Introduction and usage recommendations

In this section a number of examples for data download and visualization/analysis are presented within the R environment, through the `loadECOMS` function. Currently, there are four different seasonal to annual hindcasts and one observational gridded dataset available at **ECOMS-UDG**. All of them are available through the common interface `loadECOMS`, and therefore the argument values may vary slightly. For instance, arguments `members` and `leadMonth` do not apply in the case of the observational gridded dataset WFDEI, and hence ignored. Similarly, the output data structure may vary consequently, and forecast data types include the initialization dates and the names of the chosen members, while this information is not included for other types of gridded data.

The examples given have been kept deliberately simple in order to preserve a moderate output size (<150 Mb) and reasonable execution times (<10 minutes), although larger (and hence more time-consuming) requests can be done. The limitations in data loading depend essentially on two factors:

1. **Object size**: Requesting too large objects may deplete the available memory. Currently R runs on 32- and 64-bit operating systems, and most 64-bit OSes (including Linux, Solaris, Windows and OS X) can run either 32- or 64-bit builds of R. The memory limits depends mainly on the build, but for a 32-bit build of R on Windows they also depend on the underlying OS version. For more details, type in the R console `help("Memory-limits")`
2. **Loading time**: The time spent in a request does not depend exclusively on the size of the object to be loaded, but also 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 and try to divide the job into smaller queries. Also note that the first request after logging-in into ECOMS runs slower. This is due to some information that is stored in the cache memory after the first data request, notably improving the performance subsequently.

In the particular case of global domain selections (`lonLim` and `latLim` arguments set to `NULL`) for forecast data, it is recommended that only single-member, single-year selections are performed, due to the large size of this type of requests. Note that this is just an approximate recommendation. Object sizes also depend on the spatial resolution (CFS has approximately 1º horizontal res., while System4 is 0.75º and WFDEI 0.5º). Similarly, while GCM data will normally return data for the whole Earth (including oceans) for most variables, many observational datasets (like WFDEI) provide only data for land areas. In addition, It is always advisable to temporally aggregate to the maximum level possible. To this aim, it is possible to aggregate monthly using the argument `aggr.m` to specify a monthly aggregation function (see EXAMPLE 3), which dramatically reduces the size of the data, allowing for large global domain data requests. Type `help("loadECOMS")` for details on time aggregation options.

Basic loading examples

The `loadeR.ECOMS` package has been built using a (Linux) 32-bit OS, as well as all examples and time estimates. Thus, the performance may vary significantly depending on the aforementioned factors. As a rule of thumb, this would be a recommended usage of the `loadECOMS` function:

- EXAMPLE 1: Single Point Selections: Objects in this case are relatively small because the longitude and latitude dimensions are dropped. As a result, it is possible to load all members (in the case of forecast datasets) for the whole time span (typically around 30 years) without memory problems.
- EXAMPLE 2: Regional-Continental Domain Selections: Depending on the size of the spatial window, it may be possible to access all members, but preferably a few members should be selected, and time spans no longer than one decade.

Additional examples

- EXAMPLE 3: Calculating multi-member bias: A worked example on how to compute the bias of several members of a forecast (CFS) against an observed reference (WFDEI), and on how to specify monthly aggregations of the original data.
- EXAMPLE 4: Analysis of model drift: A worked example on how to compute and visualize the model drift of a forecast system.
- EXAMPLE 5: Bias correction and data export: A worked example on how to perform bias correction on seasonal forecasting data and, additionally, exporting the results in netcdf format.