Technical Notes
Santander Meteorology Group (CSIC-UC)
http://www.meteo.unican.es
SMG:01.2014

The ECOMS-UDG R-Package for seasonal climate forecasting data access

J. Bedia[1], M.E. Magariño[2], S. Herrera[2], R. Manzanas[1], J. Fernández[2], A.S. Cofiño[2] & J.M. Gutiérrez[1]
[1] *Instituto de Física de Cantabria, CSIC-Universidad de Cantabria, Spain.*
[2] *Dpto. de Matemática Aplicada y C.C. Universidad de Cantabria, Spain*
correspondence: bediaj@unican.es

version:v2.0-0–*16 Jun 2014*

Abstract

This document describes the `ecomsUDG.Raccess` R package, envisaged as a user-friendly R-based interface for remotely accessing different climate datasets stored at the ECOMS User Data Gateway (ECOMS-UDG), including the NCEP/CFSv2 and ECMWF/System4 hindcasts.

wiki: http://meteo.unican.es/ecoms-udg/RPackage
gitHub: https://github.com/SantanderMetGroup/ecomsUDG.Raccess

1 Introduction

The European Climate Observations, Modelling and Services initiative (ECOMS) coordinates the activities of three ongoing European projects: EUPORIAS, SPECS and NACLIM. Different activities carried out in these projects require seasonal forecasts from state-or-the-art forecasting systems (e.g. NCEP/CFSv2 or ECMWF/System4) and observational and reanalysis data for a reduced number of variables. This information can be obtained directly from the data providers, but the resulting formats, aggregations and vocabularies may not be homogeneous across datasets, thus requiring some post processing. Moreover, different data policies hold for the various datasets —which are freely available only in some cases— and therefore data access may not be straightforward. Thus, obtaining seasonal climate forecast data and other climate information and ensuring its homogenity across datasets and variables is typically a time consuming task.

The ECOMS User Data Gateway (ECOMS-UDG) has been developed by the Santander MetGroup in order to facilitate seasonal forecasting and other climate data access to end users. The needed variables have been downloaded from data providers and locally stored in a THREDDS data server implementing fine-grained user authorization and using remote data access protocols with data subsetting capabilities. Thus, users can efficiently retrieve the subsets best suited to their particular research aims (for particular regions, periods and/or ensemble members) from a large volumen of information.

The `ecomsUDG.Raccess` R package constitutes a user-friendly interface to the data stored at ECOMS-UDG, ensuring the consistency and homogeneity of the returned variables across different datasets. There is one single function –`loadECOMS`– achieving all data subsetting and download, which has a few, simple arguments for subset specification.

A comprehensive description of the ECOMS-UDG is available (and periodically updated) in the wiki http://meteo.unican.es/ecoms-udg including:

- The list of current datasets, corresponding to different reforecasts from state-of-the-art forecasting systems (e.g. CFSv2 or System4) for several decades, allowing for statistical analysis.
- The list of current variables required by ECOMS users, which include both typical variables at surface for impact studies, but also at pressure levels for statistical downscaling purposes.

- A description of the alternative tools which can be used to access this information in a user-friendly form, including the `ecomsUDG.Raccess` R package, which relies on the powerful capabilities of the Unidata's netCDF Java library.

2 ECOMS-UDG Registration

As different data policies and terms of use apply to the datasets stored at the ECOMS-UDG, a fine-grained user authorization scheme has been implemented, based on different access roles which are provided under request. For instance, while the NCEP/CFSv2 reforecast (CFSRR) dataset is publicly available, the ECMWF/System4 reforecast is restricted to ECOMS partners. Thus, the role "cfsrr" is provided to all potential users, whereas the role "system4" is limited to verified ECOMS partners. Similarly, the observational dataset WFDEI (WATCH Forcing Dataset based on ERA-Interim) has a public role so access authorization is also granted.

Registration and role request in the ECOMS-UDG can be done at the THREDDS Administration Panel (TAP, `http://meteo.unican.es/tap`). The applicants for a particular role must accept the terms of use and conditions of the corresponding datasets. More information on the registration procedure is given in the wiki-registration section.

Since all users can request the role "cfsrr" and "wfdei" after registration, the examples in this document are illustrated using these datasets, although the same examples will work for "System4" for those authorized users.

2.1 User authentication

Once a valid user name (e.g. "myUser") and password (e.g. "myPassword") are issued, HTTP authentication is directly achieved within a R session using the function `loginECOMS_UDG`:

```
> library(ecomsUDG.Raccess)
> loginECOMS_UDG(username = "myUser", password = "myPassword")
```

In case the connection is done via a proxy server, the name of the server and the proxy port must be provided filling the corresponding arguments. Type `?loginECOMS_UDG` for details.

