum varies month by month (see the n_days_in_month column) – 31 is only the theoretical upper bound.
independence_method	how consecutive photos of the same species are collapsed into one independent event: "chain" (default) or "window". See build_site_block() docs for details.
require_uninterrupted	apply O'Brien et al. (2003)'s third independence criterion (default FALSE). See build_site_block() docs.

Value

data.frame with columns Station, year, month, month_start, month_end, mid_day (day halfway between month_start and month_end, handy for a join with external daily covariates), n_days_active, n_days_in_month, N_sp (number of species with at least one event in the month), and for each species two columns: "<species>_N" (independent-event count) and "<species>_RAI" (Relative Abundance Index = count / n_days_active * 100, rounded to 2 decimals).

Note

Species names in the columns are cleaned (spaces and non-alphanumeric characters replaced by "_") via the internal `.sanitize_species_names()`.

build_site_period	<i>Build a site x period/season dataset with independent-event counts per species</i>
-------------------	---

Description

Build a site x period/season dataset with independent-event counts per species

Usage

```
build_site_period(
  recordTable,
  camOp,
  period_names,
  period_starts,
  stationCol = "Station",
  speciesCol = "Species",
  dateTimeCol = "DateTimeOriginal",
  threshold_min = 30,
  min_days = NULL,
  independence_method = c("chain", "window"),
  require_uninterrupted = FALSE
)
```

Arguments

recordTable	camtrapR-style data.frame (e.g. output of <code>recordTable()</code>).
camOp	effort matrix produced by <code>cameraOperation()</code> .
period_names	vector of period labels, e.g. <code>c("winter", "spring", "summer", "autumn")</code> . Also defines the number of groups (here: 4).
period_starts	vector of "DD/MM/YYYY" dates, same length as <code>period_names</code> , same order – the start date of each period. The day-month pattern repeats every year; the year written is only a reference used to build the date.
stationCol	name of the station column in <code>recordTable</code> (default "Station").
speciesCol	name of the species column in <code>recordTable</code> (default "Species").
dateTimeCol	name of the date/time column in <code>recordTable</code> (default "DateTimeOriginal").
threshold_min	independence threshold in minutes (default 30).
min_days	if specified, keeps only rows with <code>n_days_active</code> greater than or equal to <code>min_days</code> . Default <code>NULL</code> = no filter. Only a lower-bound check (<code>min_days >= 0</code>) is enforced here: unlike <code>build_site_month()</code> 's fixed 31-day ceiling, a period's maximum length is entirely user-defined (see the <code>period_length_days</code> column), so no universal upper bound exists to validate against.

independence_method

how consecutive photos of the same species are collapsed into one independent event: "chain" (default) or "window". See build_site_block() docs for details.

require_uninterrupted

apply O'Brien et al. (2003)'s third independence criterion (default FALSE). See build_site_block() docs.

Value

data.frame with columns Station, period_name, year, period_start ("DD/MM", no year), period_end ("DD/MM", no year), mid_day (a REAL Date, with year – the day halfway between the start and end of the period, handy for a join with external daily covariates), period_length_days, n_days_active, N_sp (number of species with at least one event in the period), and for each species two columns: "<species>_N" (independent-event count) and "<species>_RAI" (Relative Abundance Index = count / n_days_active * 100, rounded to 2 decimals).

Note

Species names in the columns are cleaned (spaces and non-alphanumeric characters replaced by "_") via the internal .sanitize_species_names().

build_site_total	<i>Build a dataset with 1 row per site, aggregating over the entire activity period</i>
------------------	---

Description

Build a dataset with 1 row per site, aggregating over the entire activity period

Usage

```
build_site_total(
  recordTable,
  camOp,
  stationCol = "Station",
  speciesCol = "Species",
  dateTimeCol = "DateTimeOriginal",
  threshold_min = 30,
  independence_method = c("chain", "window"),
  require_uninterrupted = FALSE
)
```

Arguments

recordTable	camtrapR-style data.frame (e.g. output of recordTable()).
camOp	effort matrix produced by cameraOperation().
stationCol	name of the station column in recordTable (default "Station").

speciesCol name of the species column in recordTable (default "Species").
dateTimeCol name of the date/time column in recordTable (default "DateTimeOriginal").
threshold_min independence threshold in minutes (default 30).
independence_method
 how consecutive photos of the same species are collapsed into one independent event: "chain" (default) or "window". See build_site_block() docs for details.
require_uninterrupted
 apply O'Brien et al. (2003)'s third independence criterion (default FALSE). See build_site_block() docs.

