climate4R: An R-based Open Framework for Reproducible Climate Data Access and Post-processing

M. Iturbide[a,b], J. Bedia[b,c], S. Herrera[b], J. Baño-Medina[a], J. Fernández[b], M. D. Frías[b], R. Manzanas[a],
D. San-Martín[c], E. Cimadevilla[b], A.S. Cofiño[b], J. M. Gutiérrez[a,*]

[a]*Meteorology Group. Instituto de Física de Cantabria (CSIC - Univ. de Cantabria), Santander, 39005, Spain*
[b]*Meteorology Group. Dpto. de Matemática Aplicada y Ciencias de la Computación. Universidad de Cantabria, Santander, 39005, Spain*
[c]*Predictia Intelligent Data Solutions, CDTUC, Santander, 39005, Spain*

Abstract

Climate-driven sectoral applications commonly require different types of climate data (e.g. observations, reanalysis, climate change projections) from different providers. Data access, harmonization and post-processing (e.g. bias correction) are time-consuming error-prone tasks requiring different specialized software tools at each stage of the data workflow, thus hindering reproducibility. Here we introduce `climate4R`, an R-based climate services oriented framework tailored to the needs of the vulnerability and impact assessment community that integrates in the same computing environment harmonized data access, post-processing, visualization and a provenance metadata model for traceability and reproducibility of results. `climate4R` allows accessing local and remote (OPeNDAP) data sources, such as the Santander User Data Gateway (UDG), a THREDDS-based

*Corresponding author
Email address: `gutierjm@unican.es` (J. M. Gutiérrez)

service including a wide catalogue of popular datasets (e.g. ERA-Interim, CORDEX, etc.). This provides a unique comprehensive open framework for end-to-end sectoral reproducible applications. All the packages, data and documentation for reproducing the experiments in this paper are available from `http://www.meteo.unican.es/climate4R`.

Keywords:
open science , climate indices, CMIP5, downscaling, climatic change, NetCDF-Java

---

1. Introduction

Climate data retrieval, harmonization and post-processing (e.g. bias correction) are inherent tasks for climate vulnerability and impact assessment (VIA) studies in a number of sectors such as agriculture, energy, hydrology, ecology, health or wildfires among others (see, e.g. Casanueva et al., 2014; Ewert et al., 2015; Wang et al., 2017; Challinor et al., 2018; Walsh et al., 2018; Turco et al., 2018). Typically, these sector-specific applications require data for a reduced number of surface variables from different sources (e.g. observations, reanalysis and/or global and regional climate change projections), which can be directly obtained from different data providers and/or accessed through specialized data gateways such as the Earth System Grid Federation (ESGF; Williams et al., 2015). However, the resulting formats, spatial and temporal scales and aggregations or vocabularies (variable naming and units) are, as a rule, inhomogeneous across the different data sources. Moreover, some common transformation/calibration and post-processing steps are typically applied to raw model data before their use in sectoral applications, including data collocation (e.g. regridding, temporal aggregation, or subsetting) and bias adjustment or downscaling (e.g. local scaling, quantile mapping, analogs or regression). In some cases, these steps are very technical and require different specialized tools entailing multiple specific choices that are often insufficiently documented in practical applications. As a result, obtaining and harmonizing climate data is typically an error-prone and time consuming task, often preventing from an accurate replication of the research outcomes. The difficulty of carrying out such processes remain as an important factor hampering the full exploitation of available climate data to generate actionable information leading to an "usability gap" (Lemos et al., 2012).

In order to bridge the usability gap, this paper presents a new R-based framework for climate studies, tailored to the specific needs of the VIA community, and branded as `climate4R`. R (R Core Team, 2017) is nowadays a very popular computing environment with powerful statistical modeling tools and excellent support for time series and spatial analysis, that has favoured its notable uptake by the climate community. `climate4R` has been developed as a set of seamlessly integrated packages designed to ease climate data access (`loadeR`), collocation and transformation (`transformeR`), bias correction and downscaling (`downscaleR`) and visualization (`visualizeR`), including full documentation via wikis and guided examples. Moreover, additional functionalities from existing external packages have been bridged via specific `climate4R` wrapping packages so they can be transparently used within the same framework. An example of external package integration is `climdex.pcic` (Bronaugh, 2015), which implements the climate extremes indices defined by the Expert Team on Climate Change Detection and Indices (ETCCDI, Karl et al., 1999). Finally, a provenance metadata model for traceability and reproducibility of results has been developed based on META-

CLIP (METAdata for CLImate Products, `http://www.metaclip.org`), so full metadata (including the source code) can be produced for all products generated by `climate4R`.

`climate4R` is aimed at fostering research transparency and reproducibility, issues of major concern in all experimental disciplines (see the special issue on reliability and reproducibility of published research `http://go.nature.com/huhbyr`). For example, Baker (2016) recently reported that the work published in Earth and Environment Science were mostly (over two-thirds) not reproducible. As a result, there is growing concern among the scientific community about results that cannot be reproduced. With this regard, one of the main objectives of `climate4R` is to improve transparency and reproducibility of results.

Following with the above-mentioned study by Baker (2016), the main difficulties for research reproducibility identified include 1) access restrictions to raw input data and/or results, 2) methods or code unavailable and 3) incomplete metadata documentation of the particular workflow followed to obtain a climate product. In order to circumvent these problems, the following actions have been undertaken in `climate4R`:

1. Data sources: All the data needed for the experiments described in this paper are publicly available at the Santander User Data Gateway (UDG, `http://www.meteo.unican.es/udg-wiki`), a data service seamlessly integrated with the `climate4R` framework, thus enabling a single entry point for users to a wide variety of harmonized datasets, including global and regional climate projections from the Coupled Model Intercomparison Project Phase 5 (CMIP5; Taylor et al., 2011a) and the COordinated Regional climate Downscaling EXperiment (CORDEX; Giorgi and Gutowski, 2015)

respectively (see Sec. 3 for further details).

2. Source Code: All the R packages forming `climate4R` are publicly available through the GitHub repository `http://www.github.com/SantanderMetGroup`. Moreover, the full code to reproduce all the results presented in this work (as well as extended examples) are included as auxiliary material as a paper notebook `https://github.com/SantanderMetGroup/notebooks`.
3. Metadata: The R structures handled by `climate4R` are built upon the common data model described in Sec. 2, and emphasis has been put on the inclusion of all the necessary metadata for object description, including spatiotemporal collocation details (dates/times, coordinates, geographical projection, temporal resolution, etc.) and other relevant descriptors required for their adequate characterization. Furthermore, `climate4R` is integrated within the METACLIP framework, envisaged to tackle the problem of climate product provenance description. METACLIP is based on semantics exploiting web standard Resource Description Framework (RDF, W3C, 2004), through the design of domain-specific extensions of standard vocabularies (e.g., PROV-O; PROV Working Group, 2013; Moreau et al., 2015) describing the workflow stages producing a climate product (see `http://www.metaclip.org` for more details and worked examples, including a full provenance description of Fig. 2a in this paper).

As a result, `climate4R` provides a unique framework for climate processing where most common tasks can be straightforwardly performed using a few lines of code, allowing end-to-end experimental reproducibility and facilitating the description (metadata) and documentation of the whole data flow. Although this paper focuses on the application of `climate4R` to climate change problems, this framework also allows to work with climate predictions, such as seasonal forecasts, an aspect that is separately described in Cofiño et al. (2018), with further example research applications presented in Bedia et al. (2018a) and Frías et al. (2018).

This article is structured as follows: Section 2 describes the core components of `climate4R`. Sections 3 and 4 provide further aspects and details on the Data Services Layer and the bias correction tools, respectively. Sections 5 and 6 present two illustrative case studies. The first example describes the application to calculate and bias-correct future projections of a standard ETCCDI climate index (summer days, `http://etccdi.pacificclimate.org`) for a Southern European domain using locally stored CORDEX data. The second example illustrates an extended case study accessing CORDEX data remotely from the Santander UDG. Final conclusions are provided in Sec. 7.

2. The climate4R Framework

The `climate4R` data model is based on the Grid Feature Type (for gridded data) and the Station Time Series Feature (for point data, e.g. stations or individual gridbox values) implemented in the Unidata's Common Data Model version 4 (CDM[1]). As such, the `climate4R` data access layer builds on Java to interpret these CDM features (see Sec. 3) which are inherited by the R data/metadata structures. The coordinate system for each object type includes, at least, the time and position dimensions (latitude and longitude for grids and location for point data). Besides the standard regular geographic coordinates, `climate4R` also

