Received: 14 April 2023 | Accepted: 6 July 2023

DOI: 10.1111/2041-210X.14192

Methods in Ecology and Evolution

APPLICATION

myClim: Microclimate data handling and standardised analyses in R

Matěj Man[1,2] | Vojtěch Kalčík[1] | Martin Macek[1] | Josef Brůna[1] | Lucia Hederová[1] | Jan Wild[1] | Martin Kopecký[1]

[1]Institute of Botany of the Czech Academy of Sciences, Průhonice, Czech Republic

[2]Department of Botany, Faculty of Science, Charles University, Prague 2, Czech Republic

Correspondence
Matěj Man
Email: matej.man@ibot.cas.cz

Funding information
Akademie Věd České Republiky, Grant/Award Number: RVO 67985939; Grantová Agentura České Republiky, Grant/Award Number: GACR 20-28119S and GACR 23-06614S; Technology Agency of the Czech Republic, Grant/Award Number: TACR SS02030018

Handling Editor: Sarah Goslee

Abstract

1. Microclimates have been recognised as one of the key drivers in global change biology. Durable microclimate loggers, detailed in-situ measurements and sophisticated modelling tools are increasingly available, but a lack of standardised workflows for microclimate data handling hinders synthesis across the studies and thus progress in the global change biology. To overcome these limitations, we developed an R package *myClim* for microclimate data processing, storage and analyses. The *myClim* package supports complete workflow for microclimate data handling, including reading raw logger data files, their preprocessing and cleaning, time-series' aggregation, calculation of ecologically relevant microclimatic variables, data export and storage.
2. The *myClim* package stores data in a size-efficient, hierarchical structure which respects the hierarchy of field microclimate measurement (locality > loggers > sensors). For imported microclimatic data, *myClim* provides an informative summary and automatically detects and corrects common issues like duplicated and wrongly ordered measurements. The *myClim* package also provides advanced functions for microclimate data aggregation to various timescales (e.g. days, months, years or growing seasons) as well as tools for sensor calibration, data conversion and joining of multiple microclimatic time series.
3. The *myClim* package provides advanced functions for standardised calculation of ecologically relevant microclimatic variables like freezing and growing degree days, snow cover period, soil volumetric water content and atmospheric vapour pressure deficit. Calculated microclimatic variables are stored efficiently in *myClim* data format and can be easily exported to long or wide tables for further analyses and visualisations.
4. Adopting *myClim* can facilitate large-scale syntheses, boost data sharing and increase the comparability and reproducibility of microclimatic studies. The stable version of *myClim* is available on CRAN (https://cran.r-project.org/web/packages/myClim) and the development version is available on GitHub (https://github.com/ibot-geoecology/myClim).

KEYWORDS
air temperature, microclimate, relative humidity, sensor calibration, soil moisture, soil temperature, TMS microclimate logger, vapour pressure deficit

1 | INTRODUCTION

Interacting effects of climate, topography and vegetation create a fine-scaled and temporary dynamic mosaic of microclimates, substantially different from free-air conditions recorded by standard weather stations and predicted by global climatic models (Geiger et al., 2009; Slavich et al., 2014). Local microclimate, and not regional macroclimate, directly affect organisms and key ecological processes (Körner, 2021; Nadeau et al., 2017). While the microclimate is always local, it also affects biodiversity and ecosystem processes on larger scales (Nadeau et al., 2022; Zellweger et al., 2020). Microclimate thus became a central theme in global change biology and ecology (De Frenne et al., 2021; Potter et al., 2013).

To measure a microclimate in the field, researchers used different nonspecialised industrial loggers (Hubbart et al., 2005; Lundquist & Lott, 2008; Whiteman et al., 2000) and, increasingly, also new microclimatic loggers specially designed for ecological applications (Wild et al., 2019). The wide variability of employed microclimatic loggers and their different field installation sparked studies exploring effects introduced by different logger types, radiation shields and various other adjustments like waterproofing (Holden et al., 2013; Maclean et al., 2021; Navarro-Serrano et al., 2019; Roznik & Alford, 2012; Terando et al., 2017). The rapidly increasing number of microclimatic studies (e.g. Finocchiaro et al., 2023; Greiser et al., 2020; Macek et al., 2019) and the establishment of local microclimate monitoring networks (e.g. Aalto et al., 2022; Dickerson-Lange et al., 2015; Lundquist et al., 2016) facilitate the creation of global microclimate database SoilTemp, aggregating data from thousands of localities (Lembrechts et al., 2020).

However, recent development in microclimate monitoring was not accompanied by the development of standardised methods and procedures for microclimate data handling and processing (Bramer et al., 2018). Microclimatic studies thus often use different workflows, data treatments and storage formats even for the variables measured with the same sensors. Similarly, the algorithms used to calculate microclimatic variables from field measurements often differ between studies. The lack of common processing tools, standard algorithms and data format hampers comparability across the studies and data integration over the larger scales needed in global change biology.

To overcome these limitations, we developed the *myClim* R package for microclimate data processing, storage and analysis. Here, we describe *myClim* structure, logic and functionality (Table 1) and provide code examples in Appendix S2. The *myClim* package implements the complete microclimatic workflow from the import of the raw microclimatic time series to the calculation of ecologically relevant variables in a fully reproducible and standardised way using open-source code (Figure 1). Therefore, the *myClim* R package will be useful to a wide audience and facilitate further advances in microclimate science.

1.1 | *myClim* workflow

1.2 | *myClim* data structure

Microclimatic data imported into *myClim* are stored in custom R classes and predefined lists with a hierarchical structure. This allows a combination of many loggers and localities, speeds up data manipulation and calculations, and reduces memory demand for data storage. For example, after import to *myClim*, microclimatic data originating from 2000 TOMST TMS loggers (four sensors per logger, recording every 15 min, distributed in 2000 CSV files) with a total size of 15 GB on the drive occupy only 5 GB in RAM and can be saved as the *myClim* R object to an RDS file of only 0.9 GB. This is a substantial reduction compared to the 25 GB RAM needed to load the same data to R as data tables and to the 1.2 GB needed to save those data tables as an RDS file.

The *myClim* objects have three hierarchical levels: locality, logger and sensor (Figure 2). Each hierarchical level can hold metadata (Figure 2). Besides metadata, loggers can be associated with the output of the *mc_clean* function. Sensors can hold *calibration*, that is, the correction factor and slope from the *mc_prep_calib_load* function and *states*, for example, for the path to the original files or data quality flags. Each *myClim* locality can contain an unlimited number of loggers, and each logger can have multiple sensors measuring different physical variables at different heights.

The *myClim* objects exist either in *Raw-* or *Agg-format*, see Appendix S1. The main difference between the formats is at the logger level. With original data in *Raw-format*, the level of logger is present and can be used for joining multiple downloads from the same logger. With analysis-ready data in *Agg-format*, the level of logger is missing, and time series are associated directly with localities. *Agg-format* thus allows for easily linking microclimatic time series with other locality-specific data, like species' occurrence data, topography, soil, macroclimate or habitat type. The *myClim* functions work with both *Raw-* and *Agg-formats*.

Time series with different timesteps (e.g. there are two loggers simultaneously recording on the same locality, but they are not synchronised; the first one is recording every 15 min, the second one is recording every hour) are allowed only in the *Raw-format*, but not in *Agg-format*. Therefore, the only way how to get heterogeneous time series to *Agg-format* is their aggregation to the same timestep.

1.3 | Reading the microclimatic data

The *myClim* functions read directly the native files downloaded from various microclimatic loggers (e.g. TOMST TMS loggers and Onset

TABLE 1 Overview of the *myClim* functions with a short description. The default settings are listed for the calculation functions.