Value

data.frame with columns Station, n_days_active (total active days over the entire period), N_sp (number of species with at least one event over the entire period), and for each species two columns: "<species>_N" (independent-event count over the entire period) and "<species>_RAI" (= $N / n_days_active * 100$, rounded to 2 decimals).

Note

Species names in the columns are cleaned (spaces and non-alphanumeric characters replaced by "_") via the internal .sanitize_species_names().

build_species_pair_interruptions

Compute the AA/ABA and BB/BAB intervals between two species (Parsons et al. 2016, T3/T4)

Description

Compute the AA/ABA and BB/BAB intervals between two species (Parsons et al. 2016, T3/T4)

Usage

```

build_species_pair_interruptions(
  recordTable,
  speciesA,
  speciesB,
  stationCol = "Station",
  speciesCol = "Species",
  dateTimeCol = "DateTimeOriginal",
  threshold_min = 30,
  independence_method = c("chain", "window"),
  require_uninterrupted = FALSE
)

```

Arguments

recordTable camtrapR-style data.frame (e.g. output of recordTable()).
speciesA, speciesB names of the two species to compare (as they appear in speciesCol).
stationCol name of the station column (default "Station").
speciesCol name of the species column (default "Species").
dateTimeCol name of the date/time column (default "DateTimeOriginal").
threshold_min independence threshold in minutes (default 30).
independence_method how consecutive photos of the same species are collapsed into one independent event: "chain" (default) or "window". See build_site_block() docs for details.
require_uninterrupted apply O'Brien et al. (2003)'s third independence criterion (default FALSE). See build_site_block() docs.

Value

long data.frame with columns Station, type ("AA", "ABA", "BB" or "BAB"), time_from (start of the interval) and delta_hours (duration). For the T4/T3 comparison of Parsons et al., extract the vectors with: AA <- out\$delta_hours[out\$type == "AA"] ABA <- out\$delta_hours[out\$type == "ABA"] (and likewise for BB/BAB).

build_species_pair_intervals

Compute the AB/BA intervals between two species (Niedballa et al. 2019 method), with censoring

Description

Compute the AB/BA intervals between two species (Niedballa et al. 2019 method), with censoring

Usage

```

build_species_pair_intervals(
  recordTable,
  camOp,
  speciesA,
  speciesB,
  stationCol = "Station",
  speciesCol = "Species",
  dateTimeCol = "DateTimeOriginal",
  threshold_min = 30,
  max_gap_hours = NULL,
  independence_method = c("chain", "window"),
  require_uninterrupted = FALSE
)
  
```

Arguments

recordTable	camtrapR-style data.frame (e.g. output of recordTable()).
camOp	effort matrix produced by cameraOperation() – used to determine the end of each station’s monitoring period, for censoring intervals with no subsequent event.
speciesA, speciesB	names of the two species to compare (as they appear in speciesCol).
stationCol	name of the station column (default "Station").
speciesCol	name of the species column (default "Species").
dateTimeCol	name of the date/time column (default "DateTimeOriginal").
threshold_min	independence threshold in minutes (default 30).
max_gap_hours	if specified, truncates (administrative censoring) intervals longer than this number of hours: delta_hours is capped at max_gap_hours and censored becomes TRUE. Does not discard rows. Default NULL = no truncation.
independence_method	how consecutive photos of the same species are collapsed into one independent event: "chain" (default) or "window". See build_site_block() docs for details.
require_uninterrupted	apply O’Brien et al. (2003)’s third independence criterion (default FALSE). See build_site_block() docs.

Value

long data.frame with columns Station, direction ("AB" or "BA"), time_from (timestamp of the starting event), delta_hours (hours until the next event of the other species, or until the censoring limit) and censored (TRUE if there was no subsequent event before the limit – end of monitoring or max_gap_hours). For the Niedballa comparison, extract the two vectors with: AB <- out\$delta_hours[out\$direction == "AB"] BA <- out\$delta_hours[out\$direction == "BA"] (and decide yourself whether to include or exclude rows with censored == TRUE).

camOp_soReta

Example camera effort matrix for soReta

Description

A station x day effort matrix, built once from [camtraps_soReta](#) via `camtrapR::cameraOperation()`, and bundled directly with the package: using `soReta`’s example data no longer requires `camtrapR` at all, since this object is already the finished result of that step. Any station x day matrix in this shape works equally well with every function in this package; `cameraOperation()` is simply the standard, convenient way to build one from deployment/retrieval/malfunction dates – you are not required to use `camtrapR` to build your own.

