s2dverification update meeting

September 15th, 2017

Nicolau Manubens, Alasdair Hunter

Please subscribe to the mailing list

Send an e-mail with the subject ‘subscribe’ to the following address:

s2dverification-request@bsc.es

Outline

- Status and documentation
- How to work with currently available tools
- Chunking strategy
- Developed functionality not yet available in s2dverification
- Development plan
- Release plan
- Questions

Status and documentation

Latest version is s2dverification v2.8.3, installed in the machines

Many developments ongoing, available on GitLab

EMS paper: received response. Trying to get it accepted with minor reviews

Link to the s2dverification wiki

Link to the GitLab page

Links to vignettes: s2dv user guide (outdated), computing weather regimes

Link to the CRAN page

How to: Load multiple NetCDF files

3 options:

- s2dverification::Load
  Convenient for simple cases

- /shared/earth/software/scripts/LoadMembersChunks.R
  Convenient to load file per chunk per member data sets
  Compatible with s2dverification

- startR::Start
  Convenient to load files with multiple regions, depth levels, on irregular grids…
  Can load file per chunk per member data sets
  Not compatible with s2dverification

How to: Load multiple NetCDF files

1. s2dverification::Load

Convenient for simple cases

```
data <- Load('tos', 'i00k', 'erainterim', paste0(1985:2005, '1101'),
             leadtimemax = 6, output = 'lonlat')
```

How to: Load multiple NetCDF files

2. /shared/earth/software/scripts/LoadMembersChunks.R

Convenient to load file per chunk per member data sets

Compatible with s2dv

```
source('/shared/earth/software/scripts/LoadMembersChunks.R')

new_exp <- paste0('/esarchive/exp/ecearth/t00p/monthly_mean/',
                  '$VAR_NAME$_f6h/$VAR_NAME$_Omon_EC-EARTH3_t00p_',
                  'S$START_DATE$_$MEMBER$_$CHUNK$.nc')
members <- list('19900101' = 'r1i1p1')
chunks <- list('19900101' = c('199001-199001', '199002-199002'))

data <- LoadMembersChunks('tos', new_exp, 'erainterim', '19900101',
                          members, chunks, ftimes_per_chunk = 1,
                          output = 'lonlat')
```

How to: Load multiple NetCDF files

3. startR::Start

Convenient to load files with multiple regions, depth levels, on irregular grids, …
Can load file per member per chunk
Not compatible with s2dv yet. Not all functionality available in Load is available in Start

```
library(startR)

exp <- paste0('/esarchive/exp/ecearth/t00p/monthly_mean/',
              '$var$_*/$var$_*_S$sdate$_$member$_$chunk$.nc')

data <- Start(dataset = exp,          var = 'tos',
              sdate = '19900101',     member = 'all',    chunk = 'all',
              time = indices(1:6),    lat = 'all',       lon = 'all',
              chunk_depends = 'sdate',
              time_across = 'chunk')
```

How to: Save arrays into a NetCDF file

```
library(easyNCDF)

a <- array(1:400, dim = c(5, 10, 4, 2))
names(dim(a)) <- c('lat', 'lon', 'time', 'var')
ArrayToNc(list(tos = a, prlr = a), 'tmp.nc')
```

```
> ncdump -h tmp.nc
netcdf tmp {
dimensions:
    lat = 5 ;
    lon = 10 ;
    time = UNLIMITED ; // (4 currently)
variables:
    float tos_1(time, lon, lat) ;
    float tos_2(time, lon, lat) ;
    float prlr_1(time, lon, lat) ;
    float prlr_2(time, lon, lat) ;
}
```

How to: Use downscaleR

```
##################
# Load s2dv data #
##################

library(s2dverification)
sdates <- paste0(1981:2005, '1101')

data_train <- Load('tasmax', 'ecmwf/system4_m1', NULL, sdates[1:15],
                   leadtimemin = 1, leadtimemax = 2, output = 'lonlat',
                   lonmin = 0, lonmax = 10, latmin = 20, latmax = 30,
                   nmember = 3, nprocs = 2)

data_test <-  Load('tasmax', NULL, 'jra55', sdates[16:25],
                   leadtimemin = 1, leadtimemax = 2, output = 'lonlat',
                   lonmin = 0, lonmax = 10, latmin = 20, latmax = 30,
                   nmember = 3, nprocs = 2)
```