Reading	
mc_read_data()	Read datalogger files with metadata; *mc_prep_clean* is applied by default
mc_read_files()	Read datalogger files or directories without metadata; *mc_prep_clean* is applied by default
mc_read_long()	Read time series from R data frame in long format to *myClim*
mc_read_wide()	Read time series from R data frame in wide format to *myClim*
Exploring	
mc_info()	Show overview table per sensor (e.g. start and end date, measurement time step, min and max value, number of measurements, number of NAs)
mc_info_clean()	Show overview table from time series cleaning (e.g. number of duplicated measurements, number of missing measurements, number of wrongly ordered measurements)
mc_info_count()	Show overview table of the *myClim* object (e.g. number of localities, loggers and sensors)
mc_info_meta()	Show overview table of locality metadata (e.g. locality id, longitude, latitude, elevation, offset to UTC time)
Reshaping	
mc_reshape_long()	Reshape and export the *myClim* object time series to R data frame in long format
mc_reshape_wide()	Reshape and export the *myClim* object time series to R data frame in wide format
Pre-processing	
mc_prep_calib()	Adjust measured values using sensor-specific calibration coefficients provided in sensor metadata
mc_prep_calib_load()	Load sensor-specific calibration coefficients to the sensor metadata of the *myClim* object
mc_prep_clean()	Clean microclimatic time series (automatic removal of duplicates, reordering of misaligned measurements and identification of missing measurements)
mc_prep_crop()	Crop the *myClim* object to defined period (from start date and time to end date and time)
mc_prep_merge()	Merge multiple *myClim* objects into a single object
mc_prep_meta_locality()	Update metadata or rename localities in the *myClim* object
mc_prep_meta_sensor()	Update height or rename sensors in the *myClim* object
mc_prep_solar_tz()	Calculate the local solar time based on the longitude of the locality (the offset in minutes against UTC time)
mc_prep_fillNA()	Fill missing values in time series using simple linear interpolation between last and first non-missing values, default maximum number of consecutive NAs to be interpolated: maxgap = 5
Tools	
mc_agg()	Aggregate microclimatic time series using one or more specified functions over user-defined period
mc_filter()	Filter data for specific localities and/or sensors
mc_join()	Join microclimatic time series from the same locality into single time series
mc_save()	Save the *myClim* object as RDS file compatible with the future *myClim* versions
mc_load()	Load RDS file saved with *mc_save*
Plotting	
mc_plot_image()	Plot and export time series as a heatmap using base R graphics
mc_plot_loggers()	Plot and export time series as a line plot into a separate file for each logger (optimised for TMS loggers)
mc_plot_line()	Plot and export line plots faceted by locality, allows plotting of values with two different physical units in one plot, for example, temperature and moisture
mc_plot_raster()	Plot time series as a raster heatmap, export paginated pdf or png
Calculation	
mc_calc_cumsum()	Calculate cumulative sum of the values in the time series
mc_calc_fdd()	Calculate freezing degree days, default base temperature = 0 (°C)
mc_calc_gdd()	Calculate growing degree days, default base temperature = 5 (°C)

(Continues)

TABLE 1 (Continued)

mc_calc_snow()	Estimate snow cover (TRUE/FALSE) from temperature time series, defaults: dr = 2 (°C), tmax = 0.5 (°C)
mc_calc_snow_agg()	Summary of snow (TRUE/FALSE) sensor: the sum of days with snow, first and last day of snow presence and continual snow cover longer than user defined period.
mc_calc_vwc()	Convert raw moisture signal from TMS logger to volumetric water content, default soil type: soiltype = "universal"
myClim variables	
mc_env_temp()	Calculate the standard *myClim* temperature variables (mean, *min5p*, *max95p*, daily range, GDD, FDD, number of frost days)
mc_env_moist()	Calculate the standard *myClim* soil moisture variables (mean, *min5p*, *max95p*, SD)
mc_env_vpd()	Calculate the standard *myClim* vapour pressure deficit variables (mean, *max95p*)

HOBO loggers), but it is also possible to import other climatic time series. Moreover, the *myClim* routine for data import can be customised to support other logger types like iButton, Lascar or Logtag. During the data import, *myClim* runs by default automatic time series cleaning and correction routine (see below *mc_prep_clean*), but the cleaning can be turned off and called up separately.

The myClim package can read either individual data files or all data files from a specified directory (and all its subdirectories) with the *mc_read_files* function. This function reads time series directly from files without any metadata. The time series are therefore organised in localities named according to the serial number of the logger (when available in the file header or provided as a part of the file name) or by the corresponding data file name.

To import both microclimatic time series and associated metadata, we developed the *mc_read_data* function, which reads two tables joined by locality id: (1) a table with a path to the data files, locality id and type of microclimatic logger, and (2) a table with metadata for each locality (e.g. geographic coordinates, elevation, time offset to UTC, Figure 1). The locality metadata of the *myClim* object can also be added later or updated with the *mc_prep_meta_locality* function, which can accept either a named list for updating a single metadata slot or a data frame with defined columns for multiple updates.

1.4 | Preprocessing

1.4.1 | Time

The myClim package works with UTC as well as non-UTC time data, but the *myClim* functions assume that the data are in UTC. Therefore, to work properly in non-UTC time, the temporal offset (in minutes) between the local or solar time and the UTC must be specified in the metadata of each locality. The raw time series in the *myClim* objects stay in UTC, and the offsets are applied during data aggregation. After the aggregation with local or solar time offsets, the aggregated time series is no longer in UTC but inherits the solar or local time.

Using the local or solar time could be important in ecological analysis on large spatial scales because of the photoperiod shift around the globe. Therefore, we developed a *mc_prep_solar_tz* function, which, for each microclimatic measurement, calculates the time offset to UTC from geographic coordinates of each locality provided in the metadata. With this function, local solar time can be easily calculated from WGS84 longitude coordinates:

```
lon <- list(17.03887, 13.54010, 18.39900) # list of longitudes
names(lon) <- c('91171058', '91171062', '91171063') # locality names
data_clean <- mc_prep_meta(data_clean, lon, "lon_wgs84") # update metadata
data_tz <- mc_prep_solar_tz(data_clean) # calculate solar time
```

Raw time series downloaded from microclimatic loggers can contain duplicated measurements, measurements in the wrong order or missing measurements (Aalto et al., 2022; Man et al., 2022). Moreover, the logger's internal clock can drift, or the logger can be accidentally set to recording in unrounded time, for example, when recording starts at 13:07 instead of 13:00. To fix these problems, we developed the *mc_prep_clean* function, which keeps only the first duplicated measurements, reorders wrongly ordered measurements, and rounds up time series to the closest nice break (13:07 → 13:00). Note that the *mc_prep_clean* function corrects only these problems. It cannot fix other issues like wrong measurements, low contact of soil moisture sensor with the soil, overheating of air temperature sensor due to missing sun shield, or detect loggers dislocated by animals.