3 Accessing Data

3.1 Data homogeneization

The diverse naming and storage conventions often applied by the various modelling centres requires previous dataset homogeneization. The `ecomsUDG.Raccess` package pursues this aim by defining a common *vocabulary* to be used in R sessions. The variables of each particular dataset are translated —and transformed when necessary— to the common vocabulary by means of a *dictionary*, which contains the necessary information to unequivocally define the time aggregation of the data and the conversion operations needed to get the standard units. Dictionaries are built-in in the `ecomsUDG.Raccess` package. The latest version of the dictionaries can be also checked-out in the gitHub repository.

The vocabulary is included as a dataset in the `ecomsUDG.Raccess` package. So far, only the currently available variables are included, although the vocabulary is subject to continuous update along with the ECOMS-UDG datasets (see the wiki-variables section for more details). Thus, the vocabulary defines the "standard variables" in R:

```
> data(vocabulary)
> print(vocabulary)
   identifier                          standard_name           units
1         tas                    2-meter temperature degrees Celsius
2      tasmax                maximum 2-m temperature degrees Celsius
3      tasmin                minimum 2-m temperature degrees Celsius
4          tp             total precipitation amount              mm
5         psl               air pressure at sea level              Pa
6        rsds surface downwelling shortwave radiation           W.m-2
7        rlds  surface downwelling longwave radiation           W.m-2
8         uas              eastward near-surface wind           m.s-1
9         vas             northward near-surface wind           m.s-1
10       tdps           2-meter dewpoint temperature degrees Celsius
11       snld                              snow depth  mm (water eq.)
12       huss              2-meter specific humidity         kg.kg-1
13       hurs              2-meter relative humidity               %
14         ps                   surface air pressure              Pa
15        wss                 near-surface wind speed           m.s-1
```

In conclusion, the ECOMS-UDG users do not need to worry about the different names and units of the variables across the different datasets, just by introducing the default `identifier` indicated in the vocabulary as input for the argument `var` in the `loadSeasonalForecast` function. More advanced users interested in obtaining the original model variables can retrieve them (although this is not recommended), as explained in the wiki-Rpackage section.

3.2 Data retrieval

A few examples of data retrieval using the `loadECOMS` function are presented below. These simple examples provide the recommended use of the function when working at different spatial scales, from point-scale to continental and global levels. The examples are designed to keep a moderate size (<150 MB) for the output and a reasonable execution time (<10 minutes) for remote data retrieval. However, note that the time largely depends on the characteristics of the internet connection and the ECOMS-UDG traffic load at the moment of accessing the data. Thus, if the data request takes too long, we strongly advice to simplify the requested dataset. Sometimes a slow request is much faster later on, just because of the traffic load just at the moment of requesting access.

3.2.1 Time filtering/aggregation

It is also possible to apply time filtering through the argument `time`. For instance, instead of 6-hourly data, one might be interested in obtaining data for 0, 6, 12 or 18 hours only. In this case, the required time must be specified in the argument `time` by the corrsponding time as a character string (e.g., for data at 12:00, `time = "12"`, at 6:00, `time = "06"` and so on ...). Furthermore, in the case of 6-hourly datasets, it is also possible to compute a daily mean value based on the four instantaneous data per day using `time = "DD"`. Attempts to compute a daily mean from 12-hourly variables will throw an error, while `time = "DD"` for daily variables will be simply ignored and set to its default value (`time = "none"`), meaning that no filtering or aggregation will be performed.

More elaborated worked examples are presented in the wiki-Rpackage section.

EXAMPLE 1 (Point Scale): Seasonal prediction time series for a single point can be accessed by indicating their geographical coordinates in the `lonLim` and `latLim` arguments. Note that the function `loadECOMS` doest not perform on-the-fly spatial interpolation in order to preserve the original data, and therefore, the resulting data corresponds to the closest model gridbox. The following example loads 2 members for the CFSv2 seasonal model of summer (JJA) 2m temperature data at Madrid (Spain, -3.680E, 40.40N). We consider one month lead-time forecasts for the 10-year period 1990-2001. Note that if argument `members` is set to NULL (the default), 16 members would be returned in this case (`?loadECOMS`

for full details). NIn this case we will obtain the mean daily data calculated from the 6-hourly variable using the argument `time = "DD"`.

```
> ex1 <- loadECOMS(dataset = "CFSv2_seasonal_16", var = "tas",
+ members = 1:2, lonLim=-3.7, latLim=40.4, season=6:8, years = 1991:2000,
+ leadMonth=1, time = "DD")
[2014-06-16 17:02:45] Defining homogeneization parameters for variable "tas"
NOTE: daily mean will be calculated from the 6-h instantaneous model output
[2014-06-16 17:02:46] Defining geo-location parameters
[2014-06-16 17:02:46] Defining initialization time parameters
[2014-06-16 17:02:50] Retrieving data subset ...
[2014-06-16 17:08:15] Done
> print(object.size(ex1), units = "Mb")
0.1 Mb
```

