Multi-model agreement

Multi-model agreement performs a comparison of Climate Model Projections anomalies season?? This vignette aims to illustrate a step-by-step example of how to perform a Multi-model agreement assessment using s2dverification functionalities.

1- Load dependencies

This example requires the following system libraries:

- libssl-dev
- libnecdf-dev
- cdo

You will need to install specific versions of certain R packages as follows:

```
Sys.setenv(TAR = '/bin/tar')
library(devtools)
install_git('https://earth.bsc.es/gitlab/es/startR', branch = 'develop-hotfixes-0.0.2')
install_git('https://earth.bsc.es/gitlab/es/easyNCDF', branch = 'master')
```

Too functions should be load by running the following lines in R, until integrated into current version of s2dverification

```
source('https://earth.bsc.es/gitlab/es/s2dverification/raw/develop-MagicWP5/R/AnoAgree.R')
source('https://earth.bsc.es/gitlab/es/s2dverification/raw/develop-Magic_WP6/R/WeightedMean.R')
```

All the other R packages involved can be installed directly from CRAN and loaded as follows:

```
library(s2dverification)
library(startR)
library(multiApply)
library(ggplot2)
```

2- Load files with the `Start ()` function included in the package StartR.

The full description of `Start()` and examples of its use can be found here: https://earth.bsc.es/gitlab/es/startR.

This example will use data generated by an ESMValTool namelist. The four models compared are:

- CNRM-CM5 r1i1p1
- IPSL-CM5A-LR r2i1p1
- MIROC5r2i1p1
- MPI-ESM-LR r1i1p1

First is loaded the historical period to compute the mean for the periode 1961-1990.

```r
model_names <- c('CNRM-CM5 r1i1p1', 'IPSL-CM5A-LR r2i1p1', 'MIROC5r2i1p1', 'MPI-ESM-LR r1i1p1')
climatology_filenames <- paste0("/home/Earth/ahunter/Magic/Magic_input/",
              c("tas_Amon_CNRM-CM5_historical_r1i1p1_195001-200512_remap.nc",
                "tas_Amon_IPSL-CM5A-LR_historical_r2i1p1_185001-200512_remap.nc",
                "tas_Amon_MIROC5_historical_r1i1p1_185001-201212_remap.nc",
                "tas_Amon_MPI-ESM-LR_historical_r1i1p1_185001-200512_remap.nc"))
reference_data <- Start(model=climatology_filenames, var = var0 <- 'tas',
                        var_var = 'var_names',
                        time = values(list(as.POSIXct(start_climatology <- '1961-01-01'),
                                    as.POSIXct(end_climatology <- '1990-12-31'))),
                        lat = 'all',
                        lon = 'all',
                        lon_var = 'lon',
                        return_vars = list(time = 'model', lon = 'model', lat = 'model'),
                        retrieve = TRUE)

model_filenames <- paste0("/home/Earth/ahunter/Magic/Magic_input/",
              c("tas_Amon_CNRM-CM5_rcp85_r1i1p1_200601-205512_remap.nc",
                "tas_Amon_IPSL-CM5A-LR_rcp85_200601-230012_remap.nc",
                "tas_Amon_MIROC5_rcp85_r1i1p1_200601-210012_remap.nc",
                "tas_Amon_MPI-ESM-LR_rcp85_200601-210012_remap.nc"))
rcp_data <- Start(model = model_filenames,
                  var = var0,
                  var_var = 'var_names',
                  time = values(list(as.POSIXct(start_anomaly <- '2020-01-01'),
                                    as.POSIXct(end_anomaly <- '2050-12-31'))),
                  lat = 'all',
                  lon = 'all',
                  lon_var = 'lon',
                  lon_reorder = CircularSort(0, 360),
                  return_vars = list(time = 'model', lon = 'model',
                                     lat = 'model'),
                  retrieve = TRUE)
```

The object returned by `Start()` contains, among others, four arrays with the requested monthly temperature data for each model. `summary()` function allows to check their content.

```
> dim(reference_data)
model   var  time   lat   lon
    4     1   360    96   192
> summary(reference_data)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  193.5   268.4   283.0   278.0   295.7   319.5
> dim(rcp_data)
model   var  time   lat   lon
    4     1   373    96   192
> summary(rcp_data)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
  198.2   270.5   284.3   279.7   297.1   322.0   55296
```

3- Multi-model agreement based on seasonal analysis

The multi-model agreement is a comparison based on seasonal anomalies, which are computed and saved by following these steps:

- `Season()` function from s2dverification package returns the anual time series for the selected season by defining the parameters of initial month of the data (`rmonini = 1`), the first month of the season (`rmoninf = 6`) and the final month of the season (`rmonsup = 8`). The last two parameters, `moninf` and `monsup`, have their origin with respect to the first one `monini`.

```
reference_seasonal_mean <- Season(reference_data, posdim = time_dim_ref <- which(names(dim(reference_da
                                  monini <- 1, moninf <- 6, monsup <- 8)
proj_seasonal_mean <- Season(rcp_data, posdim = time_dim_proj <- which(names(dim(rcp_data)) == "time"),
                                  monini = monini, moninf = moninf, monsup = monsup)
```

In this exemple, the boreal summer is shown by computing Juny, July and August monthly mean. The function `Season()` returns a multidimensional array containing the sesaonl time series for each model.

```
> dim(reference_seasonal_mean)
model   var  time   lat   lon
    4     1    30    96   192
> dim(proj_seasonal_mean)
model   var  time   lat   lon
    4     1    31    96   192
```

- Now the mean climatology value for each grid point and model can be computed by using the `Mean1Dim()` function belonging to the s2dverification package. Also a new dimension will be added to the object return bu `Mean1Dim()` funtion by running the function of the same package `InsertDim()` in order to obtain the same dimensions of the projections data. This `InsertDim()` has repeated the original data the required number of times.

```
climatology <- Mean1Dim(reference_seasonal_mean, time_dim_ref)
> dim(climatology)
model   var   lat   lon
    4     1    96   192
climatology <- InsertDim(climatology, posdim = time_dim_proj,
          lendim = dim(proj_seasonal_mean)[time_dim_proj])
> dim(climatology)
model   var  time   lat   lon
    4     1    31    96   192
```

- The anomaly for each model is obtain by regular substraction.

```
anomaly <- proj_seasonal_mean - climatology
> dim(anomaly)
model   var  time   lat   lon
    4     1    31    96   192
```

- In order to save this data some characteristics should be defined and the directory on to save it where you are currently executing your R session.

```
plot_dir <- "./plots_anomaly_agreement_main"
dir.create(plot_dir, recursive = TRUE)
time <- seq(as.numeric(substr(start_anomaly,1,4)), as.numeric(substr(end_anomaly,1,4)), by = 1)
month <- moninf  # WHY APRIL????
if (month <= 9) {
  month <- paste0(as.character(0), as.character(month))
}
month <- paste0("-", month, "-")
day <- "01"
time <- as.POSIXct(paste0(time, month, day), tz = "CET")
time <- julian(time, origin = as.POSIXct("1970-01-01"))
attributes(time) <- NULL
dim(time) <- c(time = length(time))
metadata <- list(time = list(standard_name = 'time', long_name = 'time', units = 'days since 1970-01-01
attr(time, "variables") <- metadata
#Save the single model anomalies
units <- attr(reference_data, 'Variables')$common[[var0]]$units
lat <- attr(reference_data, "Variables")$dat1$lat
lon <- attr(reference_data, "Variables")$dat1$lon
months <- paste0(month.abb[moninf],"-", month.abb[monsup])
attributes(lon) <- NULL
attributes(lat) <- NULL
dim(lon) <-  c(lon = length(lon))
dim(lat) <- c(lat = length(lat))
for (mod in 1 : dim(anomaly)[which(names(dim(anomaly)) == "model")]) {
  data <- anomaly[mod,1, , ,]
  data <- aperm(data, c(2,3,1))
  names(dim(data)) <- c("lat", "lon", "time")
  metadata <- list(variable = list(dim = list(list(name='time', unlim = FALSE))))
  attr(data, 'variables') <- metadata
  variable_list <- list(variable = data, lat = lat, lon = lon, time = time)
  names(variable_list)[1] <- var0
ArrayToNetCDF(variable_list,
              paste0(plot_dir, "/", var0, "_",months, "_multimodel-anomaly_", model_names,"_", start_ano
}
```

The next ncdf files will contain the anomaly data for each model:

- tas_Jun-Aug_anomaly_CNRM-CM5 r1i1p1_2020-01-01_2050-12-31_1961-01-01_1990-12-31.nc
- tas_Jun-Aug_anomaly_IPSL-CM5A-LR r2i1p1_2020-01-01_2050-12-31_1961-01-01_1990-12-31.nc
- tas_Jun-Aug_anomaly_MIROC5r2i1p1_2020-01-01_2050-12-31_1961-01-01_1990-12-31.nc
- tas_Jun-Aug_anomaly_MPI-ESM-LR r1i1p1_2020-01-01_2050-12-31_1961-01-01_1990-12-31.nc