By default, the *mc_prep_clean* function prints the summary table of time series cleaning in the console: the number of loggers, date range and the list of detected steps in seconds and minutes. This summary table from data cleaning is directly associated with the *myClim* objects and can be displayed later using the *mc_info_clean* function. Usually, data cleaning with *mc_prep_clean* function is performed automatically already when reading data with *mc_read_data* and *mc_read_files* functions (default parameter clean = TRUE). Nevertheless, this default parameter can be changed to FALSE, and the data cleaning can be done separately with *mc_prep_clean* function. Below is an example of the output from the *mc_prep_clean* function called during data reading.

```
data <- mc_read_files("c:/TMS/", dataformat_name = "TOMST", clean = TRUE)
> 5 loggers
> datetime range: 2019-09-16 - 2021-07-09
> detected steps: 900s = 15min
```

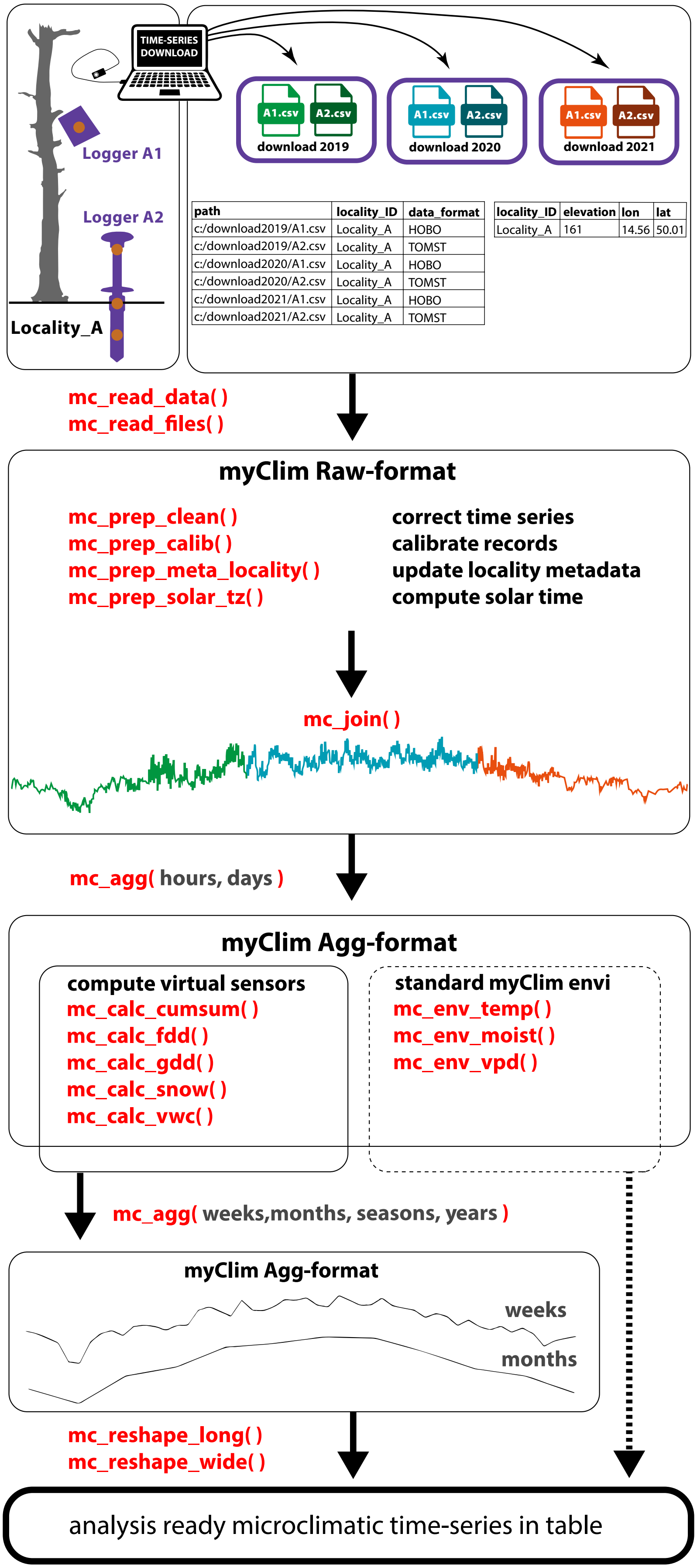

FIGURE 1 Workflow of microclimatic time series processing with the *myClim* R package.

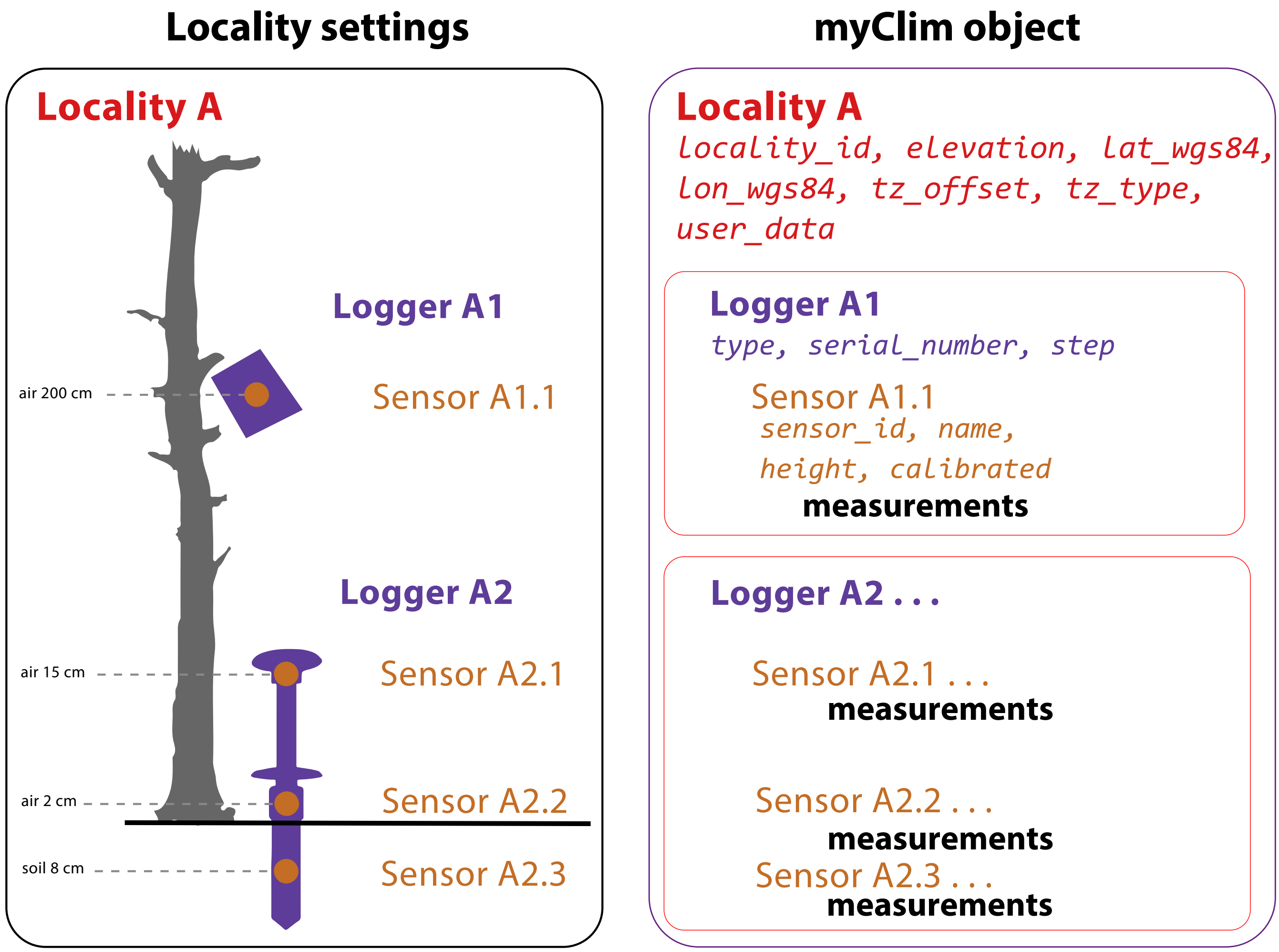

FIGURE 2 Schema of the *myClim* object in *Raw-format* with associated metadata. The locality (red) is the highest hierarchical level. On the locality, there can be one or more loggers (purple), and each logger can host one or more sensors (brown). Each hierarchical level of the *myClim* object can host its own metadata (italic). Microclimatic measurements are attached to the sensor, and time series are attached to the logger (*Raw-format*) or locality (*Agg-format*).