[1] `https://www.unidata.ucar.edu/software/thredds/current/netcdf-java/CDM/`

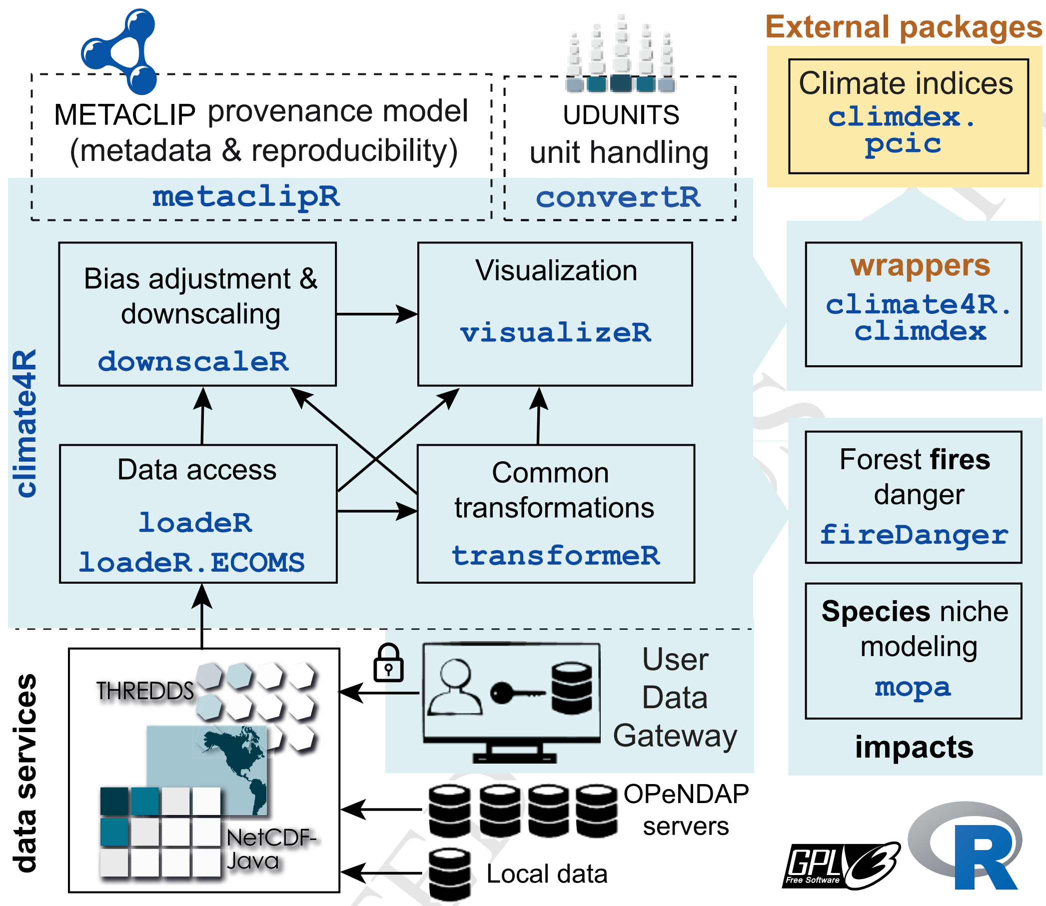

Figure 1: Schematic illustration of the `climate4R` framework consisting of three layers: (a) Data services building on NetCDF-Java and THREDDS in order to load local or remote (exposed via a THREDDS OPeNDAP service) data, and also datasets from the in-house Santander User Data Gateway (UDG); (b) The `climate4R` R bundle for data access and post-processing, formed by four core packages for data loading, transformation, downscaling (including bias correction) and visualization. These core packages are the basis for other sector-specific packages for impact analysis (e.g. forest fires, species distribution modelling, etc.) which further extend the `climate4R` capabilities. (c) External packages, which are connected to `climate4R` via specific wrapper packages. (d) Additional `climate4R` packages for extended functionality, including provenance metadata model (based on METACLIP) or unit handling (based on UDUNITS). The arrows indicate the possible data flows and the blue shading differentiates the in-house developments. All components are distributed under GNU General Public License. The THREDDS, NetCDF-Java and UDUNITS logos are courtesy of UCAR/Unidata. The R logo is ©2016 The R Foundation. The RDF icon used by METACLIP is ©1994-2006 W3C.

handles rotated-pole and Lambert conformal conic projections used in CORDEX gridded datasets[2]. Both grids and point datasets are transparently handled by all relevant `climate4R` functions.

Furthermore, the basic `climate4R` data structure includes additional dimensions, such as the *member*, which allows to work with ensembles. For instance, this extra dimension is used when loading seasonal predictions using the `loadeR.ECOMS` extension of the `loadeR` package (see Cofiño et al., 2018, for more details), tailored to the specific needs of the seasonal forecasting community. The member dimension can be also used to construct multi-model ensembles. This poses several advantages from the user point of view, as next highlighted in case study 2 (Sec. 6). For instance, most of the `climate4R` operations (e.g. index calculation and aggregation) are implemented to deal with grids containing the `member` dimension and therefore, the necessary looping over several members is done behind the scenes. Furthermore, the use of members is also beneficial from the computational point of view, since most relevant functions have the option to parallelize across members through the optional argument `parallel`, thus providing ease of use and computational efficiency.

A description of the core R packages forming the `climate4R` framework is next presented (see Fig. 1 for a schematic representation):

`loadeR` (Bedia et al., 2018b) is the central building-block of the `climate4R` bundle allowing to transparently access local and remote climate datasets (through the OPeNDAP service, see `https://www.opendap.org`) building on NetCDF-Java (see Sec. 3 for more details). Moreover, `loadeR` is

[2]`http://is-enes-data.github.io/cordex_archive_specifications.pdf`

the interface to the Santander User Data Gateway (UDG), a THREDDS-based (Unidata, 2006) service from the Santander Climate Data Service providing access to several climate datasets popular in impact studies. A comprehensive description of functionalities of this package is given in the `loadeR`'s wiki (`https://github.com/SantanderMetGroup/loadeR/wiki`), as well as installation instructions and worked examples. An extension of `loadeR` to work with climate predictions is also available (`loader.ECOMS`), dealing with the initialization time (or lead time) selection in a user-friendly way (see Cofiño et al., 2018).

`transformeR` (Bedia et al., 2018c) performs common data processing tasks such as regridding/interpolation, subsetting or spatio-temporal aggregation, among others. Unlike `downscaleR`, all the post-processing operations performed by `transformeR` do not necessarily entail a second reference observational dataset. Examples of application are available in the transformeR's wiki (`https://github.com/SantanderMetGroup/transformeR/wiki`).

`downscaleR` (Bedia et al., 2017) performs bias correction (see Sec. 4 for more details) and statistical downscaling. An introduction to the package and examples of application are available in the downscaleR's wiki (`https://github.com/SantanderMetGroup/downscaleR/wiki`).

`visualizeR` (Frías et al., 2018) performs climate data visualization, implementing basic visualization functionalities for gridded and point-based data, time series, and a set of advanced tools for forecast visualization in a form suitable to communicate the underlying uncertainty, such as tercile plots, bubble plots, climagrams, reliability categories, etc. Examples and further functionalities are detailed in the visualizeR's wiki (`https://github.com/SantanderMetGroup/visualizeR`).

Besides these core packages, `climate4R` extends its capabilities by integrating the functionalities of other external packages via auxiliary wrapping packages. For instance, the wrapper `climate4R.climdex` allows to transparently compute the 27 ETCCDI core indices implemented in the `climdex.pcic` R package[3].

Furthermore, advanced unit checking and conversion can be achieved at any point during the data analysis via the `climate4R` package `convertR` (Bedia and Herrera, 2018), that exploits the Unidata's UDUNITS-2 software libraries (Unidata, 2017) —a widely used standard containing an extensive and user-extensible unit database in XML format— through its R binding package `udunits2` (Hiebert, 2016). More information is available in the `convertR` GitHub repository (`https://github.com/SantanderMetGroup/convertR`).

In addition to the core and external `climate4R` packages, there are also specific packages for some sectoral applications, such as `fireDanger` (Bedia et al., 2018a, implementing several popular fire-weather and drought indices) or `mopa` (Iturbide et al., 2018, providing tools for species distribution modelling), which are integrated within the `climate4R` framework. With this regard, the `climate4R` data model has been conceived to minimize external dependencies and ease interoperability, relying on basic R data structures. Conversion to other data formats is straightforward for specific applications when needed, thus providing a flexible framework for interacting with other packages of the R ecosystem according to

[3]`http://github.com/pacificclimate/climdex.pcic`

Dataset	Type	Resolution(s)	Scenario	Members	Ref
WFDEI	Observations	0.50°	-	1	Weedon et al. (2014)
EWEMBI	Observations	0.50°	-	1	Lange (2016)
E-OBS	Observations	0.25° (0.22° rot)	-	1	Haylock M. R. et al. (2008)
Spain02	Observations	0.11° (0.1° rot)	-	1	Herrera et al. (2012, 2016)
ERA-Interim	Reanalysis	2°	-	1	Dee D. P. et al. (2011)
JRA55	Reanalysis	2°	-	1	Kobayashi et al. (2015)
CMIP5	Projections	2°	RCP4.5,8.5	10 GCMs	Taylor et al. (2011b)
EURO-CORDEX	Projections	0.44°, 0.11°	RCP4.5,8.5	12 RCMs	Jacob et al. (2014)
AFRICA-CORDEX	Projections	0.44°	RCP4.5,8.5	12 RCMs	Nikulin et al. (2012)

Table 1: Summary of the main public climate datasets available at the Santander User Data Gateway (UDG). For brevity, the datasets for seasonal forecasting are not included here (see Cofiño et al., 2018, and `http://meteo.unican.es/ecoms-udg/catalog` for details).

the specific user's needs. For instance, spatial data conversion to `Spatial-class` objects (Bivand et al., 2013) is internally done in `visualizeR` for specific geographical data representations, while `mopa` exploits the `raster-class` capabilities (Hijmans, 2017) to handle static climatological layers.

The following two sections provide further information on two aspects of `climate4R` of special relevance for better understanding the illustrative examples provided in this paper: the climate services layer and the available bias correction methods.

3. Data Services Layer

There is a number of R packages supporting read/write operations on NetCDF files, like `ncdf`, `ncdf4` (Pierce, 2017), `RNetCDF` (Michna, 2014) and `raster` (Hijmans, 2017), all of them supporting both NetCDF-3 and 4 with the exception of ncdf which only supports the older NetCDF-3 file format and has been therefore removed from the R-CRAN repository since 2016. loadeR goes beyond the file-oriented concept for data access, supporting reading (and writing) CDM datasets, i.e. "collections" of NetCDF files, instead of individual files. Unlike the file-based approach, the most immediate advantage from the user point of view of using such collections is that one does not need to worry about a particular directory tree structure or file naming schema when the required data is split into several files (usually due to size constraints), and only one single URL pointing to the dataset need to be used, as if all the data was contained in a single "file". loadeR allows for a direct creation of such CDM datasets from R (function makeAggregatedDataset), so multiple CDM files can be conveniently combined ("aggregated") along the selected dimension(s), a process that is fully automatized for the most usual cases that users typically face after raw data retrieval from external repositories/servers. This entails for instance joining different files of the same variable along the specified dimensions (e.g, joining files along time) and/or performing unions of different variables stored in separate files to obtain a single multi-variable dataset. However, loadeR is also able to read from single files if preferred by the user, following exactly the same procedure as reading from CDM datasets.

By exploiting the capabilities of the NetCDF-Java libraries built upon Unidata's CDM (Sec. 2), loadeR also allows for an efficient access to remote datasets via OPeNDAP, providing users a transparent access to the data regardless of whether these are stored locally or remotely. This is internally achieved through the rJava package (Urbanek, 2016) that provides a low-level interface between R and the Java virtual machine. In addition, not only NetCDF, but also a variety of other geoscientific data formats (HDF, GRIB, etc.) can be aggregated to produce CDM datasets via the NetCDF Markup Language (NcML) and accessed by `loadeR` using identical code. NcML is an XML dialect that allows not only creating CDM datasets, but also to modify (rename, add, delete and/or restructure) the data and metadata of the original NetCDF files and/or CDM datasets, without the need of modifying the original files.

3.1. The Santander User Data Gateway

Besides local and remote OPeNDAP datasets, `climate4R` is transparently connected to the User Data Gateway (UDG), from the Santander Climate Data Service hosted by University of Cantabria (`http://meteo.unican.es/udg-wiki`) consisting of two main components: (1) A THREDDS Data Server (TDS) and (2) the THREDDS Access Portal (TAP), which provide standard services for data access (e.g. OPeNDAP or the NetCDF Subset Service –NCSS–) and user management and authentication (based on data policies associated with virtual datasets), respectively. The UDG provides harmonized access to a variety of common datasets typically used in sectoral applications, including state-of-the-art global and regional climate projections such as those from CMIP5 (Taylor et al., 2011a) and CORDEX (Giorgi and Gutowski, 2015). Thus, the UDG represents a one-stop-service for climate data access where users can efficiently retrieve the subsets best suited to their particular research aims (for particular regions, periods and/or ensemble members) and where dataset access is controlled through a fine-grained authorization scheme depending on the different data policies (there is a wide variety of datasets of public access through the PUBLIC role, see Table 1).

