



**Barcelona
Supercomputing
Center**

Centro Nacional de Supercomputación



**EXCELENCIA
SEVERO
OCHOA**

R tools user meeting

23/06/2020

An-Chi Ho and Núria Pérez-Zanón

Agenda

1. Package review & update
 - s2dv
 - startR
 - CSTools
2. Issue discussion
3. Instruction of how to use Jupyter lab with R
4. The situation of using R in Nord3 and the problems you face
5. A brief illustration of shiny app -- a visualization tool in R language

In-house packages status

PACKAGE	LAST RELEASE	NEXT RELEASE	#ISSUES
s2dverification	October 2019	One more release & deprecated	89
s2dv	February 2020	July 2020	4
CSTools	February 2020	June 2020	23
ClimProjDiags	January 2020	-	2
startR	June 2020	Summer 2020	19
easyNCDF	March 2020	-	2
multiApply	September 2019	-	5

ACTION
Review
your issues
and
branches!

s2dv functions (v0.0.2)

Find details in [NEWS.md](#)

Data retrieval and formatting

Load
Reorder
InsertDim
LeapYear
ToyModel

Basic statistics

Clim	Ano
Eno	Smoothing
MeanDims	Composite
Season	
Trend	

Skill score

Corr
Regression
RMS
RMSSS
RandomWalkTest
Persistence

Configuration

ConfigApplyMatchingEntries
ConfigEditDefinition
ConfigEditEntry
ConfigFileOpen
ConfigShowSimilarEntries
ConfigShowTable

Indices

AMV
GMST
GSAT
SPOD
TPI

Plotting

AnimateMap	PlotLayout
ColorBar	PlotMatrix
PlotClim	PlotSection
PlotEquiMap	PlotStereoMap
PlotAno	

Bugfixes (issue 59)

- Fix the wrong positions in output array when requiring repetitive values from a single file using parameters 'merge_across_dims' and 'split_multiselected_dims'.



obs (daily data, one month per file)

To obtain the same dimensions as exp, we need to retrieve 12/29 to 12/31 from file '201912' twice, one for [sdate = 1, time = 8:10] and one for [sdate = 2, time = 1:3]

Bugfixes

- Specify the time zone to be 'UTC' regarding time attributes retrieval. It is consistent with netCDF file now.

```
print(attr(eraI, 'Variables')$common$time)
```

```
# old
```

```
[21] "1994-09-21 00:00:00 CEST" "1994-09-22 00:00:00 CEST"
```

```
[23] "1994-09-23 00:00:00 CEST" "1994-09-24 00:00:00 CEST"
```

```
[25] "1994-09-25 00:00:00 CEST" "1994-09-25 23:00:00 CET"
```

```
[27] "1994-09-26 23:00:00 CET" "1994-09-27 23:00:00 CET"
```

```
# new
```

```
[21] "1994-09-21 UTC" "1994-09-22 UTC" "1994-09-23 UTC" "1994-09-24 UTC"
```

```
[25] "1994-09-25 UTC" "1994-09-26 UTC" "1994-09-27 UTC" "1994-09-28 UTC"
```

CSTools content

Find details in [NEWS.md](#)

Basic functions

CST_Load
CST_Anomaly
CST_SaveExp
CST_SplitDim
CST_MergeDims
s2dv_cube
as.s2dv_cube

Correction

CST_BiasCorrection
CST_Calibration
CST_QuantileMapping
CST_BEI_Weighting
BEI_Weights
BEI_PDFBest*
CST_CategoricalForecast

Downscaling

CST_Analogs
Analogs
CST_RainFARM
CST_RFSlope
CST_RFWeights
RainFARM
RFSlope

Evaluation

CST_MultivarRMSE
CST_MultiMetric

Plotting functions

PlotMostLikelyQuantileMap
PlotForecastPDF PlotPDFsOLE
PlotCombinedMap PlotTriangles4Categories

Classification

CST_WeatherRegimes WeatherRegimes
CST_RegimeAssign RegimeAssign
CST_EnsClustering CST_MultiEOF

+ 2 new vignettes: **Weather Regimes and Ensemble Clustering**

Future future functionalities: ADAMONT, SMOP and Dynamical Bias Correction

*RainFARM: Rainfall Filtered Autoregressive Model

Issue Discussion

startR - nested dependency on the path definition

```
path <-  
'/esarchive/exp/ecearth/$expid$/diags/CMIP/EC-Ea  
rth-Consortium/EC-Earth3/historical/$member$/Om  
on/$var$/gn/$version$/var$_Omon_EC-Earth3_his  
torical_$member$_gn_$year$.nc'
```

expid	member	version
a1st	r7i1p1f1	v20190302
a1sx	r10i1p1f1	v20190308

Problem

Start() is not able to recognize more than 1 dependency among the file dimensions.

Possible solution

Add a functionality of specifying the dependency pairs*, and combine these file dimensions into one dimension 'case' in the output array**.

**e.g., depend_pairs = list(c(expid = 'a1st', member = 'r7i1p1f1', version = 'v20190302'),
c(expid = 'a1sx', member = 'r10i1p1f1', version = 'v20190308'))*

**output dim

```
case time lon lat  
2 xx yy zz
```

Jupyter Lab

Jupyter is a loose acronym meaning **Julia, Python, and R**

What is a Jupyter notebook?

Document that contain both code and rich text elements, such as figures, links, equations,

Because this mix, these documents are the ideal place to bring together an **analysis description**, and its **results**, as well as, they can be executed **perform the data analysis in real time**.