Locality id	Serial number	Start date	End date	Step seconds	Count duplicities	Count missing	Count disordered	Rounded
91171058	91171058	2020-11-22 14:45:00	2021-07-09 09:45:00	900	0	0	0	TRUE
91171062	91171062	2020-10-12 12:00:00	2021-05-20 14:15:00	900	0	0	0	FALSE
91171063	91171063	2020-09-28 10:45:00	2021-04-28 12:15:00	900	0	0	0	FALSE
91191256	91191256	2020-08-24 00:00:00	2021-06-03 07:15:00	900	95	9845	2	FALSE
94199122	94199122	2019-09-16 14:30:00	2020-12-08 10:15:00	900	182	1143	8	FALSE

1.4.2 | Sensor calibration

The low-cost sensors used in many microclimatic loggers have limited accuracy and measured values may be subject to systematic errors (Hubbart et al., 2005; Maclean et al., 2021; Navarro-Serrano et al., 2019). Therefore, we recommend the calibration of individual sensors before their deployment. The *myClim* package offers calibration functionality for correction of the measured values with sensor-specific correction factors, compensating for a constant error (using *correction factor*) or for a linearly increasing/decreasing error with measured value (using *correction slope* different from zero). The function *mc_prep_calib_load* first assigns correction factors and slopes to sensors in the *myClim* object and stores them as sensor metadata. Then, the *mc_prep_calib* function replaces the original values with corrected values calculated according to the formula (Equation 1).

$$\text{Corrected Value} = \text{Original Value} \cdot (1 + \text{correction slope}) + \text{correction factor}. \tag{1}$$

1.4.3 | Informative summaries

The functions *mc_info_count*, *mc_info*, *mc_info_clean* and *mc_info_meta* provide a general overview of the microclimatic time series stored in the *myClim* objects. The *mc_info_count* function returns the numbers of localities, loggers and sensors in the *myClim* object. The *mc_info* function returns the data frame with a summary per sensor (e.g. measurement timestep, first and last measurement date, minimum and maximum value, number of valid measurements and missing values). The *mc_info_clean* returns the data frame with the time series preprocessing log (e.g. the number of duplicated timesteps, number of measurements in the wrong order or missing measurements). The *mc_info_meta* returns the data frame with locality metadata (e.g. locality ID, coordinates, and elevation).

1.5 | Plotting

To facilitate data exploration, we designed two basic plotting functions. The *mc_plot_raster* function shows overall patterns across multiple localities (Figure 3). In contrast, the *mc_plot_lines* function shows individual lines for the time series of the sensors in one locality (Figure 4). Users can plot one or several sensors with the same physical units (e.g. temperatures measured at different heights) or plot sensors with two different physical units using the primary and secondary y-axis (e.g. soil temperature and moisture). The *myClim* plots are returned in the R environment as *ggplot* objects, which allows their further graphical adjustment with *ggplot* (Wickham, 2009).

1.6 | Processing

The function *mc_filter* subsets sensors and localities from the *myClim* object. The subsetting of localities is also possible with square brackets (e.g. tms[1]; tms[c("loc1", "loc2")]). The *mc_prep_merge* function combines several *myClim* objects together. The *mc_prep_merge* function combines all localities from all input objects and all sensors on identical localities.

The *mc_prep_fillNA* function fills small gaps (missing values) in microclimatic time series with simple linear interpolation between the first and last recorded values. It is particularly beneficial in cases where there are only a few missing measurements, such as those resulting from a brief sensor malfunction (the default maximum length of the filled gap is set to five missing measurements).

The metadata in the *myClim* object can be updated with *mc_prep_meta_locality* and *mc_prep_meta_sensor* functions. Using these functions, the user can rename locality, sensor or both. Sensor height provided in the metadata is used by *myClim* during joining time series from multiple downloads and, therefore, it is important to be set correctly. Some loggers have predefined sensor heights according to common practise, for example, the TOMST TMS with four sensors (temperature sensors: soil 8 cm, air 2 cm, air 15 cm; moisture sensor: soil 0–15 cm). Predefined sensor heights can be updated with the *mc_prep_meta_sensor* function.

1.7 | Joining time series

The local microclimate is increasingly measured over longer periods. Such long-term measurements require repeated downloads of the logger on the locality. The resulting consecutive time series need to be merged before the analysis. However, these time series may contain overlapping sections, gaps, or irrelevant measurements (e.g. measurements obtained before the field installation). Therefore, joining microclimatic time series cannot be fully automated and requires manual control.

The *mc_join* function combines multiple time series from the identical sensor type and with the same sensor height at each locality into a single, time-aligned time series using a semiautomated process. Duplicate time series fragments with identical measured values are automatically removed. In cases where overlapping parts of time series are not identical, *myClim* interactively asks the user to decide which of the conflicting time series should be used. If present, the temporal gaps between individual time series are automatically filled with NA's.

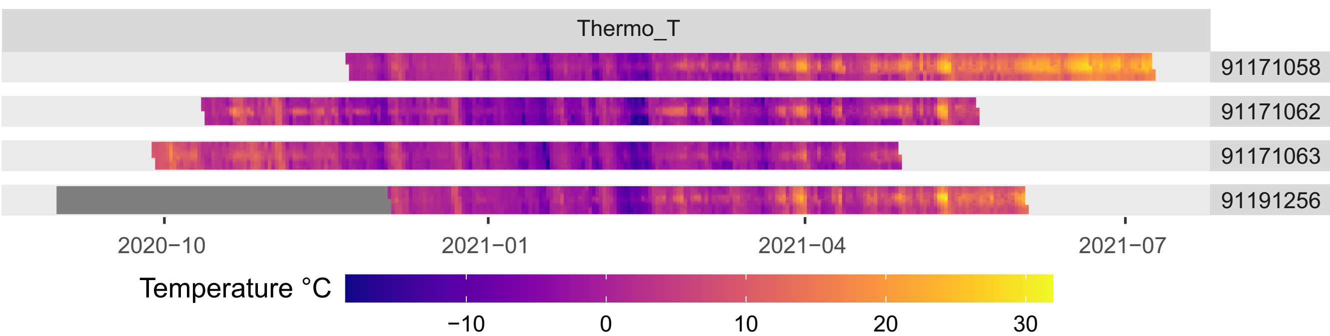

FIGURE 3 An example of *mc_plot_raster* output depicting raw time series of air temperature from TOMST Thermologgers. The measurements performed every 15 min at 2 m above the ground are displayed as faceted raster heatmaps with date on the x-axis and time of the day on the y-axis. The data were imported with *mc_read_files* without metadata; therefore, logger ID was used as a locality name. The logger 91191256 is potentially problematic since there are many missing values (shown in dark grey).

