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:v1.0–*15 Feb 2014*

Abstract

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

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) 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 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 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.

Moreover, a number of tools are being develop in order to provide different user-friendly interfaces for accessing the information (R, Python, web portals, ...). In this document we describe one of these tools, the `ecomsUDG.Raccess` R package, envisaged as a user-friendly, R-based interface for remotely subsetting/accessing the datasets stored at the ECOMS-UDG.

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.

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" after registration, the examples in this document are illustrated using this dataset, 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. The latest version of the dictionaries can be 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
```

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 easily, as explained in the wiki-Rpackage section.

3.2 Data retrieval

A few examples of data retrieval using the `loadSeasonalForecast` 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 levels. The examples are designed to keep a moderate size (<100 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. Thus, if the data request takes too long, we strongly advice to simplify the requested dataset.

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 `loadSeasonalForecast` doest not perform 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 (`?loadSeasonalForecast` for full details).

```
> ex1 <- loadSeasonalForecast("CFSv2_seasonal_16", var = "tas", members = 1:2,
+ lonLim=-3.7, latLim=40.4, season=6:8, years = 1991:2000, leadMonth=1)
[2014-02-13 13:18:35] Retrieving data from member 1 out of 2...
[2014-02-13 13:21:28] Retrieving data from member 2 out of 2...
[2014-02-13 13:24:16] Done
> print(object.size(ex1), units = "Mb")
0.3 Mb
```

Note that the time resolution of the CFSv2 reforecast stored in the ECOMS-UDG is 6-hourly data; this information is readily accesible in the corresponding dictionary. Below is an example plot of the 6-hourly time series loaded, for the two ensemble members.

```
> # Generates Fig 1
> plot(ex1$ForecastDates$Start, ex1$MemberData[[1]], ty = "l", col = "red",
+ ylab = "tas", xlab = "time")
> lines(ex1$ForecastDates$Start, ex1$MemberData[[2]], ty = "l", col = "blue")
> legend("topleft", names(ex1$MemberData), lty = 1, col = c("red","blue"))
> title("6-hourly surface (2m) temperature JJA series for Madrid")
```

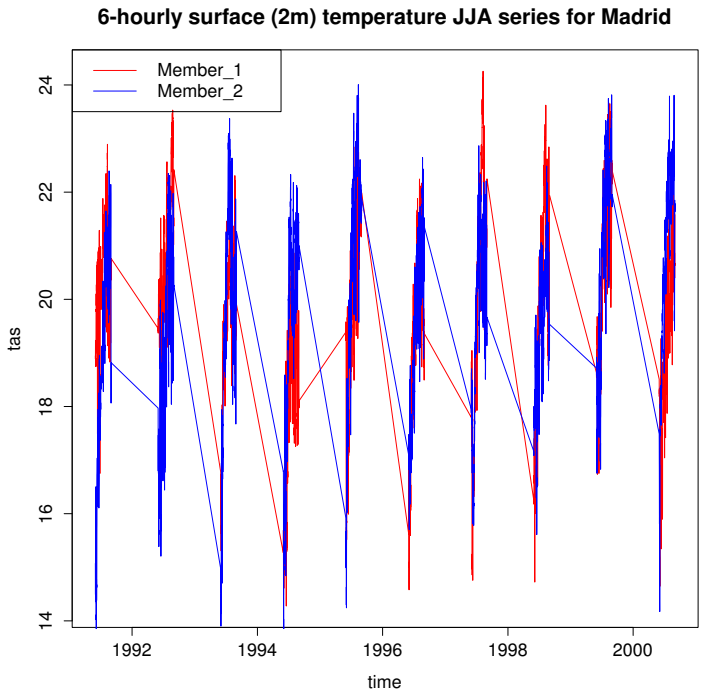

Figure 1: 6-hourly time series representation from the example 1.

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, System4 is 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).

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 (also 6-hourly, as indicated in the dictionary) forecasted in January (lead month = 3) for Europe, spanning also a 10-year period (2001-2010 in this case) 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 mean seasonal precipitation and the 95th percentile of (6-hourly) precipitation. Further examples and alternative ways of handling/plotting seasonal forecast data are provided in the wiki-Rpackage section.