4. Bias Correction Methods

The R package `downscaleR` implements several statistical downscaling (analogs, generalized linear regression, neural networks, etc.) and bias correction (scaling, parametric and empirical quantile mapping, etc.) methods, some of which have been already used and tested in the VALUE initiative (Gutiérrez et al., 2018). In this paper we focus on bias correction methods, which adjust model outputs, e.g. maximum temperature in this paper, using as reference the corresponding local observations (either point-wise stations or an interpolated grid, E-OBS in this paper). Bias correction methods are trained over a representative historical period (typically 30 years), and then applied to correct model outputs for a test (or future) period. Due to their simplicity and straightforward application, these methods have become very popular during the last decade and have been used in numerous recent papers covering different forecast temporal horizons. However, it is important to understand their assumptions and limitations in order to avoid the misuse of these techniques (see, e.g., Maraun et al., 2017; Manzanas et al., 2017b).

The `biasCorrection` function is the workhorse to apply several standard bias correction techniques, ranging from the simplest local-scaling to more sophisticated parametric or empirical quantile-quantile mapping approaches. Next, we provide a brief description of the two bias correction methods that are used in this work (for further information on all available methods, the reader is referred to the downscaleR's wiki):

Local-scaling: This method is specified by the argument `method = "scaling"`. It consists in scaling the predictions with an additive (`scaling.type = "additive"`) or multiplicative (`scaling.type = "multiplicative"`) factor, which is obtained as the difference/ratio between the predicted and the observed mean in the train period. The additive version is preferable for unbounded variables (e.g. temperature) and the multiplicative is typically used with variables with lower bound = 0 (e.g. precipitation or wind speed).

Empirical quantile mapping (EQM): This method is applied using the argument `method = "eqm"`. The EQM method does not make any assumption about the statistical distribution of the variable and consists in calibrating the empirical predicted Cumulative Distribution Function (CDF) by adjusting the model quantiles towards the observed ones (Déqué, 2007). The optional argument `n.quantiles` allows to specify the number of quantiles to be adjusted (by default, percentiles are used for the correction). Moreover, different extrapolation alternatives can be selected via the parameter `extrapolation`. For the case of precipitation, the frequency adaptation proposed by Themeßl et al. (2012) is applied by default when the predicted frequency of dry days is larger than the observed one. A precise description of the EQM method, as used in this paper, is provided in Appendix A of Gutiérrez et al. (2018).

Additionally, in order to tackle the issue of seasonality —and also model drift in seasonal forecasting (see, e.g., Manzanas, 2016),— the optional argument `window` allows to specify the center and width of a moving time window (calendar days) that can be used for independently correcting consecutive periods (e.g. months or seasons), instead of the total available period at once. Moreover, `biasCorrection` deals with the `ensemble` dimension, allowing to separately correct each member (`join.members = FALSE`, e.g. for multi-model ensembles in climate change applications), or to use the joint ensemble distribution as reference (`join.members = TRUE`, e.g. for different members of a seasonal forecast system, that are by definition statistically indistinguishable).

Furthermore, all bias correction methods can be applied in cross-validation mode with the argument `cross.val` (see the downscaleR's wiki for examples of application), which allows for leave-one-out (`"loo"`) and k-fold (`"kfold"`) cross-validation schemes (see, e.g., Maraun et al., 2015; Manzanas et al., 2017a).

In order to promote a collaborative development of the bias correction methods, these are implemented as atomic functions that receive vectors as input (observations, predictions and, for methods requiring calendar information, the corresponding dates), so contributors do not need to worry about the particularities and complexities of internal metadata handling. `biasCorrection` recursively applies these methods to the N-dimensional arrays of the `climate4R` data model, according to the different optional arguments provided (e.g. cross-validation method, parallel computing options, window size, etc.) and performing metadata update as required.

5. Example 1: Climate Indices from CORDEX Projections

The main functionalities of `climate4R` are showcased describing the complete workflow needed to compute and bias correct an ETCCDI climate index (implemented in the R package `climdex.pcic`, Bronaugh, 2015, see also `http://etccdi.pacificclimate.org/list_27_indices.shtml`) from locally stored EURO-CORDEX Regional Climate Model (RCM) data (Jacob et al., 2014). In particular, in this example we consider the projections of summer days (SU) — defined as the number of days with maximum temperature $> 25°C$— for a single model over a Mediterranean domain. The second case study (Sec. 6) will further expand on this example illustrating a more comprehensive analysis that builds a multi-model ensemble from EURO-CORDEX data, retrieved remotely from the Santander UDG.

In the following, some code is interwoven within the text in order to illustrate the main package functionalities (the lines of code are identified by the R prompt symbol “>”). As a first step, the `climate4R` packages can be installed[4] from the GitHub repository using the `devtools` package:

```
> library(devtools)
> install_github(c("SantanderMetGroup/loadeR",
                "SantanderMetGroup/loadeR.java",
                "SantanderMetGroup/transformeR",
                "SantanderMetGroup/visualizeR",
                "SantanderMetGroup/downscaleR",
                "SantanderMetGroup/climate4R.climdex")
```

5.1. Loading, collocating and harmonizing data

In this section, we show the `climate4R` data access capabilities (including on-the-fly temporal aggregation and filtering), in order to directly load monthly summer days (SU) from the original maximum daily temperature data. However, only a reduced set of indices can be directly obtained in this way. Thus, in Sec. 5.3 we revisit this example working with the original daily data. This leads to a

[4] loadeR depends on package rJava, which might present installation problems as reported by some users. See the related loadeR’s Wiki section for help and installation recommendations: `https://github.com/SantanderMetGroup/loadeR/wiki/Installation`

more general approach where a variety of indices can be computed using, e.g., the `climdex.pcic` package implementing the 27 ETCCDI core indices (which include SU).

First, we describe the use of `loadeR` to load data subsets from the two datasets used in this example: (1) remote E-OBS gridded observations from the E-OBS OPeNDAP server[5], and (2) locally stored regional climate projections from a particular EURO-CORDEX RCM (for both the historical and the RCP8.5 scenarios) previously downloaded from ESGF —see Appendix A—.

The following call to the function `loadGridData` retrieves the E-OBS maximum temperature (`var = "tx"`) field of the full year (`season = 1:12`), from a single remote NetCDF file (`dataset = eobs_url`), considering a Mediterranean spatial domain (`lonLim = c(-10, 20)`, `latLim = c(35, 46)`) for a historical period (`years = 1971:2000`). In order to compute the SU index on-the-fly at a monthly scale, optional arguments are used both for data filtering (`condition = "GT"`, `threshold = 25`, to indicate the binary filtering "strictly greater than 25") and aggregation (`aggr.m = "sum"`, to indicate the monthly aggregation function).

```
> library(loadeR)
> eobs_url <- "http://opendap.knmi.nl/knmi/thredds/
 dodsC/e-obs_0.25regular/tx_0.25deg_reg_v17.0.nc"
> SU <- loadGridData(dataset = eobs_url,
                     var = "tx",
```

[5] The E-OBS dataset URL is not persistent, being updated with each new version of the dataset. Please check the ECA&D site for the current E-OBS version and its corresponding active OPeNDAP URL at `http://opendap.knmi.nl/knmi/thredds/e-obs/e-obs-catalog.html`

```
                    season = 1:12,
                    years = 1971:2000,
                    lonLim = c(-10, 20),
                    latLim = c(35, 46),
                    aggr.m = "sum",
                    condition = "GT",
                    threshold = 25)
```

Data transformation (e.g. regridding or additional temporal aggregation), is facilitated by the various functions of the `transformeR` package, and visualization capabilities are provided by the `visualizeR` package. For instance, the following commands perform annual aggregation and plot the climatological map of the resulting annual SU index:

```
> library(transformeR); library(visualizeR)
> SU <- aggregateGrid(SU, aggr.y = list(FUN = "sum"))
> # Generates Figure 2a:
> spatialPlot(climatology(SU))
```

EURO-CORDEX regional climate change projections from the RCA RCM — driven by the EC-EARTH GCM— can be loaded in a similar way. The NetCDF files of these simulations were downloaded from ESGF and stored locally (as detailed in Appendix A):

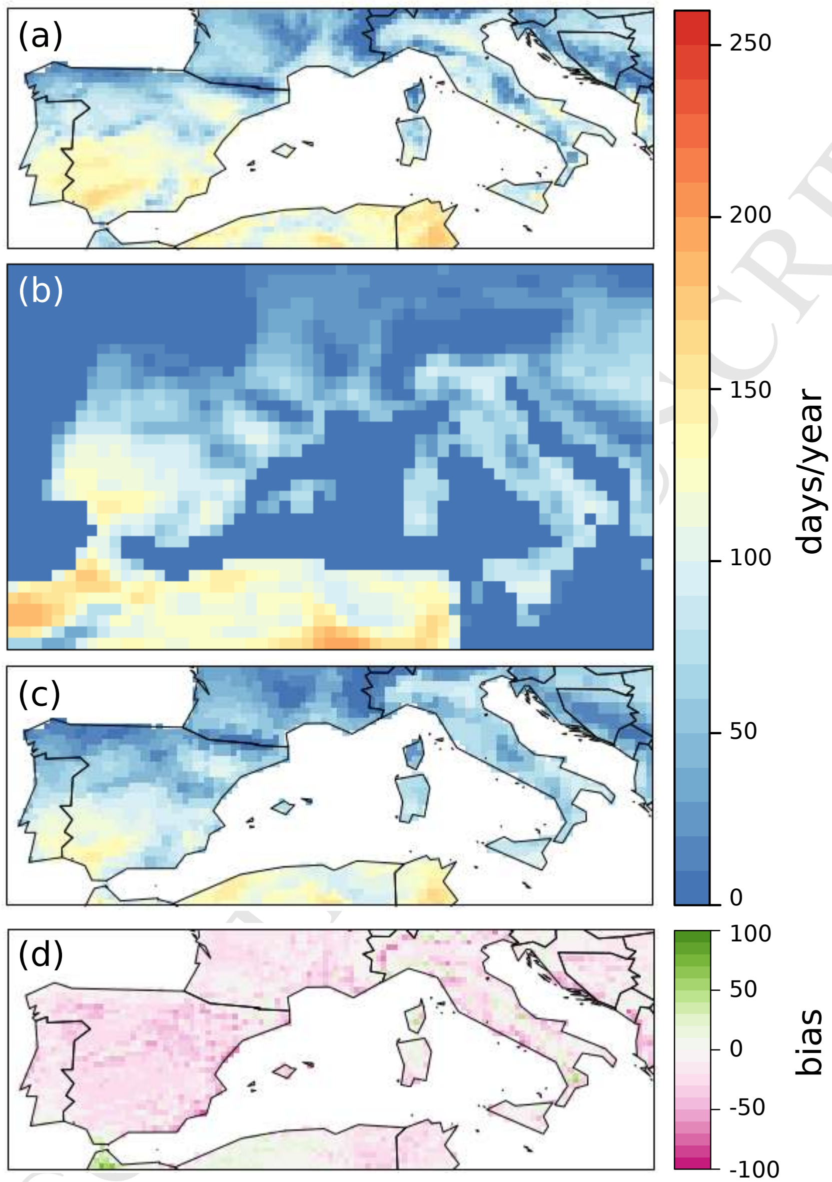

Figure 2: Annual climatology of Southern Europe summer days (ETCCDI SU index) for the reference period 1971-2000 according to: (a) 0.22° E-OBS gridded observations dataset, (b) 0.44° RCA regional climate model (driven by EC-EARTH GCM, historical scenario), (c) same as (b), but after regridding onto the regular E-OBS grid and (d) RCM bias (days/year) w.r.t. E-OBS.

```
> dir <- "/myDirectoryHistoricalScenario/"
> list.files(dir, recursive = TRUE)
# [1] "tasmax_EUR-44_EC_hist_SMHI-RCA4_2006-2010.nc"
# [2] "tasmax_EUR-44_EC_hist_SMHI-RCA4_2011-2015.nc"
# [3] "tasmax_EUR-44_EC_hist_SMHI-RCA4_2016-2020.nc"
...
```