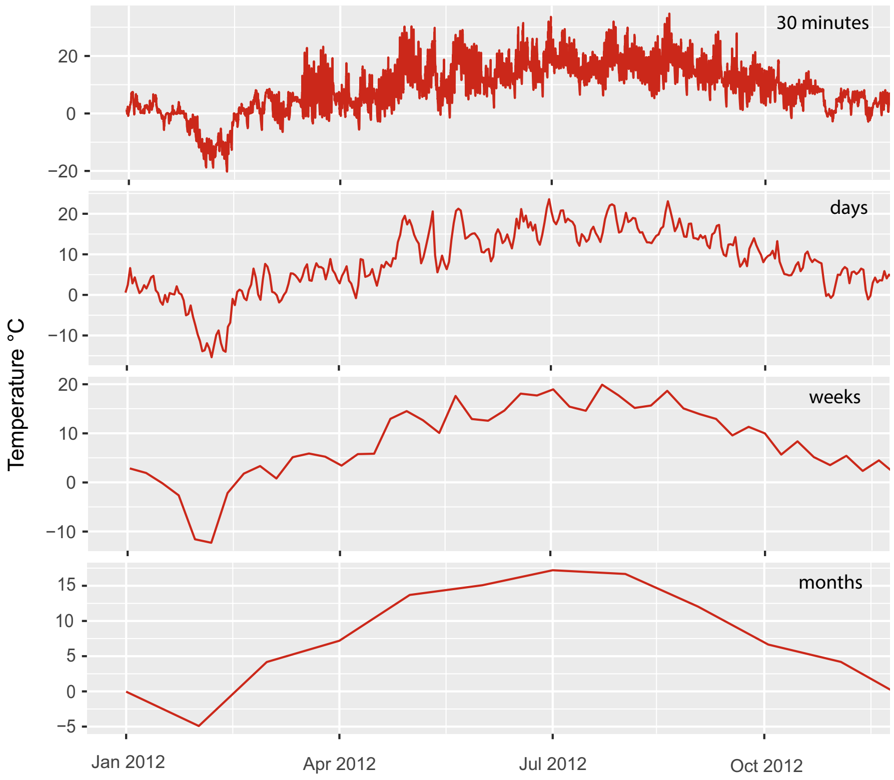

FIGURE 4 An example of a time series representing temperature measurements performed every 30 min aggregated by the *mc_agg* function to hour, day, week and month mean values and subsequently plotted with the *mc_plot_lines* function.

1.8 | Aggregating time series

Preprocessed, microclimatic data can be aggregated over user-defined timesteps with the *mc_agg* function, which simultaneously applies several numerical operations to single or multiple sensors (see code example below). The function has several predefined numerical operations (e.g. mean, range and percentile), but the user can also apply custom functions. Besides standard timesteps (e.g. hour, day, week, month and year), aggregation can also be carried out over user-defined periods using *custom_start* and *custom_end* parameters. The custom period works within an annual cycle. Thus, the user can aggregate microclimatic time series covering several years and gathers aggregated data for several growing seasons, winter seasons or hydrological years in one step.

```
# data30 = the myClim object with raw 30 min measurements
mean_day <- mc_agg(data30, period = "day", fun = "mean") # daily mean
mean_week <- mc_agg(data30, period = "week", fun = "mean") #
weekly mean
mean_month <- mc_agg(data30, period = "month", fun = "mean") #
monthly mean
```

1.9 | Microclimatic variables

The *myClim* package provides functions for the calculation of microclimatic variables from temperature, soil moisture and air humidity time series. All these functions add a new 'virtual' sensor representing a newly calculated variable to the *myClim* object with the same timestep as the input time series.

1.9.1 | mc_calc_cumsum

Cumulative sum of the values on selected sensor since the beginning of the time series. In units of the input sensor.

1.9.2 | mc_calc_gdd

Growing degree days (GDD, units °C · day) provide the contribution of each measurement to GDD as a positive difference between the actual temperature and the base temperature (default 5°C), divided by a fraction of a day represented by the measurement timestep. Values are returned as a virtual sensor with the same timestep as in the input time series. This allows the user to also consider shorter growing events than whole days, which would be otherwise ignored if GDD were calculated from the daily mean temperatures.

If the user prefers to calculate GDD from daily time series, it is possible first to aggregate data into daily timestep with *mc_agg(period = "day")* and then run *mc_calc_gdd* on this aggregated time series. To obtain the summed GDD values over longer periods (e.g. months, growing seasons, years), the user can employ the sum function for aggregation or *mc_calc_cumsum*.

1.9.3 | mc_calc_fdd

Freezing degree days (FDD, units °C · day) provide the contribution of each measurement to FDD as an absolute value of negative differences between the actual temperature and the base temperature (default 0°C), divided by a fraction of a day represented by the timestep measurement.

1.9.4 | mc_calc_snow

Snow cover detection [TRUE/FALSE] from temperature time series (Figure 5). All records within the user-defined period (the default is 1 day) are considered as snow-covered when the maximum temperature remains below a specified threshold value (default 0.5°C) and the temperature range does not exceed a defined threshold (default 2°C) on a selected temperature sensor. This function relies on the physical attributes of snow, decoupling temperatures under the snow from the variation in diurnal air temperature, and limiting the maximal temperature to the freezing point (Dickerson-Lange et al., 2015; Teubner et al., 2015). The default upper limit for temperature was set slightly above the freezing point of the water to account for measurement inaccuracy (~0.5°C for TOMST TMS loggers) and the effect of conductive heat flux from the soil that affects the sensor in contact with the ground.

1.9.5 | mc_calc_vwc

This function calculates the volumetric water content [m^3/m^3] from the raw moisture signal recorded by a TMS logger using a calibration function with user-specified empirical coefficients (Wild et al., 2019). The TMS raw moisture signal is slightly affected by soil temperature (Wild et al., 2019), and this temperature dependency is corrected by the *mc_calc_vwc* function, using the temperature of the TMS soil temperature sensor.

The relationship between the TMS raw moisture signal and the volume of water frozen in the soil is currently unknown (Wild et al., 2019) and therefore all values of volumetric water content in frozen soil (soil temperature <0°C) are replaced by NAs. This default and strongly recommended replacement, can be switched off by the user.

Coefficients of the calibration function used in the transformation from raw TMS units to volumetric water content can differ between soils according to their physical and chemical properties (e.g. bulk density, soil texture, organic matter content), and users are advised to use site-specific coefficients. When these site-specific coefficients are not available, the *myClim* user can choose coefficients for eight different soil types from Wild et al. (2019) or the universal coefficients from Kopecký et al. (2021).

1.9.6 | mc_calc_vpd

The vapour pressure deficit [kPa] calculation is based on air temperature and relative air humidity measurements, following the Magnus equation (adapted by Jones, 2013). This equation (Equation 2) also accounts for the effect of air pressure, which is calculated from site elevation specified directly as a function parameter, or in the *myClim* object locality metadata.