4- Multi-model agreement visualitzation

Considering the time average anomalies for all models (which will be saved in `average` object), `AnoAgree()` function calculates the percentages of models which agrees with a positive or negative mean in each grid point.

```
average <- Mean1Dim(anomaly,  which(names(dim(anomaly)) == "time"))
> dim(average)
model   var   lat   lon
```

```
    4     1    96   192
agreement <- AnoAgree(average, members_dim = which(names(dim(average)) == "model"))
> dim(agreement)
[1]   1  96 192
```

So, in a four models comparison case, the `agreement` object can take the next values: 100 (all models agree), 75 (only one model has oposite sign), 50 (only two models agree with the mean signal) and 25 (one model agree with the sign of the mean signal because its magnitude is higher that the other three models) per cent. This values will change with the number of compared models

The next question will be answering by the example plot: Where do 80 % or more models agree in the signal? To obtain this plot the next lines should be running in R. Remember you can modify the threshold by modifying the parameter `agreement_threshold`. The plot will be saved in the directory previously created with the name:

- Jun-Augtasanomaly(2050-2020)-(1990-1961).png

```
  agreement_threshold <- 80
  colorbar_lim <- ceiling(max(abs(max(average)),abs(min(average))))
  brks <- seq(-colorbar_lim, colorbar_lim, length.out = 11)
  title <- paste0(months, " ", var0, " anomaly (", substr(end_anomaly, 1, 4), "-", substr(start_anomaly
                 ") - (", substr(end_climatology, 1, 4), "-", substr(start_climatology, 1, 4), ")")

  PlotEquiMap(drop(Mean1Dim(average, which(names(dim(average)) == "model"))), lat = lat, lon = lon, uni
              dots = drop(agreement) >= agreement_threshold,
              fileout = paste0(plot_dir, "/", paste0(unlist(strsplit(c(title), " ")), collapse=""), ".p
```

Or selecting a region in the Euro-Atlantic domain [40ºW-45ºE; 20-60ºN]. (EuroAtl_Jun-Augtasanomaly(2050-2020)-(1990-1961).png)

```
title <- paste0("EuroAtl_", months, " ", var0, " anomaly (", substr(end_anomaly, 1, 4), "-", substr(sta
                 ") - (", substr(end_climatology, 1, 4), "-", substr(start_climatology, 1, 4), ")")
  PlotEquiMap(drop(Mean1Dim(average, which(names(dim(average)) == "model")))[60:81,c(1:25,172:192)], la
              dots = drop(agreement)[60:81,c(1:25,172:192)] >= agreement_threshold,
              fileout = paste0(plot_dir, "/", paste0(unlist(strsplit(title, " ")), collapse=""), ".png"
```

To visualize the time evolution of models agreements, the spatial average is performed by a grid pixel size weithed mean.

```
temporal <- drop(WeightedMean(anomaly, lon = lon, lat = lat, mask = NULL))
 temporal <- Smoothing(temporal, runmeanlen = 5, numdimt = time_dim <- 2)

data_frame <- as.data.frame.table(t(temporal))
years <- rep(substr(start_anomaly,1,4) : substr(end_anomaly,1,4),4)
data_frame$Year <- c(years)
names(data_frame)[2] <- "Model"
for (i in 1 : length(levels(data_frame$Model))) {
levels(data_frame$Model)[i] <- model_names[i]
}

   g <- ggplot(data_frame, aes(x = Year, y = Freq)) + theme_bw() +
       ylab(paste0("Anomaly (", units, ")")) +  xlab("Year") + theme(text=element_text(size = 12),lege
```

```
                                                                    axis.title=element_text(size = 12)
        stat_summary(data =  data_frame, fun.y= "mean", mapping = aes(x = data_frame$Year, y = data_fram
                                                                     color = data_frame$Model), geom =
        stat_summary(data =  data_frame, geom = "ribbon", fun.ymin = "min", fun.ymax = "max", mapping =
         ggtitle(paste0(months, " ", var0, " anomaly (", start_anomaly, "-", end_anomaly,") - ", "(", st
     ggsave(filename = paste0(plot_dir, "/", "Shadow_Area-averaged ", var0, "_",months, "_multimodel-ano
```

```
g <- ggplot(data_frame, aes(x = Year, y = Freq, color = Model)) + theme_bw() +
        geom_line() + ylab(paste0("Anomaly (", units, ")")) +  xlab("Year")
ggsave(filename = paste0(plot_dir, "/", "Individual_Area-averaged_", var0, "_",months, "_multimodel-anom
```