Note that, in this case, five-year periods are stored in separate files. As explained in Sec. 2, one key strength of `loadeR` is that, in addition to single files —which can be directly loaded with `loadGridData` as in the previous E-OBS case—, it can transparently work with collections of files (catalogs) with a single access point (given by a NcML file; see Sec. 3 for more details) . This greatly facilitates data access, separating the logical structure of files from the way these are accessed. The following code shows the use of functions `makeAggregatedDataset` and `dataInventory` to write a catalog including the information contained in the files within a particular directory (in this case 19 files containing maximum temperature data for the period 2006-2100), and to display an overview of the dataset from the resulting NcML file (`CDX_hist.ncml` in this example):

```
> makeAggregatedDataset(source.dir = dir,
                        recursive = TRUE,
                        ncml.file = "CDX_hist.ncml")
> di <- dataInventory("CDX_hist.ncml")
> str(di$tasmax)
# List of 4
#  $ Description: chr "Daily Maximum Near-Surf...
```

```
#  $ DataType   : chr "float"
#  $ Units      : chr "K"
#  $ Dimensions :List of 3
#   ..$ time:List of 4
#   .. ..$ Type      : chr "Time"
#   .. ..$ TimeStep  : chr "1.0 days"
#   .. ..$ Units     : chr "days since 1949-12-0...
#   .. ..$ Date_range: chr "2006-01-01T12:00:00Z...
#   ..$ lat :List of 3
#   .. ..$ Type  : chr "GeoY"
#   .. ..$ Units : chr "degrees"
#   .. ..$ Values: num [1:103] -23.2 -22.8 -22.3...
#   ..$ lon :List of 3
#   .. ..$ Type  : chr "GeoX"
#   .. ..$ Units : chr "degrees"
#   .. ..$ Values: num [1:106] -28.2 -27.8 -27.3...
```

Note that the units of this dataset are given in Kelvin (*K*). Therefore, harmonization with E-OBS units (*degC*) is required. This can be done using the function ‘udConvertGrid‘ from package ‘convertR‘ (see Sec. 2) after data load, or directly on load using the harmonization capability implemented in `climate4R` through the definition of a standard vocabulary (complying with the UDUNITS standards) and the possibility to create raw-to-standard dictionaries for particular datasets. The `climate4R` standard vocabulary is displayed by function `C4R.vocabulary`:

```
> C4R.vocabulary()
#   identifier       standard_name         units
...
```

```
# 17   tas      2-meter air temperature    degC
# 18  tasmax  maximum 2-m air temperature  degC
# 19  tasmin  minimum 2-m air temperature  degC
# 21    pr     total precipitation amount   mm
...
```

A dictionary is a text file including simple unit conversion parameters (*offset* and *scale*) as well as temporal characterization attributes (further information can be found in the wiki `https://github.com/SantanderMetGroup/loadeR/wiki/Harmonization`). The construction of a dictionary for a dataset should be carefully performed (with the help of `dataInventory`) and may require detailed information from the data owner (e.g. temporal attributes). The dictionary file is usually saved locally —for instance together with the dataset— for its repeated usage (further instructions on dictionary usage are given in the `loadGridData` help menu). For better reproducibility, in the following code chunk a dictionary for the CORDEX RCM dataset is created on-the-fly as a temporary file to convert the raw maximum temperature units (*K*) to the stand ones (*degC*). Note that the code for this variable is the same (`tasmax`) in the CORDEX and standard vocabularies, as specified in the dictionary with `short_name` and `identifier`, respectively.

```
> dic <- tempfile(pattern = "cordex", fileext = ".dic")
> writeLines(c(
  "identifier,short_name,time_step,lower_time_bound,
     upper_time_bound, cell_method,offset,scale,
     deaccum,derived,interface",
  "tasmax,tasmax,24h,0,24,max,-273.15,1,0,0,"), dic)
```

The dictionary can be passed to `loadGridData` by the optional argument `dictionary = dic`; otherwise the original data would be loaded in its original units:

```
> SUh <- loadGridData(dataset = "CDX_hist.ncml",
                      var = "tasmax",
                      season = 1:12,
                      lonLim = c(-10, 20),
                      latLim = c(35, 46),
                      years = 1971:2000,
                      aggr.m = "sum",
                      threshold = 25,
                      condition = "GT",
                      dictionary = dic)
> SUh <- aggregateGrid(SUh, aggr.y = list(FUN = "sum"))
> # Generates Fig 2b:
> spatialPlot(climatology(SUh))
```

Note that the CORDEX RCM data is provided in rotated coordinates (Figure 2b) and therefore, regridding is needed in order to compare the results with E-OBS, so basic arithmetic operations can be applied (e.g. 'difference' to obtain the bias). This can be achieved using the `interpGrid` function. It uses the nearest gridbox by default, but additionally, two different bilinear interpolation implementations are available. In this example, the rotated coordinates of the RCM are interpolated onto the regular E-OBS grid:

```
> SUh <- interpGrid(SUh, getGrid(SU))
> # Generates Fig 2c:
```

```
> spatialPlot(climatology(SUh))
> bias <- gridArithmetics(SUh, SU, operator = "-")
> # Generates Fig 2d:
> spatialPlot(climatology(bias))
```

Similar data access and regridding operations are followed to load the projections of RCP 8.5 scenario (e.g. for the period 2071-2100), obtaining the future summer days (`SUf`, Figure 3a) and the climate change signal (`delta`, Figure 3b), as the difference with the historical signal (see the auxiliary notebook for the full code).

Note that the results obtained from CORDEX are affected by systematic biases —see Fig. 2d,— which prevent their direct use in most impact studies. Therefore, these results are typically post-processed in order to adjust the bias using *bias correction* techniques.

5.2. Post-processing: Bias Correction

The function `biasCorrection` of package `downscaleR` allows applying a number of standard bias correction techniques within the `climate4R` framework (see Sec. 4). In particular, when dealing with monthly data (as in the present example), the common bias correction technique is the (additive and/or multiplicative) local scaling method (Sec. 4). The projections of future summer days (`newdata = SUf`) are corrected using the method calibrated using the historical model as training data ("predictor", `x = SUh`) and the observed reference data ("predictand", `y = SU`):

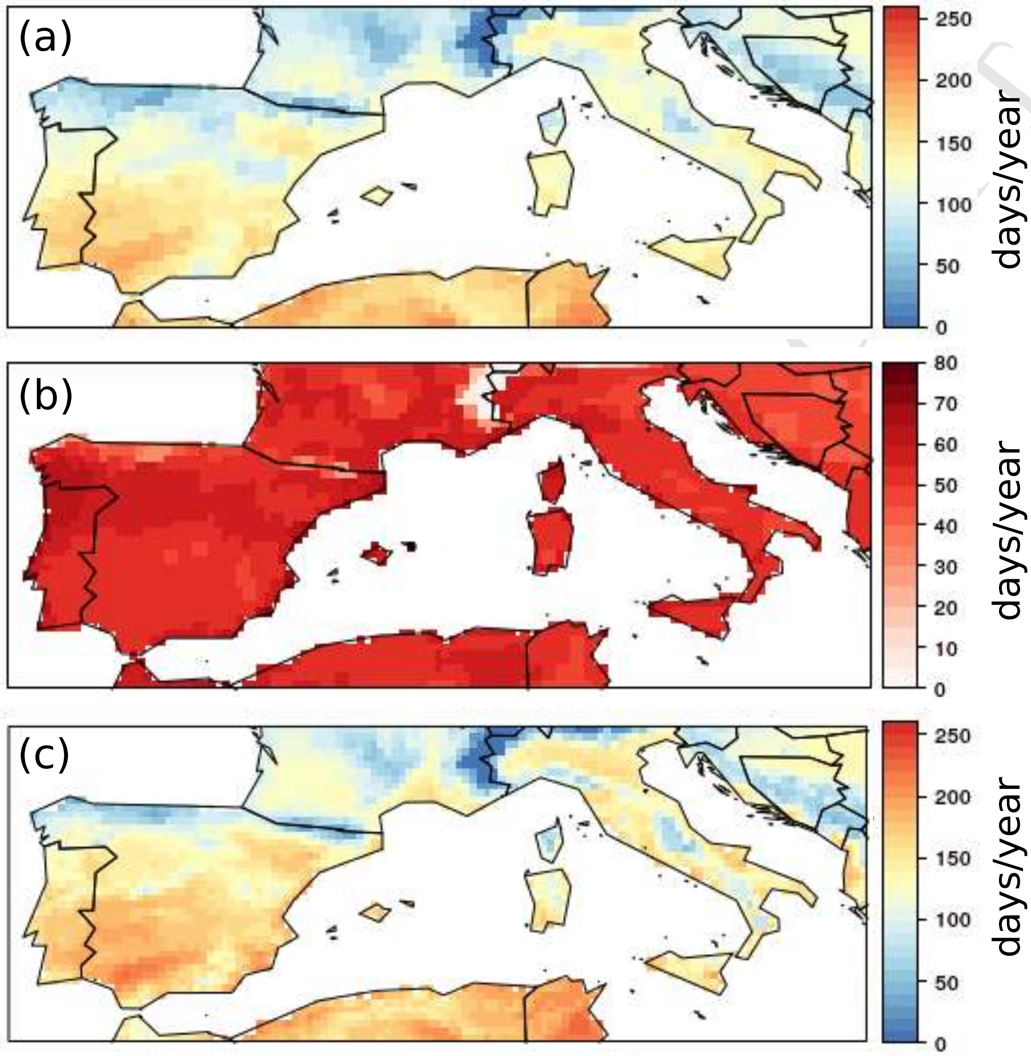

Figure 3: Climatology of Southern Europe annual SU (summer days) for the future period 2071-2100: (a) RCA (EC-EARTH driven, RCP8.5 scenario) RCM, (b) climate change signal (delta) w.r.t. the historical 1971-2000 RCA value —Figure 2c—, (c) bias corrected (additive scaling, based on E-OBS) results.

```
> library(downscaleR)
> SUf.bc <- biasCorrection(y = SU,
                           x = SUh,
                           newdata = SUf,
                           method = "scaling",
                           scaling.type = "additive")
> SUf.bc <- aggregateGrid(SUf.bc,
                          aggr.y = list(FUN = "sum"))
> # Generates Fig 3c:
> spatialPlot(climatology(SUf.bc))
```