Below is an example plot of the daily time series loaded, for the two ensemble members:

```
> plot(ex1$Dates$start, ex1$Data[ ,1], ty = "l", col = "red", ylab = "tas", xlab = "time")
> lines(ex1$Dates$start, ex1$Data[ ,2], ty = "l", col = "blue")
> legend("topleft", ex1$Members, lty = 1, col = c("red","blue"))
> title("t2m JJA series for Madrid")
> mtext(paste(round(ex1$xyCoords$x, 2), "E,", round(ex1$xyCoords$y, 2), "N"))
```

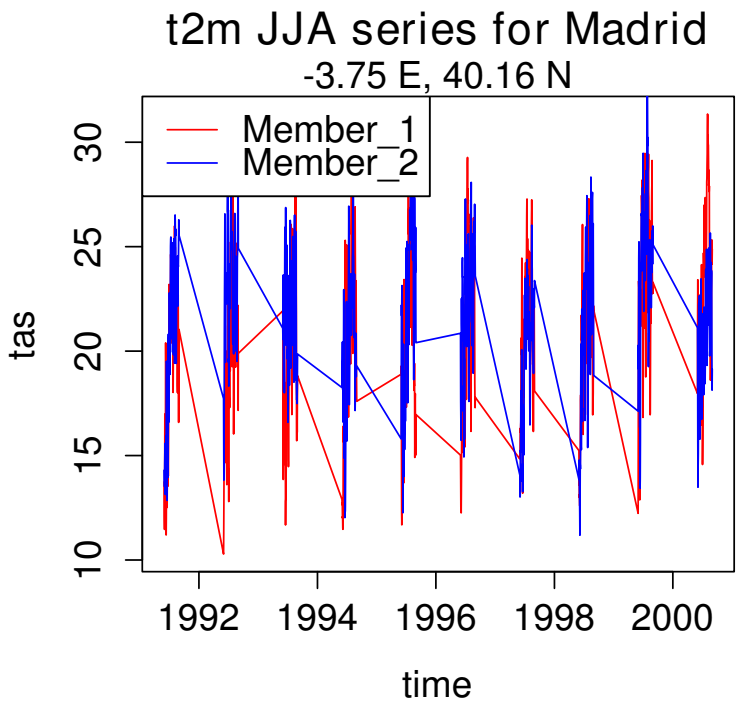

Figure 1: Time series (daily means computed from 6-hourly outputs) from the example 1.

Single point queries allow retrieving long time series for all ensemble members and years without worrying for the memory size of the returned object. However, the execution time grows linearly with the number of years, so it is advisable to access the data decade by decade to avoid long request times for this dataset. Note that the situation is different for each dataset; for instance, some of the System4 variables are defined on a daily basis and, therefore, reasonable execution times are obtained when requesting the whole period (i.e., setting the `years` parameter to NULL). The same occurs if we use the time filtering for retrieving data at particular times in the case of sub-daily variables.

EXAMPLE 2 (Continental Scale): When working with continental spatial domains, it is recommended to consider shorter time periods and/or single members in order to keep a moderate size for the resulting data. In case larger datasets are needed, the job should be divided in different calls to the function in order to avoid running out of memory. The following example loads spring (MAM) precipitation forecasted in January (lead month = 3) for Europe, spanning the 10-year period 2001-2010 and the first two members of the CFSv2 reforecast, as in the point-based example. As an illustrative example of data manipulation, it is also shown how to compute and visualize the mean precipitation.

```
> ex2 <- loadECOMS(dataset = "CFS", var = "tp", members = 1:2, lonLim = c(-15,35),
+ latLim = c(32, 75), season = 3:5, years = 2001:2010, leadMonth = 3)
[2014-06-16 17:17:53] Defining homogeneization parameters for variable "tp"
[2014-06-16 17:17:54] Defining geo-location parameters
[2014-06-16 17:17:54] Defining initialization time parameters
[2014-06-16 17:17:58] Retrieving data subset ...
[2014-06-16 17:24:57] Done
> print(object.size(ex2), units = "Mb")
142.9 Mb
```

Note that now the size of the output is over 140MB, as compared to the point-based example above. Thus at a continental scale we advise to work considering a single member or few members and a decade at a time or less, or, when relevant, apply a time filter/aggregation (see Section 3.2.1) to significantly reduce the amount of data.