How to: Use downscaleR

```
####################################
# Transform objects to downscaleR #
####################################

# Load code for bridging functions
source('https://earth.bsc.es/gitlab/es/s2dverification/raw/develop-interface-downR/R/
        DownRToS2dv.R')
source('https://earth.bsc.es/gitlab/es/s2dverification/raw/develop-interface-downR/R/
        S2dvToDownR.R')

# Apply them
data_train <- S2dvToDownR(data_train)
data_test <- S2dvToDownR(data_test)
```

How to: Use downscaleR

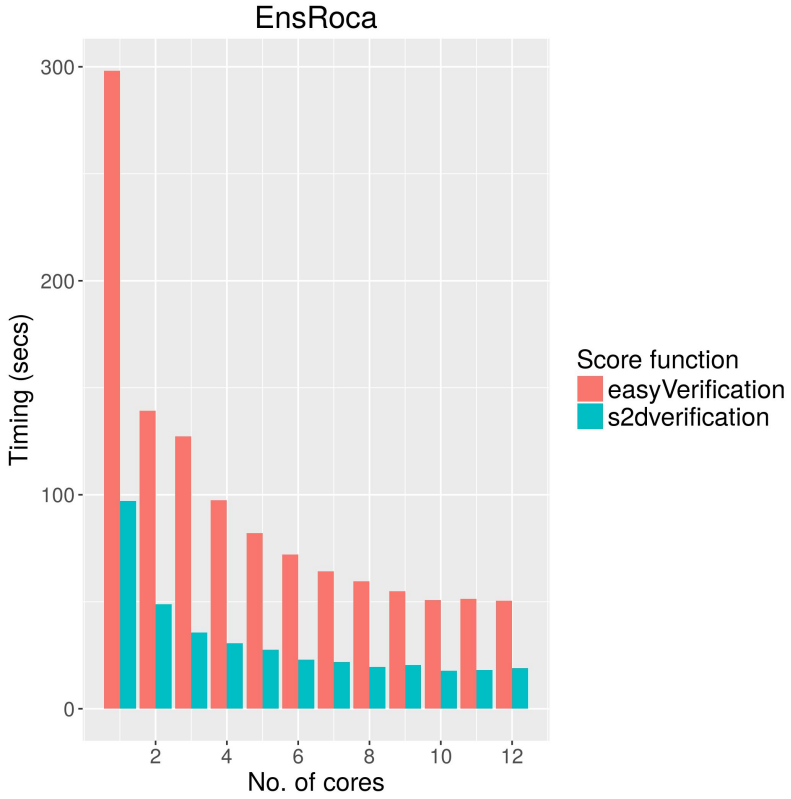
Only one forecast time step supported simultaneously

```
##############################
# Apply downscaleR functions #
##############################
library(downscaleR)

# Downscaling obs + exp data for a single forecast time step, with the GLM method
ds <- downscale(subsetGrid(data_test$obs$"jra55"$"ftimes 1 to 2", season = 11),
                subsetGrid(data_test$obs$"jra55"$"ftimes 1 to 2", season = 11),
                subsetGrid(data_train$exp$"ecmwf"$"ftimes 1 to 2", season = 11),
                method = 'glm')

##################################
# Convert back to s2dv           #
##################################
result <- DownRToS2dv(ds)
```

May develop Downscale() and BiasCorrect() functions in the future, which apply downscaleR functions to s2dverification arrays with multiple forecast time steps

How to: use multiApply to calculate scores/ extremes/anything in parallel.

- Extension of veriApply to a wider range of applications.
- Generally faster than using veriApply (or plyr functions) directly.
- Will be extended in the coming months to assist with the propagation of metadata
- See explanation, worked examples and some performance tests here.