The function `temporalPlot` displays temporal series for several datasets and periods on the same plot. `temporalPlot` is based on the powerful `lattice` package (Sarkar, 2008) and therefore, fine-tuning plotting parameters can be passed through the argument `xyplot.custom` (see the auxiliary notebook). In this case, we are plotting the series of a single gridbox, the one closest to Zaragoza (with coordinates `latLim = 41.64`, `lonLim = -0.89`).

```
> # Generates Fig. 4:
> temporalPlot("E-OBS" = SU,
             "CDX_hist" = SUh,
             "CDX_rcp85" = SUf,
             "CDX_rcp85_corrected" = SUf.bc,
             latLim = 41.64, lonLim = -0.89,
             cols = c("black", "red", "red", "blue"))
```

The resulting figure (Fig. 4) shows the inter-annual SU time series for the selected gridbox point (Zaragoza), highlighting the large model bias (red) *w.r.t.*

the observations (black) in the historical period. This figure also shows how bias correction compensates for this bias when applied to the future period (red *vs* blue for 2071-2100).

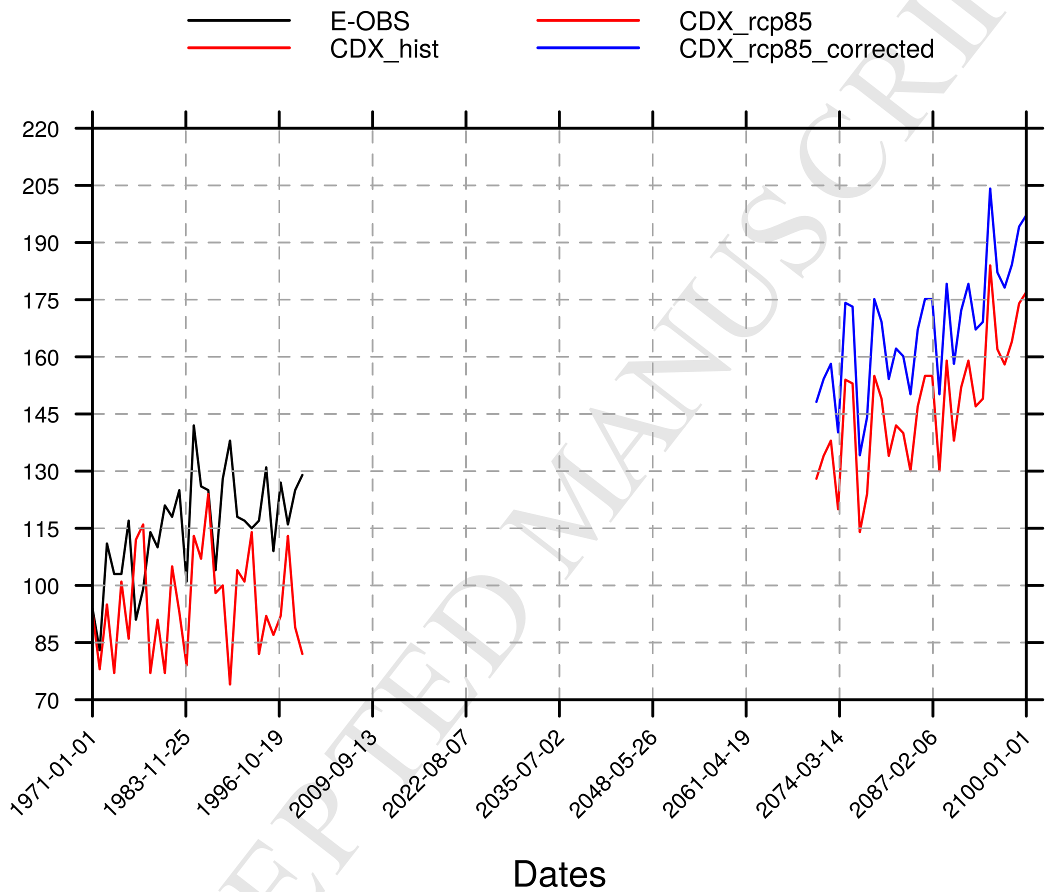

Figure 4: Annual summer days time series for a single gridbox (the one closest to Zaragoza, in the Ebro valley, Spain) for the observations (E-OBS) and the projection (original and bias corrected) in the historical and future periods.

5.3. Working with daily data

Loading aggregated data (monthly in the example above) is a useful feature allowing for an efficient use of memory. However, as we already mentioned, only a reduced set of indices can be directly obtained in this way. Therefore, in this section we revisit this example considering a more general approach using daily data and the `climate4R.climdex` package for index calculation (a wrapper of `climdex.pcic`, implementing the 27 ETCCDI core indices).

The data loading process for E-OBS (`TX`) and the historical (`TXh`) and future (`TXf`) RCM data is similar to the previous cases, but omitting the aggregation and filtering options. For instance the historical period can be loaded by:

```
> TXh <- loadGridData(dataset = "CDX_hist.ncml",
                      var = "tasmax",
                      season = 1:12,
                      lonLim = c(-10, 20),
                      latLim = c(35, 46),
                      years = 1971:2000,
                      dictionary = dic)
```

In this case, it is possible to apply bias correction methods better suited for daily data than local scaling, before calculating the index. For instance, in the example below we use empirical quantile mapping (`method = "eqm"`) with a moving window of 30 days to correct each 7-day time interval (see Sec. 4 for EQM method description and argument explanation):

```
> TXf.bc <- biasCorrection(y = TX,
                           x = TXh,
```

```
                         newdata = TXf,
                         method = "eqm",
                         window = c(30, 7),
                         extrapolation = "constant")
> SUf <- climdexGrid(tx = TXf, index.code = "SU")
> SUf.bc <- climdexGrid(tx = TXf.bc, index.code = "SU")
> # Generates Fig. 5:
> spatialPlot(climatology(SUf.bc))
```

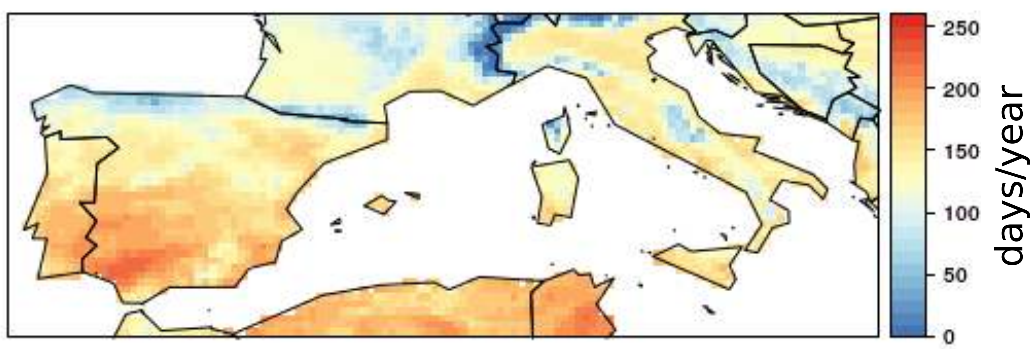

Figure 5: As Figure 3c, but for the index computed from bias corrected (empirical quantile mapping) daily maximum temperature data.

The resulting bias-corrected index (Fig. 5) is only slightly different to the one computed with monthly data in the previous section (Figures 3c). Therefore, both bias correction approaches lead to similar results in this case (see Casanueva et al., 2018, for further discussion on direct vs component-wise bias correction). More comprehensive experiments considering different indices and spanning more bias correction techniques could be easily undertaken using the functions here shown (more examples are provided in the auxiliary notebook).

6. Example 2: Working with remote data from the UDG

The Santander User Data Gateway (UDG) is a data service providing harmonized remote access to a number of popular datasets in climate studies (a summary is given in Table 1) which is seamlessly integrated with `climate4R` (see Sec. 3.1). In this section we extend the analysis performed in the previous example building a multi-model ensemble of CORDEX projections for the SU index and assessing the resulting uncertainty.

The UDG service requires (free) registration to accept the data policies of the different data providers (`http://www.meteo.unican.es/udg-wiki`). Prior to data access, authentication with valid UDG credentials is required for the current R session in order to access the UDG. Once a valid user name and password have been issued, the authentication can be done in one step within the R session using the `loginUDG` function from `loadeR`:

```
> library(loadeR)
> loginUDG("userUDG", "pswrdUDG")
# Setting credentials...
# Success!
# Go to <http://www.meteo.unican.es/udg-tap/home>
# for details on your authorized groups and datasets
```

It must be noted that it is insecure and in general not advisable to pass the user name and password in plain text within the scripts, although here it is shown this way for illustration purposes. Mechanisms exist in R to ensure a secure transfer of personal data and to avoid revealing personal passwords when sharing code (see e.g. `https://cran.r-project.org/web/packages/httr/vignettes/secrets.html`).

The function `UDG.datasets()` prints a list of the UDG datasets readily available from `climate4R` showing the `name`, `type` (i.e. observation, reanalysis or projection) and `URL`. The harmonization capability for all these datasets is given by the predefined dictionaries included in `loadeR`. The use of these internal dictionaries is activated by default when using the `name` of the target dataset as an entry for the argument `dataset` in `loadGridData`, instead of the full URL. In the following example, we use this option to load CORDEX data, thus, unlike in Example 1 (Sec. 5), there is no need for posterior conversion to the `climate4R` standard naming and units.