Data are returned as a 4D array (the `Data` slot), along with other relevant information, with the dimension names (and order) indicated by the `dimensions` attribute:

```
> str(ex2)
List of 6
 $ Variable           :List of 2
  ..$ varName   : chr "tp"
  ..$ isStandard: logi TRUE
 $ Data               : num [1:54, 1:47, 1:3680, 1:2] 0 0 0 0 0 0 0 0 0 0 ...
  ..- attr(*, "dimensions")= chr [1:4] "lon" "lat" "time" "member"
 $ xyCoords           :List of 3
  ..$ x         : num [1:54] -15 -14.1 -13.1 -12.2 -11.3 ...
  ..$ y         : num [1:47] 31.7 32.6 33.5 34.5 35.4 ...
  ..$ CRS_string: chr "+proj=longlat +ellps=WGS84 +datum=WGS84 +no_defs +towgs84=0,0,0"
 $ Dates              :List of 2
  ..$ start: POSIXlt[1:3680], format: "2001-03-01 00:00:00" "2001-03-01 06:00:00" ...
  ..$ end  : POSIXlt[1:3680], format: "2001-03-01 06:00:00" "2001-03-01 12:00:00" ...
 $ InitializationDates:List of 2
  ..$ Member_1: POSIXlt[1:10], format: "2000-11-12 00:00:00" "2000-11-12 12:00:00" ...
  ..$ Member_2: POSIXlt[1:10], format: "2000-11-12 06:00:00" "2000-11-12 18:00:00" ...
 $ Members            : chr [1:2] "Member_1" "Member_2"
```

Next, the mean MAM precipitation for the domain and time span selected is computed for each member sepparately.

```
> # Spatial mean by members
> member1 <- apply(ex2$Data[,,,1], FUN = mean, MAR = c(1,2))
> member2 <- apply(ex2$Data[,,,2], FUN = mean, MAR = c(1,2))
> # X and Y coordinates
> x <- ex2$xyCoords$x
> y <- ex2$xyCoords$y
> # We use as an example the plotting utils of library \code{fields}:
> library(fields)
> par(mfrow = c(1,2))
> image.plot(x, y, member1, asp=1)
> title("Member 1")
> world(add = TRUE)
```

```
> image.plot(x, y, member2, asp=1)
> title("Member 2")
> world(add = TRUE)
```

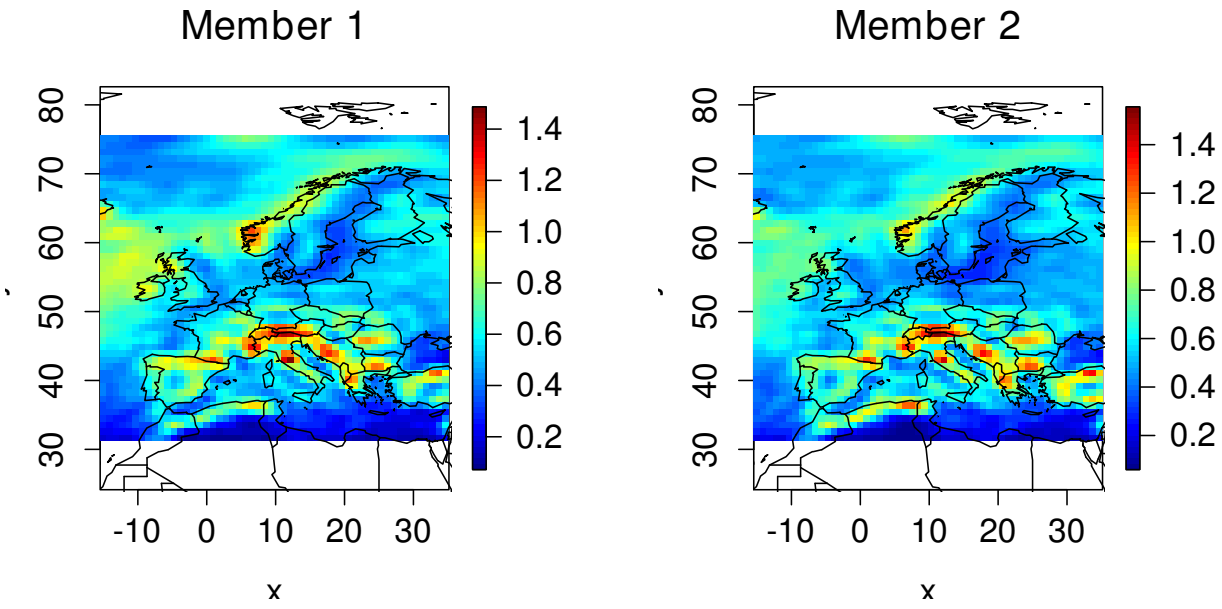

Figure 2: 3-month lead forecast for mean 6-hourly MAM precipitation (mm) of the NCEP's CFSv2 model for Europe considering the first two members and the period 2001-2010.