How to: use multiApply (Forecast binning)

Can use multiApply for preprocessing, e.g. forecast binning (see here for a faster C++ implementation of ProbBins)

```
library(multiApply)
fcst <- Load(...)$mod
obs <- Load(...)$obs
margins_fcst = list(c(3,4)) #Choose margins to apply the function over.
fcst_bins = Apply(list(fcst), margins = margins_fcst, AtomicFun = "Probins"
                  format = "probability", fcyr = "all",
                  thr = c(1/3, 2/3), parallel = TRUE)

 margins_obs = list(c(2, 3))
 obs_bins <-  Apply(list(obs), margins = margins_obs, AtomicFun = "ProbBins",
                   format = "probability",fcyr = "all",
                   thr = c(1/3, 2/3), parallel = TRUE)
```

How to: use multiApply (EnsRoca)

Worked example - forecast binning + EnsRoca

```
margins = list(c(3,4), c(3,4))
roca <- Apply(list(fcst_bins, obs_bins), margins = margins,
              AtomicFun = "EnsRoca", parallel = TRUE)
```

Comments:

- multiApply and ProbBins have been rigorously tested within the QA4Seas performance milestone and are therefore reliable.
- Extremes functions (heatwaves, droughts, floods) are also being developed under develop-MagicWP7 but are currently being tested for use in Magic.
- Computing-intensive functions in s2dverification will transparently use multiApply in the future.

How to: compute weather regimes

A function for computing weather regimes has been developed under the develop-SeaIceModes branch on the s2dv gitlab.

A vignette with an example is available here.

Currently the function takes arrays with dimensions c("sdate", "ftime", "lat", "lon"), and calculates the PCAs then applies clustering (k-means, hierarchical or k-medoids) to the data (anomalies and detrending should be done separately by the user).

Coming soon: Extension to multivariate input (apply the PCA analysis to the normalized data).

How to: use plotting functions

PlotAno, PlotClim, PlotACC, PlotBox, PlotVsLTime

Useful for s2dverification time-series
Hard to add components or customize

PlotEquiMap, PlotStereoMap, PlotLayout

Useful for s2dverification maps (data on lonlat/gaussian grids)
Lots of options but few projections and not 100% customizable

ggplot2

Highly customizable time-series, maps and layouts
Learning process. Not trivial to represent multi-dimensional arrays with ggplot2
PlotTimeSeries plots arrays that contain time-series using ggplot2

MapGenerator

Department tool for python to draw maps. Highly customizable
Will develop wrappers in the future to call MapGenerator from R. You can try using the rPython package

Chunking

You usually process complete data sets (or the complete subset you need)

This is becoming unsustainable

- Too much memory consumption
- Too much time

You should switch to processing data sets by chunks

Presentation on chunking, **Tuesday 19th September, 12:30, Aula Formación**

Developed functionality not yet available in s2d

Not yet included

- Extreme indices
- Weather regimes
- PlotTimeSeries (without provenance)
- downscaleR bridging functions
- Function to select lonlat regions
- Function to compute area-weighted means
- …

Scientists' developments (not fully adapted to s2dv):

- Vero's bias correction + Niti's improvements
- New plotting functionality: PlotMostLikelyTercile, hatching, contours, filled oceans
- Omar's functionality
- RMSE with bootstrap
- ...

Development plan

Hiring someone else soon. ESS projects upcoming

- Inclusion of already developed functionality
- Progressive integration of scientists' functionality
- Development of MAGIC functionality (which functionality?)
- QA4Seas provenance, time-series plots
- Integration of startR with Load
- Compatibility break with use of multiApply + improved interface
- Updating vignettes + user formation

Release plan

- s2dverification 2.9.0 by December
  - Extreme indices
  - Weather regimes
  - ...
- s2dverification 2.10.0 by February
  - PlotTimeSeries with provenance
  - startR integrated into Load
- s2dv 3.0.0 by April
  - Clean interface
  - No need to know array dimension order

In the meantime you can use functionality available in the development branches

Questions?

Suggestions?

Problems?

Needs?

Thank you for your attention

nicolau.manubens@bsc.es, alasdair.hunter@bsc.es