For a lighter computational and memory demand, here we restrict the analysis to the Iberian Peninsula (arbitrary spatial domains can be indicated by changing the `lonLim` and `latLim` argument values) and use the 0.44° regular grid (note that the 0.11° simulations are also available at UDG). When listing the available datasets, pattern matching can be used to locate datasets with particular characteristics through the optional argument `pattern`:

```
> mod <- UDG.datasets(pattern = "CORDEX-EUR44.*hist")
> mod$name
#[1] CORDEX-EUR44_ICHEC-EC-EARTH_r12i1p1_RCA4_v1_hist
#[2] CORDEX-EUR44_CERFACS-CNRM-CM5_r1i1p1_RCA4_v1_hist
#[3] CORDEX-EUR44_ICHEC-EC-EARTH_r1i1p1_RACMO22E_v1_hist
#[4] CORDEX-EUR44_ICHEC-EC-EARTH_r3i1p1_HIRHAM5_v1_hist
#[5] CORDEX-EUR44_IPSL-CM5A-MR_r1i1p1_RCA4_v1_hist
#[6] CORDEX-EUR44_MOHC-HadGEM2-ES_r1i1p1_RCA4_v1_hist
...
```

A multi-model ensemble (e.g. the first 6 models in this example) can be ac-

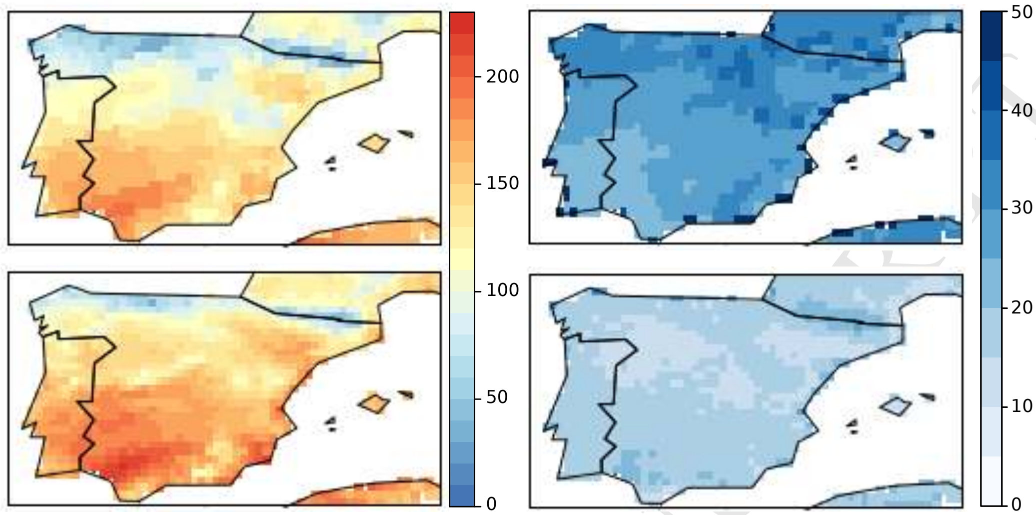

Figure 6: Summer days in Iberia for the future period 2071-2100 computed from the original RCM daily maximum temperature data (above), and daily maximum temperature bias corrected data using E-OBS (below). The left column shows the ensemble mean, whereas the right column shows the ensemble standard deviation (uncertainty).

cessed using a loop on the target datasets (`lapply` in this example):

```
> ensemble.h <- mod$name[1:6]
> TXh.list <- lapply(ensemble.h, function(x) {
                 loadGridData(dataset = x,
                              var = "tasmax",
                              season = 1:12,
                              lonLim = c(-10, 5),
                              latLim = c(36, 44),
                              years = 1971:2000)
              })
```

The six model outputs are next regridded onto the E-OBS grid (the step is detailed in the auxiliary notebook) and the multi-model ensemble is constructed with function `bindGrid`.

```
> TXh.ens <- bindGrid(TXh.list, dimension = "member")
> str(TXh.ens)
```

Note that the new ensemble data structure contains the additional dimension `member`, that includes the six members composing the multi-model, as described in Sec. 2. The same process is followed to obtain the RCP 8.5 future ensemble (`TXf.ens`, see the auxiliary notebook). As a result of arranging all the ensemble members within the same structure, SU index calculation can be performed for the whole ensemble in a single line of code. Additionally, the `member` dimension can be directly aggregated to calculate the ensemble mean and deviation (Fig. 6(top)).

```
> SUf.ens <- climdexGrid(TXf.ens, index.code = "SU")
> SUf.ens.m <- aggregateGrid(SUf.ens,
                        aggr.mem = list(FUN = mean))
> SUf.ens.sd <- aggregateGrid(SUf.ens,
                        aggr.mem = list(FUN = sd))
> # Generates Figure 6 (top):
> spatialPlot(climatology(SUf.ens.m))
> spatialPlot(climatology(SUf.ens.sd))
```

Bias correction (empirical quantile mapping in this example, `method = "eqm"`) is performed similarly, with the possibility to include further arguments (`join.members`) to control how the members are treated within the bias correction step. By default, each member is corrected separately:

```
TXf.ens.bc <- biasCorrection(y = TX,
                             x = TXh.ens,
                             newdata = TXf.ens,
                             window = c(30, 7),
                             method = "eqm")
```

The SU ensemble mean projection and the corresponding uncertainty (as characterized by the standard deviation of the multi-model) can be directly obtained for the bias-corrected data by repeating the above code producing the top panels of Fig. 6, but using the bias-corrected ensemble `TXf.ens.bc` instead of `TXf.ens`, as shown in the two bottom panels of Fig. 6. Finally, the resulting time series for the target location (Zaragoza) are shown in Fig. 7, where the uncertainty of the ensemble is depicted by shaded areas representing the multi-model range (see the auxiliary notebook for the full code).

These results show that a large reduction of the uncertainty is achieved for SU projections after correcting the bias of the original maximum temperature data, highlighting the need for bias-corrected data prior to index calculation. As SU is based on an absolute threshold (25°C), the biases of the different ensemble members largely affect the threshold exceedances, as shown in Figure 8 (see the code in the auxiliary notebook). However, these results might be different for relative (e.g. percentile-based) threshold indices that do not make use of absolute values. Unlike SU, an example for the ETCCDI index CDD (consecutive dry days) is provided in the auxiliary notebook, yielding no significant uncertainty reduction after bias correction.

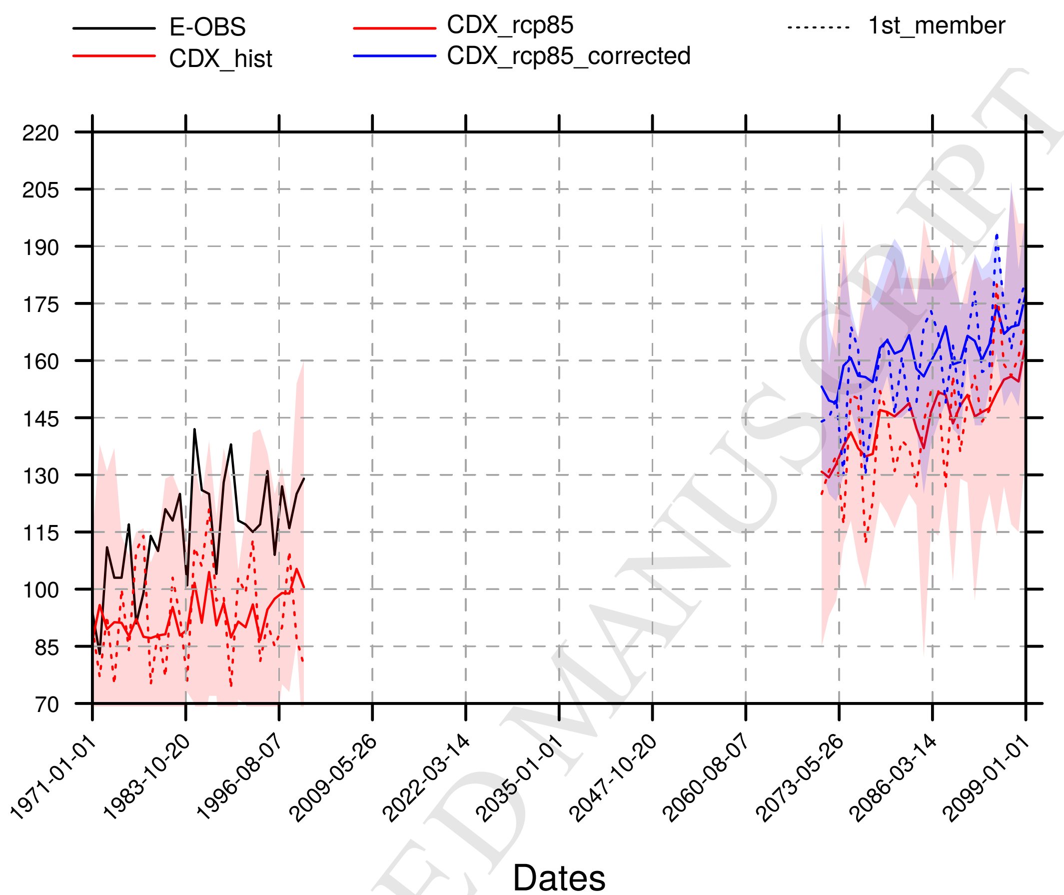

Figure 7: Annual summer days time series for a single gridbox (the one closest to Zaragoza, in the Ebro valley, Spain) computed from (red) the original RCM daily maximum temperature data, and (blue) daily maximum temperature bias corrected data using E-OBS (black). When it comes to CORDEX data, continuous lines correspond to the ensemble mean and the shadowed area to the range (uncertainty). Dashed lines correspond to the 1st member of the ensemble, the same as the one used in Sec. 5 (see Fig. 4).

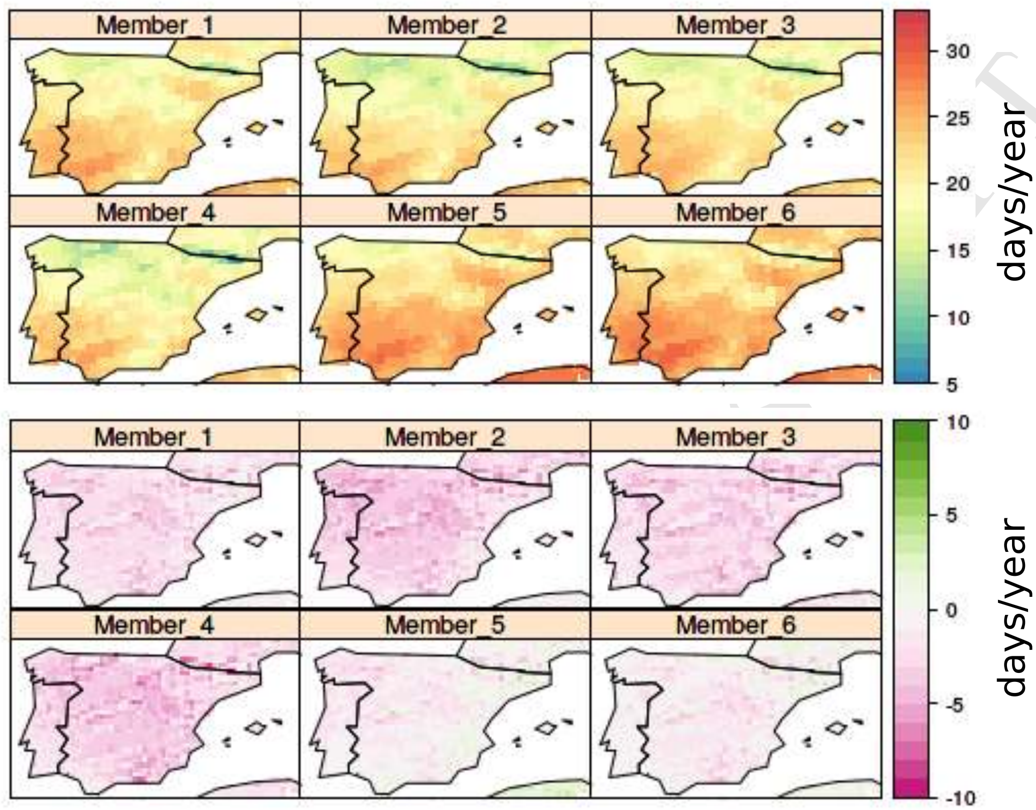

Figure 8: (Top) Maximum temperature in Iberia for the future period 2071-2100 (RCP8.5 scenario) for six CORDEX models. (Bottom) Bias of the RCMs (historical scenario w.r.t. E-OBS for the period 1971-2000).

