

CSTools R Package

Status and developments

Núria Pérez-Zanón,
Barcelona Supercomputing Center
(BSC)

Politecnico di Milano
Milan, 15 April, 2019



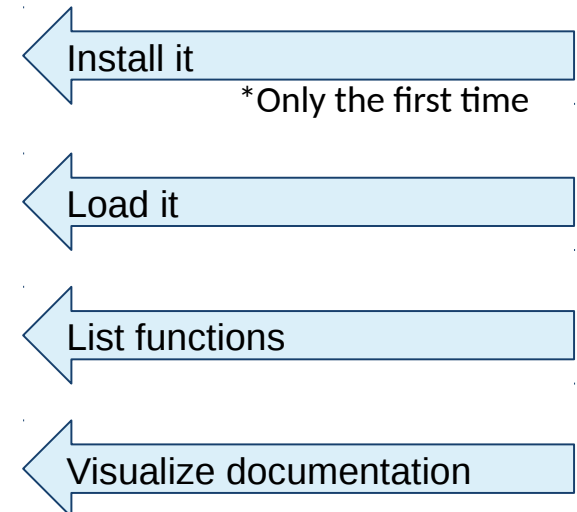


CSTools in R session

The CSTools R package v1.0.0 is published **on CRAN**
(*The Comprehensive R Archive Network*).

You can use it in R

```
> install.packages("CSTools")  
  
> library(CSTools)  
Loading required package: maps  
  
> ls("package:CSTools")  
  
> ?PlotForecastPDF
```





CSTools on CRAN

The CSTools R package v1.0.0 is published **on CRAN**
(*The Comprehensive R Archive Network*).

<https://CRAN.R-project.org/package=CSTools>

- Authorship:



- on the general list
- in the individual functions
- in the vignettes

- References

- in the general list and
- the individual functions
- in the vignettes



CSTools: Assessing Skill of Climate Forecasts on Seasonal-to-Decadal Timescales

Author: BSC-CNS [cph], Louis-Philippe Caron  [aut], Jost von Hardenberg  [aut], Nuria Perez-Zanon [aut, cre], Llorenç Lledo [aut], Nicolau Manubens [aut], Niti Mishra [ctb], Veronica Torralba [aut], Deborah Verfaillie [aut], Lauriane Batte [ctb], Jesus Peña [ctb], Bert van Schaeybroeck [ctb]



CSTools: Assessing Skill of Climate Forecasts on Seasonal-to-Decadal Timescales

multivariate verification, as well as basic and advanced tools to obtain tailored products. Doblas-Reyes et al. (2005) <[doi:10.1111/j.1600-0870.2005.00104.x](https://doi.org/10.1111/j.1600-0870.2005.00104.x)>. Mishra et al. (2018) <[doi:10.1007/s00382-018-4404-z](https://doi.org/10.1007/s00382-018-4404-z)>. Terzago et al. (2018) <[doi:10.5194/nhess-18-2825-2018](https://doi.org/10.5194/nhess-18-2825-2018)>. Torralba et al. (2017) <[doi:10.1175/JAMC-D-16-0204.1](https://doi.org/10.1175/JAMC-D-16-0204.1)>. D'Onofrio et al. (2014) <[doi:10.1175/JHM-D-13-096.1](https://doi.org/10.1175/JHM-D-13-096.1)>.



Package ‘CSTools’

April 24, 2019

Title Assessing Skill of Climate Forecasts on Seasonal-to-Decadal Timescales

Version 1.0.0

Description Exploits dynamical seasonal forecasts in order to provide information relevant to stakeholders at the seasonal timescale. The package contains process-based methods for forecast calibration, bias correction, statistical and stochastic downscaling, optimal forecast combination and multivariate verification, as well as basic and advanced tools to obtain tailored products.

Doblas-Reyes et al. (2005) <doi:10.1111/j.1600-0870.2005.00104.x>.
Mishra et al. (2018) <doi:10.1007/s00382-018-4404-z>.
Terzago et al. (2018) <doi:10.5194/nhess-18-2825-2018>.
Torralba et al. (2017) <doi:10.1175/JAMC-D-16-0204.1>.
D’Onofrio et al. (2014) <doi:10.1175/JHM-D-13-096.1>.