Usage

```
camOp_soReta
```

Format

A matrix with 5 rows (stations, matching [camtraps_soReta](#)) and 120 columns (one per calendar day from 01/01/2026 to 30/04/2026, column names "YYYY-MM-DD"). Values are 1 (camera active that day), 0 (camera not yet deployed, already retrieved, or down due to a malfunction period), or NA (before deployment/after retrieval, camtrapR's convention for "no camera present at all" as opposed to "present but not working").

Source

Synthetic data generated for this package; see `data-raw/create_sample_data.R` for the full generation script (fixed random seed, fully reproducible; requires camtrapR to regenerate, though not to use the resulting object).

See Also

[camtraps_soReta](#), the station table this matrix was built from.

camtraps_soReta	<i>Example station table for soReta</i>
-----------------	---

Description

A small, entirely SYNTHETIC camera-trap station table, in the format expected by `camtrapR::cameraOperation()`. Built specifically as the example dataset for this package, so it never needs to depend on camtrapR's own sample data (which lacks an individual-count column and uses different species codes across examples).

Usage

`camtraps_soReta`

Format

A data frame with 5 rows and 7 columns:

- Station** station ID, "S_01" to "S_05".
- Setup_date** deployment date, "DD/MM/YYYY".
- Retrieval_date** retrieval date, "DD/MM/YYYY". All stations share the same retrieval date, 30/04/2026.
- Problem1_from,Problem1_to** start/end of the first malfunction period for that station, "DD/MM/YYYY", or "" if none.
- Problem2_from,Problem2_to** start/end of a second malfunction period, "DD/MM/YYYY", or "" if none. Only stations S_03 and S_05 have a second problem period in this example.

Source

Synthetic data generated for this package; see `data-raw/create_sample_data.R` for the full generation script.

See Also

[recordTable_soReta](#), the matching example recordTable.

extract_radians	<i>Extract, per species (and optionally per one or more groupings), a vector of times in radians</i>
-----------------	--

Description

Extract, per species (and optionally per one or more groupings), a vector of times in radians

Usage

```
extract_radians(
  recordTable,
  dateTimeCol = "DateTimeOriginal",
  stationCol = "Station",
  speciesCol = "Species",
  threshold_min = 30,
  group_cols = c(stationCol, speciesCol),
  group_col = NULL,
  independence_method = c("chain", "window"),
  require_uninterrupted = FALSE
)
```

Arguments

recordTable	camtrapR-style data.frame (e.g. output of recordTable()).
dateTimeCol	name of the date/time column (default "DateTimeOriginal").
stationCol	name of the station column (default "Station").
speciesCol	name of the species column (default "Species").
threshold_min	independence threshold in minutes (default 30).
group_cols	vector of column names used to group events when computing independence (default c(stationCol, speciesCol)).
group_col	vector of one or more column names used to split each species' result into successive nested levels (e.g. "bimonth", or c("Station", "bimonth")). Default NULL = no splitting, a single vector per species.
independence_method	how consecutive photos of the same species are collapsed into one independent event, within each group_cols subsequence: "chain" (default) or "window". See build_site_block() docs for details.
require_uninterrupted	apply O'Brien et al. (2003)'s third independence criterion (default FALSE): two photos are only subject to the time threshold if no photo of ANY other species falls chronologically between them at that station (stationCol), regardless of group_cols. See build_site_block() docs.

Details

The radians are in CLOCK TIME, not solar time. For comparisons across seasons or different latitudes, convert the result with `activity::solartime()` or `overlap::sunTime()` before passing it to `densityPlot()/fitact()` – this function does not do that.

Value

Named list per species. If `group_col` is NULL: `result[[species]]` is a numeric vector of radians. If `group_col` has 1 element: `result[[species]][[group]]`. If it has 2 or more: one further nested level per column, in the same order as `group_col`.

<code>radians_to_env</code>	<i>Create a separate object in the environment for each vector in a nested list of radians</i>
-----------------------------	--

Description

Create a separate object in the environment for each vector in a nested list of radians

Usage

```
radians_to_env(
  radians_list,
  prefix = "Rad_",
  sep = "_",
  envir = parent.frame()
)
```

Arguments

<code>radians_list</code>	(possibly nested) list produced by <code>extract_radians()</code> .
<code>prefix</code>	prefix for the created object names (default "Rad_").
<code>sep</code>	separator between the pieces of the name (default "_").
<code>envir</code>	environment in which to create the objects (default the environment from which you call the function).