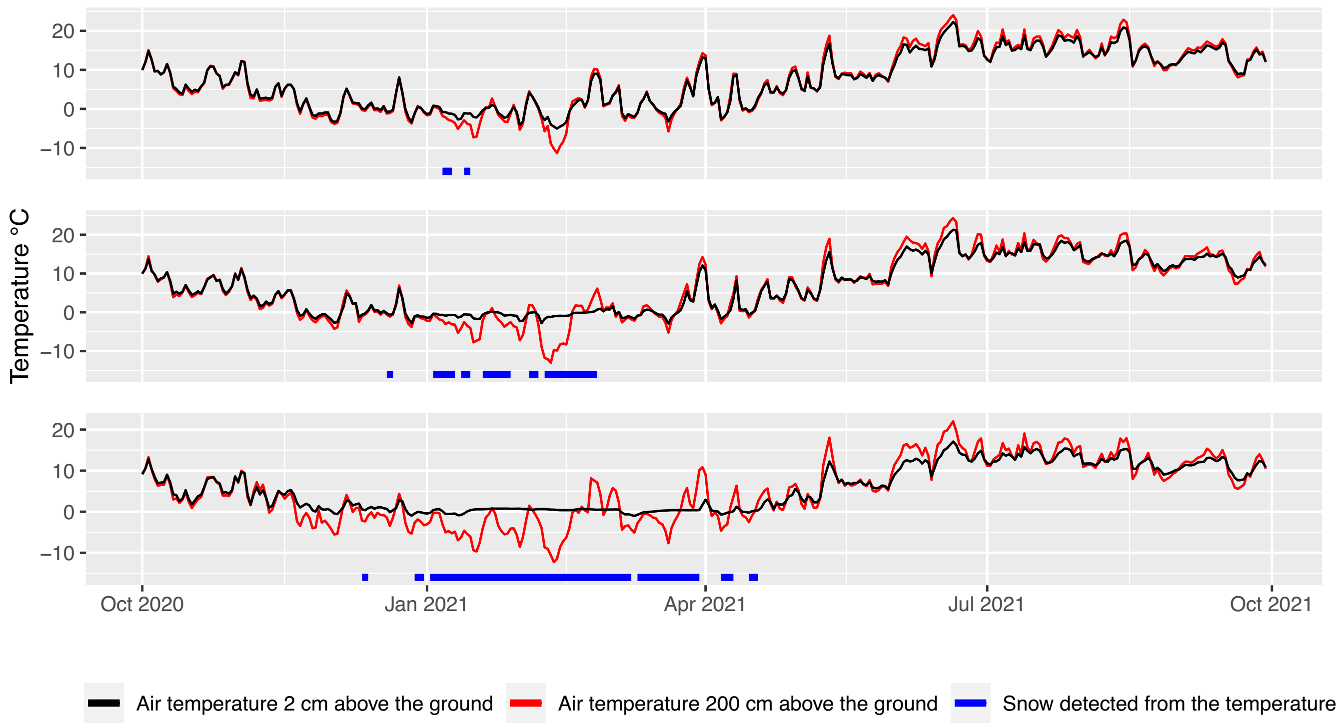

FIGURE 5 An example of the snow cover detection using *mc_calc_snow* function from near ground (+2 cm) air temperatures measured at three different localities in the Czech Republic.

$$\mathrm{VPD} = f \cdot a \cdot e^{\left(\frac{b \cdot T}{c+T}\right)} \cdot \left(1 - \frac{RH}{100}\right), \tag{2}$$

where T is the air temperature in degrees (°C), RH is the relative humidity in %, $a = 0.61121$, $b = 18.678 - (T/234.5)$, $c = 257.14$, f is the enhancement factor, which corrects for vapour pressure in moist air compared to pure water vapour: $f = 1.00072 + \left(10e-7 \cdot P \cdot \left(0.032 + 5.9 \cdot 10e-6 \cdot T^2\right)\right)$, and P is air pressure, which is estimated from an elevation: $P = 101300 \cdot e^{\left(-\frac{\text{elevation}}{8200}\right)}$.

1.10 | *myClim* set of microclimatic variables

To provide a standardised set of ecologically relevant microclimatic variables, we combine several *myClim* functions into three user-friendly wrapper functions: *mc_env_temp*, *mc_env_moist* and *mc_env_vpd*. In contrast to other *myClim* functions that return *myClim* objects, these wrapper functions return analysis-ready tables with a standardised set of environmental variables derived from time series of air/soil temperatures, soil moisture and relative air humidity (Table 2).

The *mc_env* functions work only with time series with steps equal to or shorter than 1 day. The *mc_env* automatically uses all available sensors in the *myClim* object and returns all possible variables based on sensor type and height/depth measurement (Table 2).

The *mc_env_temp function* first aggregates time series to a daily period and then aggregates to the final period specified by a user (e.g. month, year, growing season). Because FDDs and GDDs are always aggregated with the sum function, these two variables are not first aggregated to the daily timesteps.

The *mc_env_moist function* needs time series of volumetric water content (VWC) measurements as input. Therefore, the moisture measurements of the raw soil must be first converted to VWC. For TMS loggers, this can be done with the *mc_calc_vwc* function, which converts the raw TMS moisture signal into VWC and creates a new virtual VWC sensor. Since the daily oscillation of soil moisture is small, *mc_env_moist* works on the original VWC time series and does not perform prior daily aggregation as *mc_env_temp* and *mc_env_vpd* functions.

1.11 | Data export

After calculations and aggregations, results in the *myClim* format can easily be exported to the standard R data frame format, either with the functions *mc_reshape_wide* or *mc_reshape_long*. In both functions, the user can either reshape all data in the *myClim* object or select only specific localities and sensors. The first column of the wide table specifies the date and time; the accompanying columns are unique combinations of sensors and localities. To export multiple sensors from different localities, we recommend using a long format having only five columns (locality id, logger serial number, sensor name, date with time and value).

Finally, *mc_save* and *mc_load* functions save/load the *myClim* objects. We strongly recommend using these functions for long-term data backup, as the *myClim* objects saved and loaded with these functions will be compatible with future versions of the *myClim* package.

TABLE 2 The set of *myClim* environmental variables returned by the *mc_env* wrapper functions. Environmental variables are returned for each unique measurement height/depth and automatically named after the variable (T, VWC, VPD), height/depth of the sensor (e.g. air 15 cm; soil 8 cm; soil 0–15 cm) and calculation functions (e.g. mean, GDD5).

Description	Example
mc_env_temp	
Minimum temperature = 5th percentile of daily minimum temperatures	*T.air_15_cm.min5p*
Mean temperature = mean of daily mean temperatures	*T.air_15_cm.mean*
Maximum temperature = 95th percentile of daily maximum temperatures	*T.air_15_cm.max95p*
Temperature range = mean of daily temperature range (i.e. difference between daily minima and maxima)	*T.air_15_cm.drange*
Growing degree days = sum of degree days above base temperature (default 5°C)	*T.air_15_cm.GDD5*
Freezing degree days = sum of degree days below base temperature (default 0°C)	*T.air_15_cm.FDD0*
Frost days = number of days with frost (daily minimum <0°C)	*T.air_15_cm.frostdays*
mc_env_moist	
Minimum soil moisture = 5th percentile of VWC values	*VWC.soil_0_15_cm.5p*
Mean soil moisture = mean of VWC values	*VWC.soil_0_15_cm.mean*
Maximum soil moisture = 95th percentile of VWC values	*VWC.soil_0_15_cm.95p*
Standard deviation of VWC values	*VWC.soil_0_15_cm.sd*
mc_env_vpd	
Mean vapour pressure deficit = mean of daily mean VPD	VPD.air_150_cm.mean
Maximum vapour pressure deficit = 95th percentile of daily maximum VPD	VPD.air_150_cm.max95p

2 | DISCUSSION AND FUTURE OUTLOOK

The *myClim* package supports various data sources, including the most common microclimate dataloggers as well as the import of simple data tables from weather stations, climate reanalyses (Kalnay et al., 1996; Muñoz-Sabater et al., 2021) and mechanistic microclimate models (Kearney & Porter, 2017; Maclean et al., 2019). The *myClim* package thus provides a unified framework that enhances data compatibility across studies and stimulates comparisons among the outputs of climate and microclimate models and in-situ measurements. The handling of microclimate data has traditionally involved several processing steps using various packages and approaches. The level of data quality checking and cleaning was, therefore, heavily dependent on the skills and experience of an individual researcher. Adopting *myClim* will standardise microclimatic data workflow across studies, thus facilitate large-scale syntheses, boosting data sharing and increase the comparability and reproducibility of microclimatic studies.

Adopting *myClim* will also facilitate data sharing within the global microclimatic community, for example, through the SoilTemp database (Lembrechts et al., 2020). The *myClim* package can already be connected to TubeDB, an open-source database designed to handle climate station data (Wöllauer et al., 2021). In the future, we plan to provide also *myClim* functions for automatic data reshaping to the SoilTemp database format and possibly also for direct data download from the SoilTemp.