Depends R (>= 3.2.0), maps

Imports s2dverification, rainfarmr, multiApply, ncd4, plyr, abind, data.table, reshape2, ggplot2, graphics, grDevices, stats, utils

Suggests zeallot, testthat, knitr, rmarkdown

VignetteBuilder knitr

License Apache License 2.0

Encoding UTF-8

LazyData true

RoxygenNote 5.0.0

NeedsCompilation no

Author BSC-CNS [cph],
Louis-Philippe Caron [aut] (<<https://orcid.org/0000-0001-5221-0147>>),
Jost von Hardenberg [aut] (<<https://orcid.org/0000-0002-5312-8070>>),
Nuria Perez-Zanon [aut, cre],
Llorenç LLedo [aut],
Nicolau Manubens [aut],
Niti Mishra [ctb],



CST_BiasCorrection *Bias Correction based on the mean and standard deviation adjustment*

Description

This function applies the simple bias adjustment technique described in Torralba et al. (2017). The adjusted forecasts have an equivalent standard deviation and mean to that of the reference dataset.

Usage

```
CST_BiasCorrection(exp, obs)
```

Arguments

exp	an object of class <code>s2dv_cube</code> as returned by <code>CST_Load</code> function, containing the seasonal forecast experiment data in the element named <code>\$data</code>
obs	an object of class <code>s2dv_cube</code> as returned by <code>CST_Load</code> function, containing the observed data in the element named <code>\$data</code> .

Value

an object of class `s2dv_cube` containing the bias corrected forecasts in the element called `$data` with the same dimensions of the experimental data.

Author(s)

Verónica Torralba, <veronica.torralba@bsc.es>

References

Torralba, V., F.J. Doblas-Reyes, D. MacLeod, I. Christel and M. Davis (2017). Seasonal climate prediction: a new source of information for the management of wind energy resources. *Journal of Applied Meteorology and Climatology*, 56, 1231-1247, doi:10.1175/JAMC-D-16-0204.1. (CLIM4ENERGY, EUPORIAS, NEWA, RESILIENCE, SPECS)



CSTools: Assessing Skill of Climate Forecasts on Seasonal-to-Decadal Timescales

Vignettes: [Multi-model Skill Assessment](#)
[Multivariate RMSE](#)
[RainFARM](#)



Vignette is an instructive tutorial demonstrating **practical uses** of the software **with discussion** of the interpretation of the results.

Computational requirements

Multivariate Root Mean Square Error (RMSE)

To run this vignette, the next R packages should be installed and loaded:

```
library(s2dverification)
library(RColorBrewer)
```

Library *CSTools*, should be installed from CRAN and loaded:

```
install.packages("CSTools")
library(CSTools)
```

1.- Load data

In this example, the seasonal temperature and precipitation forecasts, initialized in november, will be used to assess the glosea5 seasonal forecasting system from the Met Office, by computing the multivariate RMSE for both temperature and precipitation.

The parameters defined are the initializing month and the variables:

```
mth = '11'
temp = 'tas'
precip = 'prlr'
```

The simulations available for this model cover the period 1992-2012. So, the starting and ending dates can be defined by running the following lines:

```
ini <- 1992
fin <- 2012
start <- as.Date(paste(ini, mth, "01", sep = ""), "%Y%m%d")
end <- as.Date(paste(fin, mth, "01", sep = ""), "%Y%m%d")
datseq <- format(seq(start, end, by = "year"), "%Y%m%d")
```

The grid in which all data will be interpolated should be also specified. The observational dataset used in this example is the EraInterim.

```
grid <- "256x128"
obs <- "erainterim"
```

Using the `CST_Load` function from **CSTool package**, the data available in our data store can be loaded. The following lines show how this function can be used. Here, the data is loaded from a previous saved `.RData` file: Ask nuria.perez at bsc.es for the data to run the recipe.

Analysis settings:

- variable
- region
- dataset
- ...



Vignette is an instructive tutorial demonstrating practical uses of the software with discussion of the interpretation of the results.

Using the `CST_Load` function from **CSTool package**, the data available in our data store can be loaded. The following lines show how this function can be used. Here, the data is loaded from a previous saved `.RData` file: Ask [nuria.perez at bsc.es](mailto:nuria.perez@bsc.es) for the data to run the recipe.

```
require(zeallot)

glosea5 <- list(path = '/esnas/exp/glosea5/specs-seasonal_ilp1/$STORE_FREQ$_mean/$VAR_NAME$_allmemb/$VAR_NAME$_$S

c(exp_T, obs_T) %<-%
  CST_Load(var = temp, exp = list(glosea5),
           obs = obs, sdates = dateseq, leadtimemin = 2, leadtimemax = 4,
           latmin = 25, latmax = 75, lonmin = -20, lonmax = 70, output = 'lonlat',
           nprocs = 1, storefreq = "monthly", sampleperiod = 1, nmember = 9,
           method = "bilinear", grid = paste("r", grid, sep = ""))

c(exp_P, obs_P) %<-%
  CST_Load(var = precip, exp = list(glosea5),
           obs = obs, sdates = dateseq, leadtimemin = 2, leadtimemax = 4,
           latmin = 25, latmax = 75, lonmin = -20, lonmax = 70, output = 'lonlat',
           nprocs = 1, storefreq = "monthly", sampleperiod = 1, nmember = 9,
           method = "bilinear", grid = paste("r", grid, sep = ""))
#save(exp_T, obs_T, exp_P, obs_P, file = "./tas_prlr_toydata.RData")

# Or use the following line to load the file provided in .RData format:
load(file = "./tas_prlr_toydata.RData")
```