Value

(invisible) the vector of names of the created objects. The function is called for its side effect of creating the objects in `envir`, not for its return value.

recordTableIndividuals_soReta

Example individually-identifiable recordTable for soReta

Description

A small, entirely SYNTHETIC individual-level recordTable, in the format expected by `build_cmr_day()/build_cmr_block` (the same shape as `camtrapR`'s own `recordTableIndividual()` output). Built by taking the "red deer" detections already present in [recordTable_soReta](#) and assigning each one to one of 8 individually-recognizable deer, with unequal, realistic recapture rates (a few "resident" individuals seen often, others seen only once or twice) – purely for illustration, not derived from any real survey. Unlike the CMR examples elsewhere in `camtrapR`-based tutorials, this dataset requires no dependency on `camtrapR`'s own bundled data.

Usage

```
recordTableIndividuals_soReta
```

Format

A data frame with 103 rows and 4 columns:

Station station ID, matching [camtraps_soReta](#).

Species always "red deer" in this dataset.

Individual individual ID, "RD_01" to "RD_08".

DateTimeOriginal date and time of the photo, POSIXct (UTC).

Source

Synthetic data generated for this package; see `data-raw/create_individuals_dataset.R` for the full generation script (fixed random seed, fully reproducible; run after `data-raw/create_sample_data.R`, since it starts from that script's `recordTable_soReta` object).

See Also

[recordTable_soReta](#), the recordTable this dataset's red deer detections were taken from; [camtraps_soReta](#), the matching example station table.

recordTable_soReta	<i>Example recordTable for soReta</i>
--------------------	---------------------------------------

Description

A small, entirely SYNTHETIC camtrapR-style recordTable: four species (wolf, red fox, wild boar, red deer), detections spread across the five stations in [camtraps_soReta](#), only on days those stations were actually active. Detection times are weighted towards night and twilight hours. Individual counts (N_individuals) follow different distributions per species – red fox and red deer mostly solitary (max 2 and 4 respectively), wolf up to small-pack sizes (max 8), wild boar in larger sounders on average (max 30) – purely for illustration, not derived from any real survey.

Usage

```
recordTable_soReta
```

Format

A data frame with 404 rows and 4 columns:

Station station ID, matching [camtraps_soReta](#).

DateTimeOriginal date and time of the photo, POSIXct (UTC).

Species one of "wolf", "red fox", "wild boar", "red deer".

N_individuals number of animals visible in that photo.

Details

Detections are clustered into "visits" of 1 to 4 photos rather than spread one-per-day, so that most active days have no detection at all, while the days that do often have several photos close together in time – including two deliberately placed sequences (one for "wolf" at station S_02, one for "wild boar" at station S_03) with gaps of exactly 16 and 25 minutes between consecutive photos: with the default 30-minute threshold, `independence_method = "chain"` collapses each of these into a single event, while `"window"` splits it into two – worked examples for the vignette's discussion of `independence_method/require_uninterrupted`.

Source

Synthetic data generated for this package; see `data-raw/create_sample_data.R` for the full generation script (fixed random seed, fully reproducible).

See Also

[camtraps_soReta](#), the matching example station table.

Index

* datasets

- camOp_soReta, [27](#)
- camtraps_soReta, [28](#)
- recordTable_soReta, [32](#)
- recordTableIndividuals_soReta, [31](#)

as_capture_strings, [2](#)

build_cmr_block, [3](#)
build_cmr_day, [4](#)
build_day_total, [5](#)
build_diel_binomial_block, [6](#)
build_diel_binomial_month, [7](#)
build_diel_binomial_period, [7](#)
build_diel_day, [8](#)
build_group_day_total, [9](#)
build_group_size_block, [10](#)
build_group_size_day, [12](#)
build_group_size_events, [13](#)
build_group_size_month, [14](#)
build_group_size_period, [15](#)
build_group_size_total, [16](#)
build_occupancy_block, [18](#)
build_occupancy_day, [19](#)
build_site_block, [19](#)
build_site_day, [21](#)
build_site_month, [22](#)
build_site_period, [23](#)
build_site_total, [24](#)
build_species_pair_interruptions, [25](#)
build_species_pair_intervals, [26](#)

camOp_soReta, [27](#)
camtraps_soReta, [27](#), [28](#), [28](#), [31](#), [32](#)

extract_radians, [29](#)

radians_to_env, [30](#)
recordTable_soReta, [29](#), [31](#), [32](#)
recordTableIndividuals_soReta, [31](#)