```
> ex2 <- loadSeasonalForecast(dataset = "CFSv2_seasonal_16", var = "tp",
+ members = 1:2, lonLim = c(-15,35), latLim = c(32, 75), season = 3:5,
+ years = 2001:2010, leadMonth = 3)
[2014-02-13 12:32:19] Retrieving data from member 1 out of 2...
[2014-02-13 12:36:04] Retrieving data from member 2 out of 2...
[2014-02-13 12:39:33] Done
> print(object.size(ex2), units = "Mb")
134.2 Mb
```

Note that now the size of the output is over 100MB, as compare to the point-based example above. Thus at a continental scale we advise to work with CFSv2 (6-hourly data) considering a single member

and decade at a time. For System4 (daily) the full information for a single member (all years) could be loaded at a time.

Next, the mean MAM precipitation for the domain and time span selected is computed:

```
> # Time components (e.g.: years) can be directly get from the "POSIXlt" class
> years <- ex2$ForecastDates$Start$year + 1900
> # This function computes the mean seasonal precip
> pr.accum <- function(x) {
+       pr.acc <- matrix(ncol = ncol(x), nrow = length(years))
+       for (i in 1:ncol(x)) {
+             pr.acc[ ,i] <- tapply(x[ ,i], INDEX = years, FUN = sum)
+       }
+       return(colMeans(pr.acc))
+}
> # Mean precipitation is calculated for each member
> df <- sapply(ex2$MemberData, pr.accum)
> # Creating a SpatialGridDataFrame
> sgdf.mn <- SpatialGridDataFrame(ex2$LonLatCoords, data = as.data.frame(df))
> # Preparation of world map lines for superposing onto the maps
> data(world_map)
> l1 <- list("sp.lines", as(world_map, "SpatialLines"))
> # Generates Fig 2
> spplot(sgdf.mn, scales = list(draw = TRUE), aspect = "iso", sp.layout = list(l1),
+   col.regions = rev(topo.colors(21)), layout = c(2,1))
```

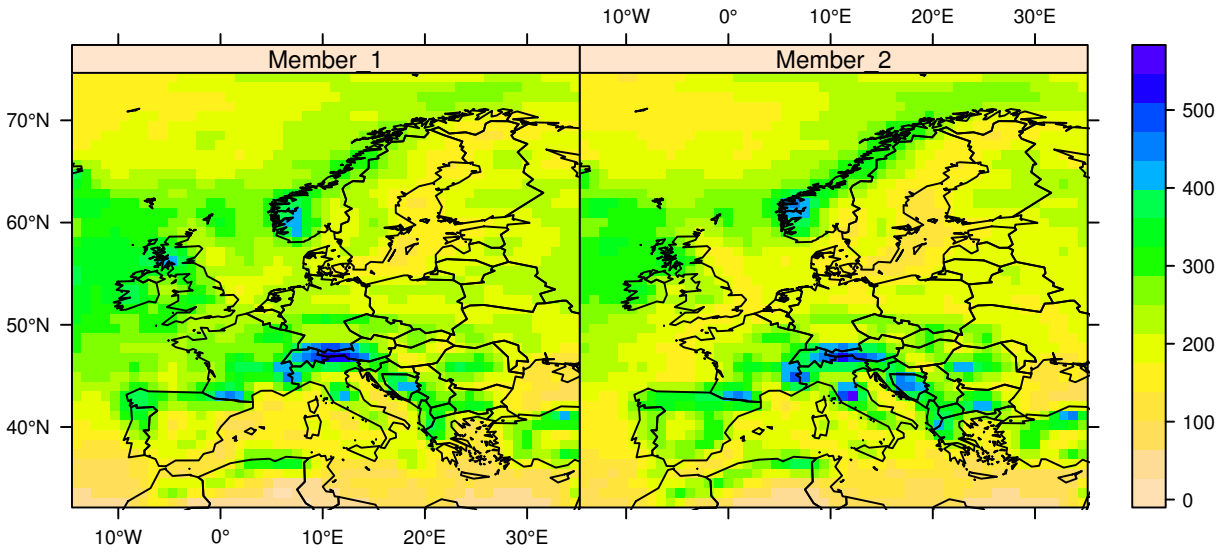

Figure 2: 3-month lead forecast for mean MAM precipitation (mm) of the NCEP's CFSv2 model for Europe considering the first two members and the period 2001-2010.

The calculation of the 95th percentile is next performed from the data matrix of the 1st member:

```
> # Function to compute the 95th percentile
> p95 <- function(x) {quantile(x, probs = .95, names = FALSE)}
> p95.m1 <- apply(ex2$MemberData[[1]], MARGIN = 2, FUN = p95)
> # Creation of a SpatialGridDataFrame
> sgdf.p95 <- SpatialGridDataFrame(ex2$LonLatCoords, as.data.frame(p95.m1))
> names(sgdf.p95) <- "Member_1"
> # Generates Fig. 3
> spplot(sgdf.p95, scales = list(draw = TRUE), sp.layout = list(l1), aspect = "iso",
+ main = "95th percentile of 6-h MAM precip (1st member)")
```

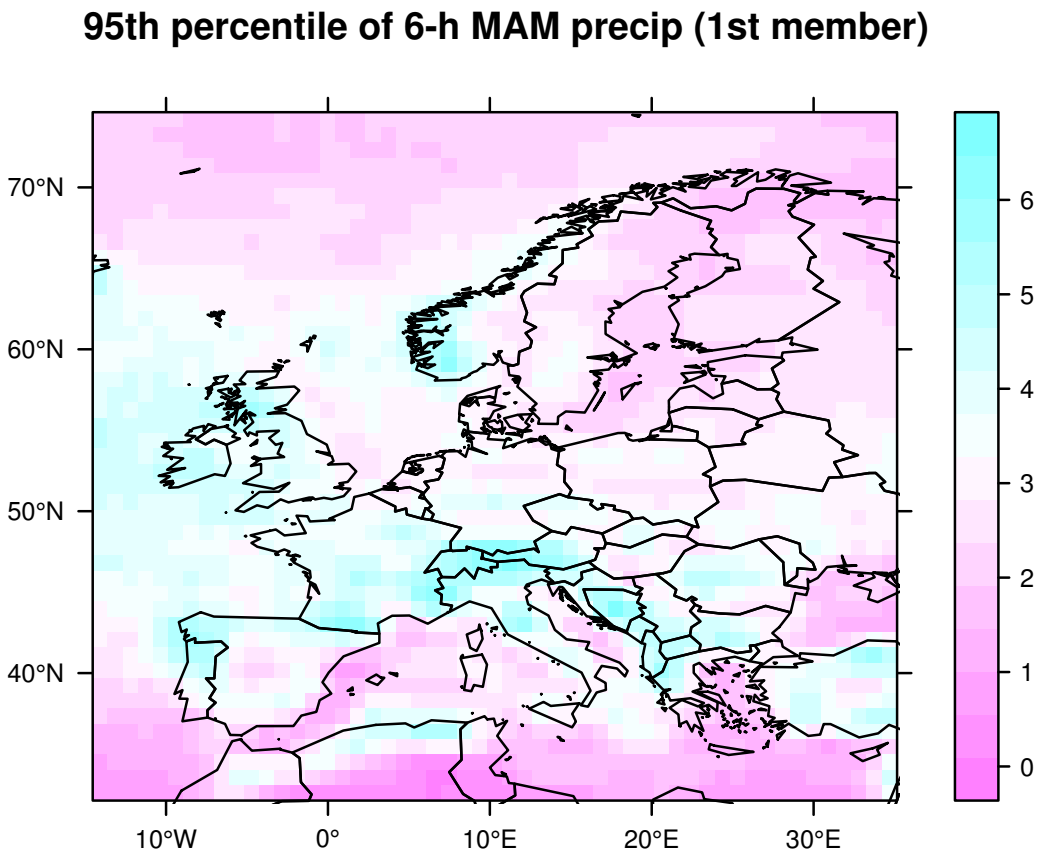

Figure 3: 95th percentile of 6-hourly precipitation (mm/6h) predicted at 3-month lead for spring (MAM, 2001-2010), calculated from the 1st member of the CFSv2 model.