CST_Load
loads
temperature data

CST_Load
loads
precipitation data

There should be four new elements loaded in the R working environment: `exp_T`, `obs_T`, `exp_P` and `obs_P`. The first two elements correspond to the experimental and observed data for temperature and the other are the equivalent for the precipitation data. It's possible to check that they are of class `sd2v_cube` by running:

```
class(exp_T)
class(obs_T)
class(exp_P)
class(obs_P)
```

checks for the
users to follow
the steps



Vignette is an **instructive tutorial** demonstrating **practical uses** of the software **with discussion** of the interpretation of the results.

Latitudes and longitudes of the common grid can be saved:

```
Lat <- exp_T$lat
Lon <- exp_T$lon
```

CST_Anomaly
cross validation
option

The next step is to compute the anomalies of the experimental and observational data using `CST_Anomaly` function, which could be applied over data from each variable, and in this case it's compute applying cross validation technique over individual members:

```
c(ano_exp_T, ano_obs_T) %<-% CST_Anomaly(exp = exp_T, obs = obs_T, cross = TRUE, memb = TRUE)
c(ano_exp_P, ano_obs_P) %<-% CST_Anomaly(exp = exp_P, obs = obs_P, cross = TRUE, memb = TRUE)
```

The original dimensions are preserved and the anomalies are stored in the `data` element of the correspondent object:

```
> str(ano_exp_T$data)
 num [1, 1:9, 1:21, 1:3, 1:35, 1:64] -1.647 1.575 2.77 0.048 -1.886 ...
 - attr(*, "dimensions")= chr [1:6] "dataset" "member" "sdate" "ftime" ...
> str(ano_obs_T$data)
 num [1, 1, 1:21, 1:3, 1:35, 1:64] 0.0235 1.546 1.3885 -0.344 -5.972 ...
 - attr(*, "dimensions")= chr [1:6] "dataset" "member" "sdate" "ftime" ...
```

checks for the
users to follow the
steps

Two lists containing the experiment `ano_exp`, and the observation, `ano_obs`, lists should be put together to serve as input of the function to compute multivariate RMSEs.

Furthermore, some weights can be applied to the difference variables based on their relative importance (if no weights are given, a value of 1 is automatically assigned to each variable). For this example, we'll give a weight of 2 to the temperature dataset and a weight of 1 to the precipitation dataset:

Setting inputs for
next step

```
ano_exp <- list(ano_exp_T, ano_exp_P)
ano_obs <- list(ano_obs_T, ano_obs_P)
weight <- c(2, 1)
```



Vignette is an instructive tutorial demonstrating **practical uses** of the software **with discussion** of the interpretation of the results.

2.- Computing and plotting multivariate RMSEs

The multivariate RMSE gives an indication of the forecast performance (RMSE) for multiple variables simultaneously. Variables can be weighted based on their relative importance. It is obtained by running the `CST_MultivarRMSE` function:

```
mvrmse <- CST_MultivarRMSE(exp = ano_exp, obs = ano_obs, weight)
```

