Package 'cumba'

November 19, 2026

Type Package

Title Carbon Use Model for Yield and Brix Assessment in Processing Tomato

Version 0.1.0

Description A daily time-step crop model for processing tomato that simulates thermal time, phenology, root growth, a three-layer soil-water balance, carbon accumulation modulated by heat, cold and water stress, and predicts fresh-fruit yield and Brix at harvest. Two top-level entry points are provided: cumba_experiment() replays observed irrigation schedules (used for calibration), and cumba_scenario() runs the model in deficit-irrigation scenario mode, with irrigation triggered automatically when the crop water status falls below phase-specific thresholds. Methods build on Yan and Hunt (1999) <doi:10.1006/anbo.1999.0955>, Zhao et al. (2019) <doi:10.1016/j.eja.2019.01.009>, Casadebaig et al. (2011) <doi:10.1016/j.agrformet.2010.09.012> and Prudent et al. (2010) <doi:10.1093/jxb/erq318>; the field dataset used for calibration is described in Carucci et al. (2024) <doi:10.1016/j.dib.2024.110225>.

License GPL-3

URL https://github.com/GeoModelLab/cumba, https://GeoModelLab.github.io/cumba/

BugReports https://github.com/GeoModelLab/cumba/issues

RoxygenNote 7.3.3

Encoding UTF-8

Imports crayon, dplyr, lubridate, tibble

Depends R (>= 4.1)

LazyData true

Suggests knitr, rmarkdown, ggplot2, testthat (>= 3.0.0)

VignetteBuilder knitr

Config/testthat/edition 3

NeedsCompilation no

Author Simone Bregaglio [aut, cre] (ORCID:
<https://orcid.org/0000-0001-8381-2626>),
Federica Carucci [aut],
Luca Bechini [aut] (ORCID: <https://orcid.org/0000-0002-8152-6121>),
Marcella Michela Giuliani [aut] (ORCID:
<https://orcid.org/0000-0001-9014-1955>),
Giuseppe Gatta [aut] (ORCID: <https://orcid.org/0000-0001-8381-7785>)

Maintainer Simone Bregaglio <simoneugomaria.bregaglio@crea.gov.it>

Repository CRAN

Date/Publication 2026-11-19 11:10:12 UTC

Contents

cumbaParameters 2
cumba_experiment 3
cumba_scenario 5
tomatoFoggia 8

Index **9**

`cumbaParameters` *Model Parameters for Cumba*

Description

This dataset contains the model parameters used in the Cumba package: RUE, Theat, k0 and FruitWaterContentDecreaseMax hold the values calibrated on the bundled tomatoFoggia experiments, the others literature or measured defaults. Each element corresponds to a model parameter and includes its description, current value, minimum, and maximum.

Usage

```
cumbaParameters
```

Format

A named list where each entry is a list with:

description Text description of the parameter

value The parameter value (calibrated or default)

min The minimum allowed value

max The maximum allowed value

Examples

```
names(cumbaParameters)
cumbaParameters$Tbase$value
```

`cumba_experiment` *CUMBA model - experiment runner*

Description

Runs the daily time-step CUMBA simulation for one or more sites/experiments, computing tomato yield and Brix from weather inputs and irrigation options. The function validates inputs, converts a list-of-lists parameter object to a tibble of numeric values when needed, and prints a blue message indicating it is running in *experiment* mode.

Usage

```
cumba_experiment(
  weather,
  param,
  estimateRad = TRUE,
  estimateET0 = TRUE,
  irrigation_df,
  irrigationEfficiency = 0.9,
  fullOut = FALSE,
  verbose = FALSE
)
```

Arguments

`weather`	A `data.frame` (or tibble) of daily weather by site with at least `Site`, `Tx`, `Tn`, `P`, `DATE` and, based on flags, either `Lat` (to estimate radiation) or `Rad` (provided radiation). If `estimateET0 = FALSE`, an `ET0` column must be present.
`param`	Model parameters. Either: • a *named list* where each element is itself a list with fields `value`, `min`, `max`, `description`; this form is automatically converted to a tibble of parameter *values*; or • a tibble/data.frame already containing the required parameter values. The object must contain (as names/columns) all of: `Tbase`, `Topt`, `Tmax`, `Theat`, `Tcold`, `FIntMax`, `CycleLength`, `TransplantingLag`, `FloweringLag`, `HalfIntGrowth`, `HalfIntSenescence`, `InitialInt`, `RUE`, `KcIni`, `KcMax`, `RootIncrease`, `RootDepthMax`, `RootDepthInitial`, `FieldCapacity`, `WiltingPoint`, `DepletionFraction`, `FloweringSlope`, `FloweringMax`, `k0`, `FruitWaterContentMin`, `FruitWaterContentMax`, `FruitWaterContentInc`, `FruitWaterContentDecreaseMax`. Cardinal temperatures are validated so that `Tmax` > `Topt` > `Tbase`.

`estimateRad` Logical. If `TRUE` (default), solar radiation is estimated from temperature (Hargreaves) and `Lat` must be in `weather`. If `FALSE`, `weather` must include `Rad`.