The development of *myClim* is an ongoing effort, and we will further expand its functionality. Nevertheless, we also welcome active user contributions, preferably through opening new issues or submitting pull requests on GitHub (https://github.com/ibot-geoecology/myClim). Our future plans include the implementation of semiautomatic data quality control mechanisms for the detection of compromised records, such as those from TMS loggers pulled out from the soil, records not originating from the field, or suspicious outliers from local microclimatic logger networks.

3 | CONCLUSION

Here, we described the *myClim* R package for microclimate data processing, storage and analyses. The *myClim* package provides a complete workflow for microclimate data handling, including a reading of raw data files from microclimatic loggers, their preprocessing and cleaning, time series aggregation, calculation of ecologically relevant microclimatic variables, and flexible data export options. The *myClim* R package thus implements the complete microclimatic workflow from the import of the raw microclimatic time series to the calculation of ecologically relevant variables in a standardised and fully reproducible manner using open-source code. Such technical advance is crucial for much-needed global data syntheses and will facilitate wider incorporation of microclimate into global change biology and ecology.

AUTHOR CONTRIBUTIONS

Matěj Man, Vojtěch Kalčík, Martin Macek, Josef Brůna, Lucia Hederová, Jan Wild and Martin Kopecký jointly conceived the initial idea and made significant contributions to writing the first draft. Matěj Man, Vojtěch Kalčík and Martin Macek took the lead in software development. Josef Brůna, Lucia Hederová and Martin Macek provided valuable assistance with data curation. Jan Wild and Martin Kopecký played key roles in providing supervision and conceptualisation. All authors actively participated in the development of the package and made important contributions to editing and providing feedback on manuscript drafts.

ACKNOWLEDGEMENTS

We thank all people who provided constructive feedback during the development of *myClim*. We also thank Miska Luoto for hosting us in beautiful Finland during the initial writing and both reviewers and the editor for useful comments and suggestions on the manuscript. This study was supported by the Czech Science Foundation (projects GACR 20-28119S and 23-06614S), the Czech Academy of Sciences (project RVO 67985939) and the Technology Agency of the Czech Republic (project TACR SS02030018).

CONFLICT OF INTEREST STATEMENT

No conflict of interest has been declared.

PEER REVIEW

The peer review history for this article is available at https://www.webofscience.com/api/gateway/wos/peer-review/10.1111/2041-210X.14192.

DATA AVAILABILITY STATEMENT

The stable version of *myClim* package, documentation, tutorial and example data are available on CRAN (https://cran.r-project.org/web/packages/myClim). The development version and source code can be assessed at GitHub (https://github.com/ibot-geoecology/myClim) and version 1.0.8 of the package used for this manuscript is archived on Zenodo (Man et al., 2023).

ORCID

Matěj Man https://orcid.org/0000-0002-4557-8768
Martin Macek https://orcid.org/0000-0002-5609-5921
Josef Brůna https://orcid.org/0000-0002-4839-4593
Lucia Hederová https://orcid.org/0000-0003-1283-0952
Jan Wild https://orcid.org/0000-0003-3007-4070
Martin Kopecký https://orcid.org/0000-0002-1018-9316

REFERENCES

Aalto, J., Tyystjärvi, V., Niittynen, P., Kemppinen, J., Rissanen, T., Gregow, H., & Luoto, M. (2022). Microclimate temperature variations from boreal forests to the tundra. *Agricultural and Forest Meteorology*, *323*, 109037. https://doi.org/10.1016/j.agrformet.2022.109037

Bramer, I., Anderson, B. J., Bennie, J., Bladon, A. J., De Frenne, P., Hemming, D., Hill, R. A., Kearney, M. R., Körner, C., Korstjens,

A. H., Lenoir, J., Maclean, I. M. D., Marsh, C. D., Morecroft, M. D., Ohlemüller, R., Slater, H. D., Suggitt, A. J., Zellweger, F., & Gillingham, P. K. (2018). Advances in monitoring and modelling climate at ecologically relevant scales. *Advances in Ecological Research*, *58*, 101–161. https://doi.org/10.1016/BS.AECR.2017.12.005

De Frenne, P., Lenoir, J., Luoto, M., Scheffers, B. R., Zellweger, F., Aalto, J., Ashcroft, M. B., Christiansen, D. M., Decocq, G., De Pauw, K., Govaert, S., Greiser, C., Gril, E., Hampe, A., Jucker, T., Klinges, D. H., Koelemeijer, I. A., Lembrechts, J. J., Marrec, R., … Hylander, K. (2021). Forest microclimates and climate change: Importance, drivers and future research agenda. *Global Change Biology*, *27*(11), 2279–2297. https://doi.org/10.1111/GCB.15569

Dickerson-Lange, S. E., Lutz, J. A., Martin, K. A., Raleigh, M. S., Gersonde, R., & Lundquist, J. D. (2015). Evaluating observational methods to quantify snow duration under diverse forest canopies. *Water Resources Research*, *51*(2), 1203–1224. https://doi.org/10.1002/2014WR015744

Finocchiaro, M., Médail, F., Saatkamp, A., Diadema, K., Pavon, D., & Meineri, E. (2023). Bridging the gap between microclimate and microrefugia: A bottom-up approach reveals strong climatic and biological offsets. *Global Change Biology*, *29*(4), 1024–1036. https://doi.org/10.1111/gcb.16526

Geiger, R., Aron, R. H., & Todhunter, P. (2009). *The climate near the ground* (7th ed.). Rowman & Littlefield.

Greiser, C., Ehrlén, J., Meineri, E., & Hylander, K. (2020). Hiding from the climate: Characterizing microrefugia for boreal forest understory species. *Global Change Biology*, *26*(2), 471–483. https://doi.org/10.1111/gcb.14874

Holden, Z. A., Klene, A. E., Keefe, R. F., & Moisen, G. G. (2013). Design and evaluation of an inexpensive radiation shield for monitoring surface air temperatures. *Agricultural and Forest Meteorology*, *180*, 281–286. https://doi.org/10.1016/j.agrformet.2013.06.011

Hubbart, J., Link, T., Campbell, C., & Cobos, D. (2005). Evaluation of a low-cost temperature measurement system for environmental applications. *Hydrological Processes*, *19*(7), 1517–1523. https://doi.org/10.1002/hyp.5861

Jones, H. G. (2013). *Plants and microclimate: A quantitative approach to environmental plant physiology* (Vol. *9780521279*). Cambridge University Press. https://doi.org/10.1017/CBO9780511845727

Kalnay, E., Kanamitsu, M., Kistler, R., Collins, W., Deaven, D., Gandin, L., Iredell, M., Saha, S., White, G., Woollen, J., Zhu, Y., Chelliah, M., Ebisuzaki, W., Higgins, W., Janowiak, J., Mo, K. C., Ropelewski, C., Wang, J., Leetmaa, A., … Joseph, D. (1996). The NCEP/NCAR 40-year reanalysis project. *Bulletin of the American Meteorological Society*, *77*(3), 437–471. https://doi.org/10.1175/1520-0477(1996)077<0437:TNYRP>2.0.CO;2

Kearney, M. R., & Porter, W. P. (2017). NicheMapR—An R package for biophysical modelling: The microclimate model. *Ecography*, *40*(5), 664–674. https://doi.org/10.1111/ecog.02360

Kopecký, M., Macek, M., & Wild, J. (2021). Topographic wetness index calculation guidelines based on measured soil moisture and plant species composition. *Science of the Total Environment*, *757*(1), 143785. https://doi.org/10.1016/j.scitotenv.2020.143785

Körner, C. (2021). The climate plants experience. In *Alpine plant life* (pp. 65–88). Springer International Publishing. https://doi.org/10.1007/978-3-030-59538-8_4