7. Conclusions

This paper introduces the `climate4R` framework for accessing and post-processing climate data within the R computing environment, and describes its main components (data services, core packages and external packages) and functionalities, including two practical illustrative case studies that showcase its main functionalities. The first example describes the application to calculate and bias-correct future projections of a standard ETCCDI climate index (summer days) for a Southern European domain from locally stored CORDEX data. The second example illustrates an extended case study using remote data (from the Santander UDG) to construct an ensemble of future regional climate projections for different climate indices and to analyze the sensitivity of the results (including the potential reduction of uncertainty after bias correction). Moreover, a companion notebook allows the full reproducibility of the examples (`https://github.com/SantanderMetGroup/notebooks`).

Throughout these examples it has been shown how the different tools available in the `climate4R` framework allow for: 1) an easy harmonized access of user-defined slices from complex datasets —either locally or remotely via OPeNDAP—, 2) flexible data handling, 3) quick and powerful visualization capabilities and 4) straightforward application of a wide range of bias correction methods, providing an intuitive interface for undertaking many different climate data operations usually required by the climate VIA community, and easing the performance of complex research experiments and their end-to-end reproducibility.

Acknowledgements

We acknowledge the E-OBS dataset and the data providers in the ECA&D project, the THREDDS Data Server (TDS) software developed by UCAR/Unidata (`http://doi.org/10.5065/D6N014KG`), and the Earth System Grid Federation (ESGF) infrastructure. Authors are grateful to the modelling groups from the CORDEX and CMIP5 initiatives and the R developers for providing free datasets and software facilitating open science. This work has been funded by the Spanish R+D Program of the Ministry of Economy and Competitiveness, through grants MULTI-SDM (CGL2015-66583-R) and INSIGNIA (CGL2016-79210-R), co-funded by ERDF/FEDER. We would like to thank the two anonymous reviewers for their valuable suggestions and comments.

Software and data availability

- All data used in this paper is publicly available (details are provided in Sections 3, 5 and 6).

- climate4R packages used in this paper are the following:

  'loadeR' (version 1.4.6)
  'transformeR' (version 1.4.4)
  'downscaleR' (version 3.0.3)
  'visualizeR' (version 1.2.2)
  'climate4R.climdex' (version 0.1.4)

- Developers in alphabetical order: J. Baño-Medina, J. Bedia, E. Cimadevilla, A.S. Cofiño, J. Fernández, M. D. Frías, J. M. Gutiérrez, S. Herrera, M. Iturbide, R. Manzanas, D. San-Martín.

- Website: https://github.com/SantanderMetGroup.

- Hardware requirement: General-purpose computer.

- Programming language: R.

- Software requirement: R version 3.5.1 or later.

- Installation code:

```
> library(devtools)
> install_github(c(
"SantanderMetGroup/loadeR.java",
    "SantanderMetGroup/loadeR",
    "SantanderMetGroup/transformeR",
    "SantanderMetGroup/visualizeR",
    "SantanderMetGroup/downscaleR",
    "SantanderMetGroup/climate4R.climdex")
```

Licensing

This software is made freely available under the terms and conditions of the GNU General Public License Version 3.

Appendix A. Downloading data through ESGF

Earth System Grid Federation (ESGF, `https://esgf.llnl.gov/mission.html`) is a worldwide distributed infrastructure for the management and access to the climate data produced in different international initiatives as the different phases of the Coupled Model Intercomparison Project (CMIP) or the Coordinated Regional Climate Downscaling Experiment (CORDEX). ESGF nodes (`https://esgf.llnl.gov/nodes.html`) are the access point to search, explore and download this large amount of data independently on the server in which they are located. In spite of the common access, in order to download the data several previous steps should be made, introducing some difficulties in the process. First, the user should make the registration and obtain the corresponding ESGF account identified by the user's "OpenID" (`https://en.wikipedia.org/wiki/OpenID`) and password. Second, the user should enrol in the groups in which the user is interested (e.g. CMIP5, CORDEX, etc.). Without this step, the user can explore the available data, but can not download it. After data search, the user can add the selected datasets to its Data Cart which can be directly downloaded, dataset by dataset, using her/his OpenId. Alternatively, several shell scripts (e.g. `wget-YYYYMMDDHHMMSS.sh`) can be generated to download the selected dataset using the terminal. To use these scripts the user should have the ESGF-Credentials installed in its home (see e.g. `https://meteo.unican.es/trac/wiki/ESGFGetCredentials` or `https://github.com/ESGF/esgf-getcert` for more details). However, note that on the one hand, the credentials will be valid for just 72 hours and, on the other hand, the scripts can not be modified or adapted to download other datasets. To execute the script, the user can use a BASH shell code similar to the next:

```
DIR=~/.esg
USR=https://esgf-node/esgf-idp/openid/userName
PASS=userPassword
# Retrieve the credentials
export PATH=/root/java/oracle/jdk1.7.0_79/bin:$PATH
java -jar ./getESGFCredentials-0.1.4.jar --openid
    $USR --password $PASS --writeall --output $DIR
unset X509_USER_PROXY
# Executing the script in the terminal:
bash wget-YYYYMMDDhhmmss.sh
# Executing the script in a PBS queue
qsub -d $PWD -V wget-YYYYMMDDhhmmss.sh
```

Baker, M., 2016. 1,500 scientists lift the lid on reproducibility. Nature News 533, 452. doi:10.1038/533452a.

Bedia, J., Golding, N., Casanueva, A., Iturbide, M., Buontempo, C., Gutiérrez, J.M., 2018a. Seasonal predictions of Fire Weather Index: Paving the way for their operational applicability in Mediterranean Europe. Climate Services 9, 101–110. doi:10.1016/j.cliser.2017.04.001.

Bedia, J., Herrera, S., 2018. convertR: A climate4R Package for Unit Conversion and Variable Derivation. URL: https://github.com/SantanderMetGroup/convertR. R package version 0.1.3.

Bedia, J., Herrera, S., Iturbide, M., 2018b. loadeR: The generic data loading package of the climate4R Bundle. URL: https://github.com/SantanderMetGroup/loadeR/wiki. R package version 1.4.6.

Bedia, J., Iturbide, M., Baño-Medina, J., Herrera, S., Manzanas, R., Gutiérrez, J.M., 2017. downscaleR: An R Package for Statistical Downscaling. URL: http://github.com/SantanderMetGroup/downscaleR/wiki. R package version 3.0.3.

Bedia, J., Iturbide, M., Herrera, S., Baño-Medina, J., 2018c. transformeR: An R package for climate data manipulation and transformation. URL: http://github.com/SantanderMetGroup/transformeR/wiki. R package version 1.4.4.

Bivand, R.S., Pebesma, E., Gomez-Rubio, V., 2013. Applied spatial data analysis with R, Second edition. Springer, NY. URL: http://www.asdar-book.org.

Bronaugh, D., 2015. climdex.pcic: PCIC Implementation of Climdex Routines. URL: https://CRAN.R-project.org/package=climdex.pcic. R package version 1.1-6.

Casanueva, A., Bedia, J., Herrera, S., Fernández, J., Gutiérrez, J.M., 2018. Direct and component-wise bias correction of multi-variate climate indices: the percentile adjustment function diagnostic tool. Climatic Change 147, 411–425. doi:10.1007/s10584-018-2167-5.

Casanueva, A., Frías, M.D., Herrera, S., San-Martín, D., Zaninovic, K., Gutiérrez, J.M., 2014. Statistical downscaling of climate impact indices: testing the direct approach. Climatic Change 127, 547–560. doi:10.1007/s10584-014-1270-5.

Challinor, A.J., Muller, C., Asseng, S., Deva, C., Nicklin, K.J., Wallach, D., Vanuytrecht, E., Whitfield, S., Ramirez-Villegas, J., Koehler, A.K., 2018. Improving the use of crop models for risk assessment and climate change adaptation. Agricultural Systems 159, 296 – 306. doi:10.1016/j.agsy.2017.07.010.

Cofiño, A., Bedia, J., Iturbide, M., Vega, M., Herrera, S., Fernández, J., Frías, M.D., Manzanas, R., Gutiérrez, J.M., 2018. The ECOMS User Data Gateway: Towards seasonal forecast data provision and research reproducibility in the era of Climate Services. Climate Services 9, 33–43. doi:10.1016/j.cliser.2017.07.001.

Dee D. P., Uppala S. M., Simmons A. J., Berrisford P., Poli P., Kobayashi S., Andrae U., Balmaseda M. A., Balsamo G., Bauer P., Bechtold P., Beljaars A.

C. M., van de Berg L., Bidlot J., Bormann N., Delsol C., Dragani R., Fuentes M., Geer A. J., Haimberger L., Healy S. B., Hersbach H., Hólm E. V., Isaksen L., Kållberg P., Köhler M., Matricardi M., McNally A. P., Monge-Sanz B. M., Morcrette J.-J., Park B.-K., Peubey C., de Rosnay P., Tavolato C., Thépaut J.-N., Vitart F., 2011. The ERA-Interim reanalysis: configuration and performance of the data assimilation system. Q J R Meteorol Soc 137, 553–597. doi:10.1002/qj.828.

Déqué, M., 2007. Frequency of precipitation and temperature extremes over France in an anthropogenic scenario: Model results and statistical correction according to observed values. Global and Planetary Change 57, 16–26. doi:10.1016/j.gloplacha.2006.11.030.

Ewert, F., Rotter, R., Bindi, M., Webber, H., Trnka, M., Kersebaum, K., Olesen, J., van Ittersum, M., Janssen, S., Rivington, M., Semenov, M., Wallach, D., Porter, J., Stewart, D., Verhagen, J., Gaiser, T., Palosuo, T., Tao, F., Nendel, C., Roggero, P., Bartosova, L., Asseng, S., 2015. Crop modelling for integrated assessment of risk to food production from climate change. Environmental Modelling & Software 72, 287 – 303. doi:10.1016/j.envsoft.2014.12.003.