EXAMPLE 3 (Global Scale): For this example we will load the gridded observational dataset WFDEI. In particular, we will load the daily surface (2m) minimum temperature for all land areas globally in boreal winter (DJF) for the year 2010:

```
> ex3 <- loadECOMS(dataset = "WFDEI", var = "tasmin", lonLim = NULL, latLim = NULL,
+ season = c(12,1,2), years = 2010)
[2014-06-16 18:43:51] Defining homogeneization parameters for variable "tasmin"
[2014-06-16 18:43:52] Defining geo-location parameters
[2014-06-16 18:43:52] Defining time selection parameters
[2014-06-16 18:44:03] Done
```

Note that, unlike in the previous examples, we have omitted the `leadMonth` and `members` arguments, as this is not a forecast dataset, but a gridded observational dataset lacking the initialization and ensemble dimensions.

```
> a <- apply(ex3$Data, FUN = mean, MAR = c(1,2))
> image.plot(ex3$xyCoords$x, ex3$xyCoords$y, a, asp = 1, xlab = "", ylab = "")
> world(add = TRUE)
```

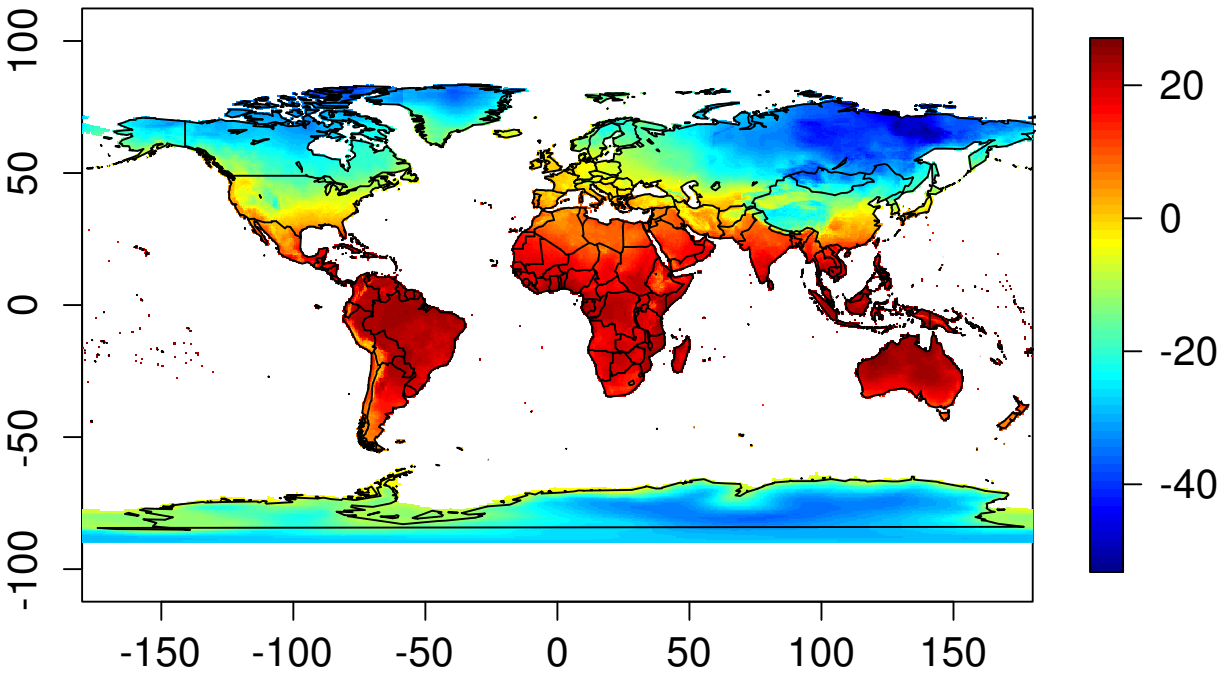

Figure 3: Global mean boreal winter (DJF) surface minimum temperature of the WFDEI dataset of year 2010.