Lembrechts, J. J., Aalto, J., Ashcroft, M. B., De Frenne, P., Kopecký, M., Lenoir, J., Luoto, M., Maclean, I. M. D., Roupsard, O., Fuentes-Lillo, E., García, R. A., Pellissier, L., Pitteloud, C., Alatalo, J. M., Smith, S. W., Björk, R. G., Muffler, L., Ratier Backes, A., Cesarz, S., … Nijs, I. (2020). SoilTemp: A global database of near-surface temperature. *Global Change Biology*, *26*(11), 6616–6629. https://doi.org/10.1111/gcb.15123

Lundquist, J. D., & Lott, F. (2008). Using inexpensive temperature sensors to monitor the duration and heterogeneity of snow-covered areas. *Water Resources Research*, *44*(4). https://doi.org/10.1029/2008WR007035

Lundquist, J. D., Roche, J. W., Forrester, H., Moore, C., Keenan, E., Perry, G., Cristea, N., Henn, B., Lapo, K., McGurk, B., Cayan, D. R., & Dettinger, M. D. (2016). Yosemite Hydroclimate Network: Distributed stream and atmospheric data for the Tuolumne River watershed and surroundings. *Water Resources Research*, *52*(9), 7478–7489. https://doi.org/10.1002/2016WR019261

Macek, M., Kopecký, M., & Wild, J. (2019). Maximum air temperature controlled by landscape topography affects plant species composition in temperate forests. *Landscape Ecology*, *34*(11), 2541–2556. https://doi.org/10.1007/s10980-019-00903-x

Maclean, I. M. D., Duffy, J. P., Haesen, S., Govaert, S., De Frenne, P., Vanneste, T., Lenoir, J., Lembrechts, J. J., Rhodes, M. W., & Van Meerbeek, K. (2021). On the measurement of microclimate. *Methods in Ecology and Evolution*, *12*(8), 1397–1410. https://doi.org/10.1111/2041-210X.13627

Maclean, I. M. D., Mosedale, J. R., & Bennie, J. J. (2019). Microclima: An r package for modelling meso- and microclimate. *Methods in Ecology and Evolution*, *10*(2), 280–290. https://doi.org/10.1111/2041-210X.13093

Man, M., Kalčík, V., Macek, M., Brůna, J., Hederová, L., Wild, J., & Kopecký, M. (2023). myClim: Version 1.0.8; archive only. *Zenodo*, https://doi.org/10.5281/ZENODO.8136147

Man, M., Wild, J., Macek, M., & Kopecký, M. (2022). Can high-resolution topography and forest canopy structure substitute microclimate measurements? Bryophytes say no. *Science of the Total Environment*, *821*, 153377. https://doi.org/10.1016/j.scitotenv.2022.153377

Muñoz-Sabater, J., Dutra, E., Agustí-Panareda, A., Albergel, C., Arduini, G., Balsamo, G., Boussetta, S., Choulga, M., Harrigan, S., Hersbach, H., Martens, B., Miralles, D. G., Piles, M., Rodríguez-Fernández, N. J., Zsoter, E., Buontempo, C., & Thépaut, J.-N. (2021). ERA5-land: A state-of-the-art global reanalysis dataset for land applications. *Earth System Science Data*, *13*(9), 4349–4383. https://doi.org/10.5194/essd-13-4349-2021

Nadeau, C. P., Giacomazzo, A., & Urban, M. C. (2022). Cool microrefugia accumulate and conserve biodiversity under climate change. *Global Change Biology*, *28*(10), 3222–3235. https://doi.org/10.1111/gcb.16143

Nadeau, C. P., Urban, M. C., & Bridle, J. R. (2017). Coarse climate change projections for species living in a fine-scaled world. *Global Change Biology*, *23*(1), 12–24. https://doi.org/10.1111/gcb.13475

Navarro-Serrano, F., López-Moreno, J. I., Azorin-Molina, C., Buisán, S., Domínguez-Castro, F., Sanmiguel-Vallelado, A., Alonso-González, E., & Khorchani, M. (2019). Air temperature measurements using autonomous self-recording dataloggers in mountainous and snow covered areas. *Atmospheric Research*, *224*(March), 168–179. https://doi.org/10.1016/j.atmosres.2019.03.034

Potter, K. A., Woods, H. A., & Pincebourde, S. (2013). Microclimatic challenges in global change biology. *Global Change Biology*, *19*(10), 2932–2939. https://doi.org/10.1111/gcb.12257

Roznik, E. A., & Alford, R. A. (2012). Does waterproofing Thermochron iButton dataloggers influence temperature readings? *Journal of Thermal Biology*, *37*(4), 260–264. https://doi.org/10.1016/j.jtherbio.2012.02.004

Slavich, E., Warton, D. I., Ashcroft, M. B., Gollan, J. R., & Ramp, D. (2014). Topoclimate versus macroclimate: How does climate mapping methodology affect species distribution models and climate change projections? *Diversity and Distributions*, *20*(8), 952–963. https://doi.org/10.1111/ddi.12216

Terando, A. J., Youngsteadt, E., Meineke, E. K., & Prado, S. G. (2017). Ad hoc instrumentation methods in ecological studies produce highly biased temperature measurements. *Ecology and Evolution*, *7*(23), 9890–9904. https://doi.org/10.1002/ece3.3499

Teubner, I. E., Haimberger, L., & Hantel, M. (2015). Estimating snow cover duration from ground temperature. *Journal of Applied Meteorology and Climatology*, *54*(5), 959–965. https://doi.org/10.1175/JAMC-D-15-0006.1

Whiteman, C. D., Hubbe, J. M., & Shaw, W. J. (2000). Evaluation of an inexpensive temperature datalogger for meteorological applications. *Journal of Atmospheric and Oceanic Technology*, *17*(1), 77–81. https://doi.org/10.1175/1520-0426(2000)017<0077:EOAITD>2.0.CO;2

Wickham, H. (2009). *Elegant graphics for data analysis (ggplot2)*. Springer.

Wild, J., Kopecký, M., Macek, M., Šanda, M., Jankovec, J., & Haase, T. (2019). Climate at ecologically relevant scales: A new temperature and soil moisture logger for long-term microclimate measurement. *Agricultural and Forest Meteorology*, *268*, 40–47. https://doi.org/10.1016/j.agrformet.2018.12.018

Wöllauer, S., Zeuss, D., Hänsel, F., & Nauss, T. (2021). TubeDB: An on-demand processing database system for climate station data. *Computers & Geosciences*, *146*, 104641. https://doi.org/10.1016/j.cageo.2020.104641

Zellweger, F., De Frenne, P., Lenoir, J., Vangansbeke, P., Verheyen, K., Bernhardt-Römermann, M., Baeten, L., Hédl, R., Berki, I., Brunet, J., Van Calster, H., Chudomelová, M., Decocq, G., Dirnböck, T., Durak, T., Heinken, T., Jaroszewicz, B., Kopecký, M., Máliš, F., … Coomes, D. (2020). Forest microclimate dynamics drive plant responses to warming. *Science*, *368*(6492), 772–775. https://doi.org/10.1126/science.aba6880

SUPPORTING INFORMATION

Additional supporting information can be found online in the Supporting Information section at the end of this article.

Appendix S1: Schema of the *myClim objects* (i) *Raw-format* and (ii) *Agg-format*. The schema is helpful to navigate through the myClim objects using standard R syntax [] $ @ if necessary.

Appendix S2: The tutorial (practical code exaples) how to use *myClim* for microclimatic data processing in R.

How to cite this article: Man, M., Kalčík, V., Macek, M., Brůna, J., Hederová, L., Wild, J., & Kopecký, M. (2023). *myClim*: Microclimate data handling and standardised analyses in R. *Methods in Ecology and Evolution*, *00*, 1–13. https://doi.org/10.1111/2041-210X.14192