`estimateET0` Logical. If `TRUE` (default), reference evapotranspiration (ET0) is estimated from temperature (Hargreaves). If `FALSE`, `weather` must include an `ET0` column.

`irrigation_df` A `data.frame` with the irrigation schedule for each experiment/site, containing columns `ID`, `Site`, `YEAR`, `DATE`, `WVOL`.

`irrigationEfficiency` Numeric (0-1). Fraction of the observed applied water (`WVOL`) that effectively reaches the root zone. Default `0.9` (90%, typical for drip irrigation). The `irrigation` column in the output retains the original gross `WVOL` values; only the soil water balance uses the net fraction (`WVOL * irrigationEfficiency`). Set to `1` to disable the correction.

`fullOut` Logical. If `FALSE` (default) returns key outputs; if `TRUE` returns all internal variables.

`verbose` Logical. If `TRUE`, progress messages are printed to the console; default `FALSE`.

Details

On start, the function prints `"CUMBA running in experiment mode"` in blue (via **crayon**). It checks that all required columns/elements are present and throws informative errors if anything is missing. Sites are inferred from the unique values of `weather$Site`.

If `param` is supplied as a list-of-lists with a `value` field, it is converted internally with `tibble::as_tibble(lapply(param, `[[`, "value"))`.

Value

A `data.frame` containing the input `weather` plus the daily CUMBA outputs per site and date; the set of columns depends on `fullOut`.

Required inputs

weather: must include columns `Site`, `Tx`, `Tn`, `P`, `DATE`. Depending on the flags:

- if `estimateRad` = `TRUE` (default), column `Lat` is required;
- if `estimateRad` = `FALSE`, column `Rad` is required;
- if `estimateET0` = `FALSE`, column `ET0` is required.

`irrigation_df`: must include columns `ID`, `Site`, `YEAR`, `DATE`, `WVOL`.

See Also

`as_tibble`, `blue`, `red`

Examples

```
data(tomatoFoggia)
data(cumbaParameters)

# Replay one growing season from the bundled field dataset
wx <- subset(tomatoFoggia$weather, format(DATE, "%Y") == "2018")
out <- cumba_experiment(wx, cumbaParameters,
                        irrigation_df = tomatoFoggia$irrigation)
head(out)
```

`cumba_scenario` *CUMBA model - automatic irrigation scenario runner*

Description

Runs the daily time-step CUMBA simulation in *scenario* (automatic irrigation) mode. Unlike `cumba`, which requires an observed irrigation schedule, `cumba_scenario` computes irrigation autonomously based on the crop water stress index, phenological stage, and user-defined stress thresholds. This is the function used by the CUMBA Shiny app for decision support and what-if analysis.

Usage

```
cumba_scenario(
  weather,
  param,
  estimateRad = TRUE,
  estimateET0 = TRUE,
  transplantingDOY = 120,
 irrigationStrategy = list(vegetative = list(wsLevel = 0.5, turnMin = 2), reproductive =
   list(wsLevel = 0.5, turnMin = 2), ripening = list(wsLevel = 0.5, turnMin = 2)),
  irrigationStopCycle = 100,
  irrigationEfficiency = 0.9,
  fullOut = FALSE,
  irrigationOverride = NULL,
  verbose = FALSE
)
```

Arguments

`weather`	A `data.frame` (or tibble) of daily weather by site. See *Required weather columns* above.
`param`	Model parameters. Either a *named list* where each element is itself a list with fields `value`, `min`, `max`, `description` (as returned by `cumbaParameters`), or a tibble/data.frame of numeric parameter values already unpacked. See `cumba` for the full list of required parameter names.

estimateRad Logical. If TRUE (default), solar radiation is estimated from temperature (Hargreaves); Lat must be in weather. If FALSE, weather must include Rad.

estimateET0 Logical. If TRUE (default), ET0 is estimated from temperature (Hargreaves). If FALSE, weather must include ET0.

transplantingDOY
Integer. Day of year (1-365) of transplanting. The model starts accumulating GDD and running the water balance from this day. Default 120 (approx. 30 April).

irrigationStrategy
A named list with three phenological phases - vegetative, reproductive, ripening - each containing:

wsLevel Numeric (0-1). Per-phase threshold applied to ws = 1 - WSI, the complement of the water-stress index WSI (derived from the fraction of transpirable soil water; 0 = no stress, 1 = maximum stress). Irrigation is triggered when ws < wsLevel, i.e. when WSI exceeds 1 - wsLevel; higher values trigger at lower stress and keep the soil wetter (1 approaches full irrigation).

turnMin Integer. Minimum number of days that must have elapsed since the last irrigation event before a new one is allowed.