Frías, M.D., Iturbide, M., Manzanas, R., Bedia, J., Fernández, J., Herrera, S., Cofiño, A.S., Gutiérrez, J.M., 2018. An R package to visualize and communicate uncertainty in seasonal climate prediction. Environmental Modelling & Software 99, 101–110. doi:10.1016/j.envsoft.2017.09.008.

Giorgi, F., Gutowski, W.J., 2015. Regional dynamical downscaling and the

CORDEX initiative. Annual Review of Environment and Resources 40, 467–490. doi:10.1146/annurev-environ-102014-021217.

Gutiérrez, J.M., Maraun, D. abd Widmann, M., Huth, R., Hertig, E., Benestad, R., Roessler, R., Wibig, T., Wilcke, R., Kotlarski, S., San-Martín, D., Herrera, S., Bedia, J., Casanueva, A., Manzanas, R., Iturbide, M., Vrac, M., 2018. An intercomparison of a large ensemble of statistical downscaling methods over Europe: Results from the VALUE perfect predictor cross-validation experiment. International Journal of Climatology. doi:10.1002/joc.5462.

Haylock M. R., Hofstra N., Klein Tank A. M. G., Klok E. J., Jones P. D., New M., 2008. A European daily high-resolution gridded data set of surface temperature and precipitation for 1950–2006. Journal of Geophysical Research: Atmospheres 113. doi:10.1029/2008JD010201.

Herrera, S., Fernández, J., Gutiérrez, J.M., 2016. Update of the Spain02 gridded observational dataset for EURO-CORDEX evaluation: assessing the effect of the interpolation methodology. International Journal of Climatology 36, 900–908. doi:10.1002/joc.4391.

Herrera, S., Gutiérrez, J.M., Ancell, R., Pons, M.R., Frías, M.D., Fernández, J., 2012. Development and analysis of a 50-year high-resolution daily gridded precipitation dataset over Spain (Spain02). International Journal of Climatology 32, 74–85. doi:10.1002/joc.2256.

Hiebert, J., 2016. udunits2: Udunits-2 Bindings for R. URL: https://CRAN.R-project.org/package=udunits2. R package version 0.13.

Hijmans, R.J., 2017. raster: Geographic Data Analysis and Modeling. URL: https://CRAN.R-project.org/package=raster. R package version 2.6-7.

Iturbide, M., Bedia, J., Gutiérrez, J.M., 2018. Tackling Uncertainties of Species Distribution Model Projections with Package mopa. The R Journal 10, 122–139.

Jacob, D., Petersen, J., Eggert, B., Alias, A., Christensen, O.B., Bouwer, L.M., Braun, A., Colette, A., Déqué, M., Georgievski, G., Georgopoulou, E., Gobiet, A., Menut, L., Nikulin, G., Haensler, A., Hempelmann, N., Jones, C., Keuler, K., Kovats, S., Kröner, N., Kotlarski, S., Kriegsmann, A., Martin, E., van Meijgaard, E., Moseley, C., Pfeifer, S., Preuschmann, S., Radermacher, C., Radtke, K., Rechid, D., Rounsevell, M., Samuelsson, P., Somot, S., Soussana, J.F., Teichmann, C., Valentini, R., Vautard, R., Weber, B., Yiou, P., 2014. EURO-CORDEX: new high-resolution climate change projections for european impact research. Regional Environmental Change 14, 563–578. doi:10.1007/s10113-013-0499-2.

Karl, T.R., Nicholls, N., Ghazi, A., 1999. CLIVAR/GCOS/WMO Workshop On Indices And Indicators For Climate Extremes. Workshop Summary. Climatic Change 42, 3–7. doi:10.1007/978-94-015-9265-9_2.

Kobayashi, S., Ota, Y., Harada, Y., Ebita, A., Moriya, M., Onoda, H., Onogi, K., Kamahori, H., Kobayashi, C., Endo, H., Miyaoka, K., Takahashi, K., 2015. The JRA-55 reanalysis: General specifications and basic characteristics. Journal of the Meteorological Society of Japan. Ser. II 93, 5–48. doi:10.2151/jmsj.2015-001.

Lange, S., 2016. EartH2Observe, WFDEI and ERA-Interim data Merged and Bias-corrected for ISIMIP (EWEMBI). GFZ Data Services. doi:10.5880/pik.2016.004.

Lemos, M.C., Kirchhoff, C.J., Ramprasad, V., 2012. Narrowing the climate information usability gap. Nature Climate Change 2, 789–794. doi:10.1038/nclimate1614.

Manzanas, R., 2016. Statistical downscaling of precipitation in seasonal forecasting: Advantages and limitations of different approaches. Ph.D. thesis. University of Cantabria (PhD Programme in Science, Technology and Computation). URL: http://hdl.handle.net/10803/398783.

Manzanas, R., Gutiérrez, J.M., Fernández, J., van Meijgaard, E., Calmanti, S., Magariño, M.E., Cofiño, A.S., Herrera, S., 2017a. Dynamical and statistical downscaling of seasonal temperature forecasts in Europe: Added value for user applications. Climate Services 9, 44–56. doi:10.1016/j.cliser.2017.06.004.

Manzanas, R., Lucero, A., Weisheimer, A., Gutiérrez, J.M., 2017b. Can bias correction and statistical downscaling methods improve the skill of seasonal precipitation forecasts? Climate Dynamics 50, 1161–1176. doi:10.1007/s00382-017-3668-z.

Maraun, D., Shepherd, T.G., Widmann, M., Zappa, G., Walton, D., Gutiérrez, J.M., Hagemann, S., Richter, I., Soares, P.M.M., Hall, A., Mearns, L.O., 2017. Towards process-informed bias correction of climate change simulations. Nature Climate Change 7, 764–773. doi:10.1038/nclimate3418.

Maraun, D., Widmann, M., Gutiérrez, J.M., Kotlarski, S., Chandler, R., Hertig, E., Wibig, J., Huth, R., R.A.I., W., 2015. VALUE: A framework to validate downscaling approaches for climate change studies. Earth's Future 3, 1–14. doi:10.1002/2014EF000259.

Michna, P., 2014. RNetCDF: R Interface to NetCDF Datasets. URL: https://CRAN.R-project.org/package=RNetCDF. R package version 1.6.3-1.

Moreau, L., Groth, P., Cheney, J., Lebo, T., Miles, S., 2015. The rationale of PROV. Web Semantics: Science, Services and Agents on the World Wide Web 35, 235–257. doi:10.1016/j.websem.2015.04.001.

Nikulin, G., Jones, C., Giorgi, F., Asrar, G., Büchner, M., Cerezo-Mota, R., Christensen, O.B., Déqué, M., Fernandez, J., Hänsler, A., van Meijgaard, E., Samuelsson, P., Sylla, M.B., Sushama, L., 2012. Precipitation Climatology in an Ensemble of CORDEX-Africa Regional Climate Simulations. Journal of Climate 25, 6057–6078. doi:10.1175/JCLI-D-11-00375.1.

Pierce, D., 2017. ncdf4: Interface to Unidata netCDF (Version 4 or Earlier) Format Data Files. URL: https://CRAN.R-project.org/package=ncdf4. R package version 1.16.

PROV Working Group, 2013. PROV-O: The PROV Ontology. W3C Recommendation. URL: https://www.w3.org/TR/2013/REC-prov-o-20130430/.

R Core Team, 2017. R: A Language and Environment for Statistical Computing. Technical Report. R Foundation for Statistical Computing. Vienna, Austria. URL: https://www.R-project.org.

Sarkar, D., 2008. Lattice: Multivariate Data Visualization with R. Springer, New York. URL: http://lmdvr.r-forge.r-project.org. iSBN 978-0-387-75968-5.

Taylor, K.E., Stouffer, R.J., Meehl, G.A., 2011a. An overview of CMIP5 and the experiment design. Bull. Amer. Meteor. Soc. 93, 485–498. doi:10.1175/BAMS-D-11-00094.1.

Taylor, K.E., Stouffer, R.J., Meehl, G.A., 2011b. An Overview of CMIP5 and the Experiment Design. Bulletin of the American Meteorological Society 93, 485–498. doi:10.1175/BAMS-D-11-00094.1.

Themeßl, M.J., Gobiet, A., Heinrich, G., 2012. Empirical-statistical downscaling and error correction of regional climate models and its impact on the climate change signal. Climatic Change 112, 449–468. doi:10.1007/s10584-011-0224-4.

Turco, M., Cánovas, J.R., Bedia, J., Jerez, S., Montávez, J., Llasat, M.C., Provenzale, A., 2018. Exacerbated fires in mediterranean europe due to anthropogenic warming projected with non-stationary climate-fire models. Nature Communications (in press).

Unidata, 2006. Thredds data server. doi:10.5065/D6N014KG.

Unidata, 2017. UDUNITS-2. UCAR/Unidata. Boulder, CO. URL: https://www.unidata.ucar.edu/software/udunits/, doi:https://ezid.cdlib.org/id/doi:10.5065/D6KD1WN0. version 2.2.20-1.

Urbanek, S., 2016. rJava: Low-Level R to Java Interface. URL: https://CRAN.R-project.org/package=rJava. R package version 0.9-8.

W3C, 2004. Resource Description Framework (RDF): Concepts and Abstract Syntax. URL: https://www.w3.org/TR/2004/REC-rdf-concepts-20040210/.

Walsh, J.E., Bhatt, U.S., Littell, J.S., Leonawicz, M., Lindgren, M., Kurkowski, T.A., Bieniek, P.A., Thoman, R., Gray, S., Rupp, T.S., 2018. Downscaling of climate model output for alaskan stakeholders. Environmental Modelling & Software (in press). doi:10.1016/j.envsoft.2018.03.021.

Wang, J., Nathan, R., Horne, A., Peel, M.C., Wei, Y., Langford, J., 2017. Evaluating four downscaling methods for assessment of climate change impact on ecological indicators. Environmental Modelling & Software 96, 68–82. doi:10.1016/j.envsoft.2017.06.016.

Weedon, G.P., Balsamo, G., Bellouin, N., Gomes, S., Best, M.J., Viterbo, P., 2014. The WFDEI meteorological forcing data set: WATCH Forcing Data methodology applied to ERA-Interim reanalysis data. Water Resources Research 50, 7505–7514. doi:10.1002/2014WR015638.

Williams, D.N., Balaji, V., Cinquini, L., Denvil, S., Duffy, D., Evans, B., Ferraro, R., Hansen, R., Lautenschlager, M., Trenham, C., 2015. A global repository for planet-sized experiments and observations. Bull. Amer. Meteor. Soc. 97, 803–816. doi:10.1175/BAMS-D-15-00132.1.

Highlights

- climate4R is an R-based framework for accessing and post-processing climate data
- climate4R builds on NetCDF-Java and allows accessing local and remote (OPeNDAP) data
- The UDG is a climate service envisaged as a data access layer for climate4R
- climate4R provides access to widely used and harmonized public datasets via UDG
- climate4R favours end-to-end reproducibility of sectoral impact studies