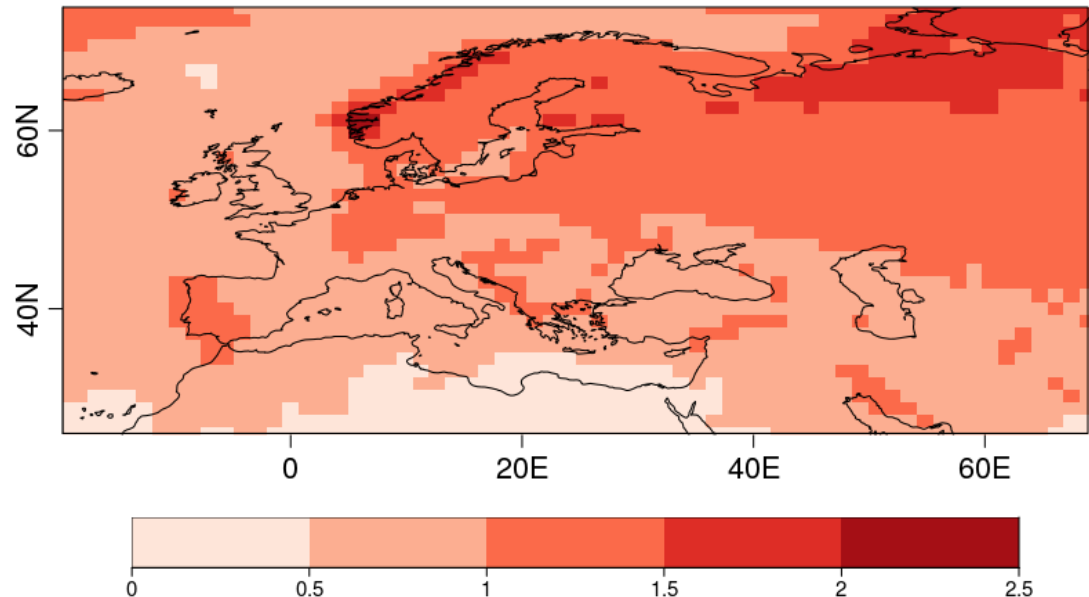
The function `CST_MultivarRMSE` returns the multivariate RMSE value for 2 or more variables. The output is a `CSTool` object containing the RMSE values in the `data` element and other relevant information:

```
> class(mvrmse)
> str(mvrmse$data)
 num [1, 1, 1, 1:35, 1:64] 0.764 0.8 0.67 0.662 0.615 ...
> str(mvrmse$Variable)
 Named chr [1:2] "tas" "prlr"
- attr(*, "names")= c
```

The following lines plot the

```
PlotEquiMap(mvrmse$dat
  toptitle =
  bar_limits
  fileout =
```

Multivariate RMSE tas, prlr 1992 - 2012



CST_MultivarRMSE
cross validation
option

checks for the
users to follow the
steps

Visualization



CSTools on GitLab

On-line software for Git repository management

<https://earth.bsc.es/gitlab/external/cstools>

- Version Control System:
 - share files
 - keep track of changes
 - safety development
- Utilities
 - issues tracking
 - continuous Integration
 - private repository