The Jupyter Notebook App:

- produces these notebook documents
- allow to edit and run notebooks via a web browser
- can be executed on a PC without internet access
- can be installed on a remote server

Case 1) If you are at BSC or you want to open a remote browser

```
Terminal conected to your WS: ssh user@bscearthXXX.int.bsc.es
```

```
module load R/3.6.1-foss-2015a-bare  
module load jupyterlab/1.2.2-foss-2015a-Python-3.7.3
```

← Load required modules

```
jupyter-lab
```

← Run Jupyter Lab

Case 2) If you are working remotely and you want to open the browser locally

```
Terminal conected to your WS: ssh user@bscearthXXX.int.bsc.es
```

```
module load R/3.6.1-foss-2015a-bare  
module load jupyterlab/1.2.2-foss-2015a-Python-3.7.3
```

← Load required modules

```
jupyter-lab --no-browser --port=8888 --ip=bscearthXXX.int.bsc.es
```

← Run JupyterLab

Jupyter Lab in Nord 3

1. **connect** to Nord3
2. **create** the file **jupyter_nord3.cmd** in your Nord3 home (/home/bsc32/bsc32YYYY)
3. paste the content from the box and save
Note: if you use the info in the wiki add the R module and the X11 connection
Note: Change the number of cores with -n parameter
Note: Change the memory if needed with -M parameter

Find details in [ES-Wiki](#)



jupyter_nord3.cmd

```
#!/bin/bash
#BSUB -n 1
#BSUB -M 1800
#BSUB -oo output_%J.out
#BSUB -eo output_%J.err
#BSUB -J jupyter-notebook
#BSUB -W 01:00
#BSUB -q x11

# get tunneling info
XDG_RUNTIME_DIR=""
port=$(shuf -i8000-9999 -n1)
node=$(hostname -s)
user=$(whoami)

# print tunneling instructions jupyter-log
echo -e "

MacOS or linux terminal command to create your ssh
tunnel
ssh -N -L ${port}:${node}:${port}
${user}@nord1.bsc.es

Use a Browser on your local machine to go to:
localhost:${port} (prefix w/ https:// if using
password)
"

# load modules
module load jupyterlab R

# DON'T USE ADDRESS BELOW.
# DO USE TOKEN BELOW
jupyter-lab --no-browser --port=${port}
--ip=${node}
```



Jupyter Lab in Nord 3

4. submit this file to the nord 3 queue by running:

```
bsub < jupyter nord3.cmd
```

5. list the files in your Nord3 home and open

```
vim output_ZZZZZZ.out
```

6. copy the line

```
ssh -N -L 9029:s01r2b30:9029 bsc32339@nord1.bsc.es
```

7. paste in your local terminal, provide password and don't wait (nothing is going to happen)

8. from output_ZZZZZ.out, copy

```
localhost:9029
```

9. paste it in your local browser

10. close output_ZZZZ.out and open output_ZZZZ.err. Copy from there the token like:

```
a9fbf7057b8ef7d6af58ac3409e8005f45dbe94e06d54211
```

11. Paste it in the box in the browser

READY!!

Next time you can start in step 4.
remove output* when you finish using the session.

Jupyter Lab

- change boxes from 'Markdown' to 'code'
- press play button
- try letters, arrows and numbers and see what happen or see the video and documentation below
 - a,b,c,d,x,v, dd, y, m, enter, shift + enter,
 - add to text #, *, **, -, ```, ---

More info:

https://jupyterlab.readthedocs.io/en/stable/getting_started/overview.html

R in Nord3

Nord3 access and configuration

Information in the ES-Wiki: <https://earth.bsc.es/wiki/doku.php?id=computing:nord3>

Note: remember to add the module lines provided in the wiki to your `.bashrc` in Nord 3 home

Interactive usage:

- `bsub -q interactive -W 06:00 -n 1 -M 16000 -ls /bin/bash`
- `bsub -W 2:00 -n 1 -M 16000 -q x11 -o errorfile xterm`

startR configuration:

- you must follow the ES-Wiki and the [startR practical guide](#) (e.g.: there is information about how to set up the passwordless access)
- new FAQ about possible errors reported the first time you use Nord3 and how to solve them

<https://earth.bsc.es/gitlab/es/startR/-/blob/master/inst/doc/faq.md#6-errors-using-a-new-cluster-setting-nord3>

Nord 3 issues

Interactive usage:

- ❑ Default nodes (460-480 nodes): 32 GB/node
- ❑ Medium memory nodes (120 nodes): 64 GB/node
- ❑ High memory nodes (100 nodes): 128 GB/node

Plotting:

- [issue #10 in s2dv](#)
- [issue #52 in CSTools](#)

Loading data:

- Type I: [solved]

```
*** caught bus error ***
```

```
address 0x2b4060608000, cause 'non-existent physical address'
```

- Type II: [work around when they randomly happens ... randomly until we have a pattern]

```
Error unserialize(socklist[[n]]) : error reading from connection
```

temporal solution: `num_procs = 1 (startR)/ nprocs = 1 (Load)`

- Others? Contact us and remember to provide code to reproduce the problems.

Shiny app

shiny app

Shiny is an R package that makes it easy to build **interactive web apps** straight from R. With a visualization tool, you can compare the variables by simple clicks, saving the time and effort of repetitive coding.

Examples

- cp_shiny: http://bscesshiny.bsc.es:3838/es/cp_shiny/
 - A tool for visualizing and monitoring the BSC-ES experimental/diagnostic data.
 - Retrieve data and plot the figures on the fly
- C3S_34c-shiny_app (made by Carlos Delgado)
 - Pre-generate the figures then print them on the app

Any questions?