Default: all phases use wsLevel = 0.5 and turnMin = 2.

irrigationStopCycle
Numeric (0-100). Cycle-completion percentage beyond which automatic irrigation is suppressed regardless of crop water status. Useful to implement a cut-off irrigation strategy near harvest. Default 100 (no cut-off; irrigation runs until the end of the cycle).

irrigationEfficiency
Numeric (0-1). Fraction of applied water that effectively reaches the root zone. Default 0.9 (90%, typical for drip irrigation). The model computes the net soil-water deficit and divides it by this value to obtain the *gross* volume the farmer must apply; the irrigation column in the output reports this gross amount, while the soil water balance uses only the net fraction (gross * efficiency). Set to 1 to disable the efficiency correction (e.g. for sprinkler or furrow irrigation where efficiency is accounted for elsewhere).

fullOut Logical. If FALSE (default), the function returns the key agronomic outputs (stage, swc, yield, brix, irrigation, p) plus site/date identifiers. If TRUE, all internal state variables are included.

irrigationOverride
A data.frame (or NULL) that lets the user manually override the model's automatic irrigation decision for specific dates. Must contain columns date (class Date) and action ("skip" or "applied"). An optional mm column specifies the volume when action = "applied".

"skip" Forces irrigation = 0 on that date, even if the model would have irrigated.

"applied" Forces irrigation = mm on that date, even if the model would not have irrigated.

Default NULL (no overrides).

verbose Logical. If TRUE, progress messages are printed to the console; default FALSE.

Details

On each simulated day the model: (1) updates the two-layer soil-water balance; (2) computes the water-stress index `WSI` from the fraction of transpirable soil water (FTSW), and its complement `ws = 1 - WSI`; (3) decides whether to irrigate - irrigation is applied when `ws < wsLevel` for the current phase, at least `turnMin` days have elapsed since the last event, `cycleCompletion < irrigationStopCycle`, and the crop is in an active phenological phase (`phenoCode > 0`); (4) applies `irrigationEfficiency` so that the gross applied volume equals `net deficit / efficiency` while only the net fraction enters the soil; (5) computes biomass accumulation, fruit growth, and Brix dynamics.

Any `irrigationOverride` entries are applied after the automatic decision and before the water-balance update, so they correctly propagate to soil moisture in subsequent days.

The function prints `"CUMBA running in deficit irrigation mode"` in blue (via **crayon**) when called.

Value

A `data.frame` of daily outputs. When `fullOut = FALSE` (default) the columns are: `site`, `year`, `doy`, `p` (precipitation, mm), `irrigation` (mm), `stage` (phenological stage label), `swc` (soil water content, %), `yield` (fruit fresh weight, t/ha), `brix` (degrees Brix). When `fullOut = TRUE`, all internal state variables are included.

Required weather columns

`weather` must contain `Site`, `Tx`, `Tn`, `P`, `DATE`. Additionally:

- if `estimateRad = TRUE` (default), `Lat` is required;
- if `estimateRad = FALSE`, `Rad` must be present;
- if `estimateET0 = FALSE`, `ET0` must be present.

See Also

`cumba` for the experiment-mode runner that requires an observed irrigation schedule; `cumbaParameters` for the default parameter set.

Examples

```
data(tomatoFoggia)
data(cumbaParameters)

# Autonomous deficit-irrigation scenario on one growing season
wx <- subset(tomatoFoggia$weather, format(DATE, "%Y") == "2018")
out <- cumba_scenario(
  weather          = wx,
  param            = cumbaParameters,
  transplantingDOY = 120,
  irrigationStrategy = list(
    vegetative   = list(wsLevel = 0.5, turnMin = 4),
    reproductive = list(wsLevel = 0.5, turnMin = 4),
    ripening     = list(wsLevel = 0.5, turnMin = 4)
```

```
  )
)
head(out)
```

`tomatoFoggia` *Tomato field trial data from Foggia (Italy)*

Description

A dataset containing weather, irrigation, management, and production data from processing tomato field trials conducted near Foggia (Apulia, Italy). The dataset is organized as a list of four data frames.

Usage

```
tomatoFoggia
```

Format

A list of four data frames:

`weather` Daily weather data with columns: DATE (POSIXct), Tx (max temp, degC), Tn (min temp, degC), P (precipitation, mm), Rad (radiation, MJ m-2 d-1), RHx (max RH, Date (Date), Site ("Foggia"), Lat (latitude, numeric).

`irrigation` Irrigation records with columns: ID (plot id), DATE (POSIXct), WVOL (mm), YEAR, CV (cultivar), TRANS (transplanting, POSIXct), HARV (harvest, POSIXct), IRR_M (management, e.g. "50 Site ("Foggia").

`management` Plot identifiers and treatments with columns: ID, YEAR, CV, TRANS (POSIXct), HARV (POSIXct), IRR_M.

`production` Production and quality data with columns: ID, yield_ref (t ha-1), yield_ref_sd (t ha-1), brix_ref (degBrix), brix_ref_sd (degBrix).

Source

Carucci et al. (2024) doi:10.1016/j.dib.2024.110225

Examples

```
data(tomatoFoggia)
str(tomatoFoggia)
head(tomatoFoggia$weather)
```

Index

∗ **datasets**
cumbaParameters, 2
tomatoFoggia, 8

as_tibble, *4*

blue, *4*

cumba, *5*, *7*
cumba_experiment, 3
cumba_scenario, 5
cumbaParameters, 2, *5*, *7*

red, *4*

tomatoFoggia, 8