R package skeleton

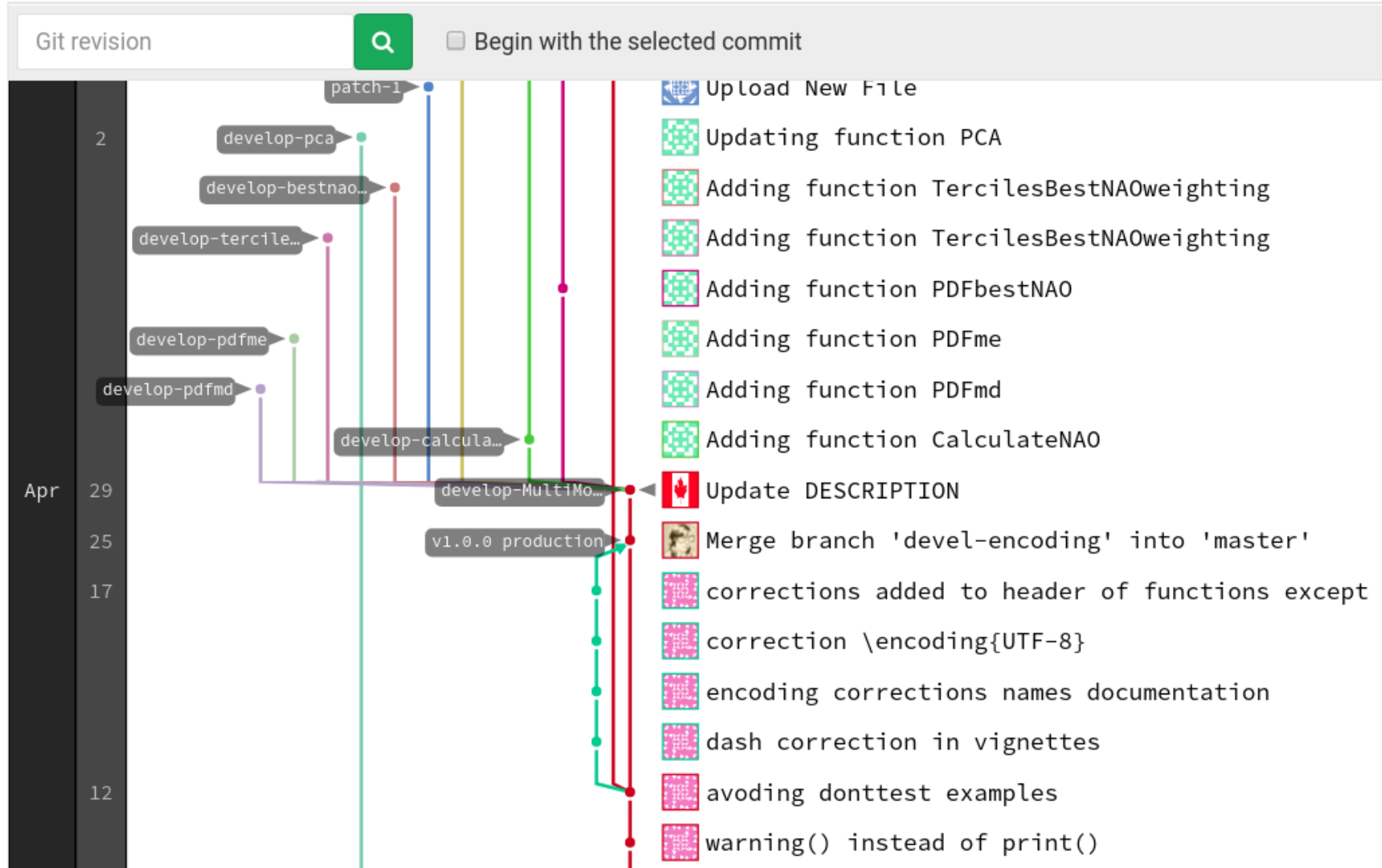
Name	
📁 R	← Functions code
📁 data	
📁 man	← Automatic built documentation
📁 tests	← Automatic run tests
📁 vignettes	← Vignettes!
📄 .Rbuildignore	
📄 .gitignore	
📄 .gitlab-ci.yml	
📄 DESCRIPTION	← General documentation
📄 NAMESPACE	
📄 README.md	



Contributions: by branch and commit

master

You can move around the graph by using the arrow keys.





If you find an error
If you have doubts about how to contribute
If you have doubts about the usage

Open an Issue

External > CStools > Issues

Open 4 Closed 5 All 9



Edit issues

New issue



Search or filter results...

Created date



MultiEOFs function

#9 · opened 1 day ago by Jost von Hardenberg



updated 1 day ago

Bug in PlotMostLikelyQuantileMap/PlotCombinedMap

#8 · opened 1 week ago by Verónica Torralba-Fernández Code bug



updated 1 week ago

Documentation updated: adding new function to MEDSCOPE

#3 · opened 3 months ago by Nuria Pérez-Zanón



updated 3 months ago

Type of license

#1 · opened 4 months ago by Jost von Hardenberg



updated 1 month ago

[Email a new issue to this project](#)



Branches:

PROTECTED

master Update DESCRIPTION

production Merge branch 'devel-encoding' into 'master'

DEVELOPMENTS

develop-SMOP: Computes a set of atmospheric parameters

dyn-bias_correction: New function to compute the properties of the underlying attractor.

develop-MultiModelEnsembleCombination

develop-ens-biascorrection

multieof: fix naming of output dimensions

[Issue #9](#)

develop-pdfbestnao: Updating function PDFbestNAO

develop-adamont: First Adamont functions adapted to CStools

patch-1 Upload New File

develop-pdfsfснао: Updating function PDFsfснао

develop-pca Updating function PCA

develop-bestnaoweighting Adding function TercilesBestNAOweighting

develop-tercilesbestnaoweighting Adding function TercilesBestNAOweighting

develop-pdfme Adding function PDFme

develop-pdfmd Adding function PDFmd

develop-calculatenaо Adding function CalculateNAO

BUG FIXES
ENHANCEMENTS

develop-PlotMost-bug: lower limit legend bug fix

[Issue #8](#)

RainFARM_docs



Useful links:

How to contribute

Development policy <https://bit.ly/2HjIF1L>

Guidelines <https://bit.ly/2E56AjQ>

Issues <https://earth.bsc.es/gitlab/external/cstools/issues>



09:00–13:00 **Split sessions**

WP3 (A3 room) workshop on tools + hands-on session on R formatting of the developed functions

nuria.perez@bsc.es

Thanks

