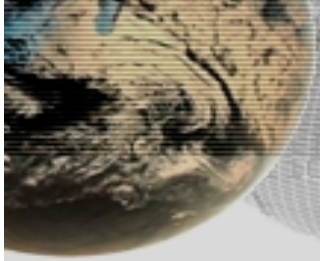


http://www.meteo.unican.es

Downscaling tools for adapting climate predictions to the user's needs

A.S. Cofiño, J.M. Gutiérrez, J. Fernández, J. Bedia, M. Vega, S. Herrera, M.D. Frías, M. Iturbide, M.E. Magariño, and R. Manzanas

http://www.meteo.unican.es/udg-wiki

Maialen Iturbide
miturbide@ifca.unican.es
Santander Meteorology Group

UNIVERSIDAD DE CANTABRIA CSIC EUPORIAS SPECS

Contents

1. The ECOMS User Data Gateway (UDG)
   - Harmonized access to virtual datasets
   - Transparent access using R: examples (`loadeR.ECOMS`)

2. R package `downscaleR` for downscaling
   - Worked bias correction example
   - Worked downscaling example

3. Integration with other R tools
   - Verification (`easyVerification`)
   - Forecast skill visualization (`visualizeR`)

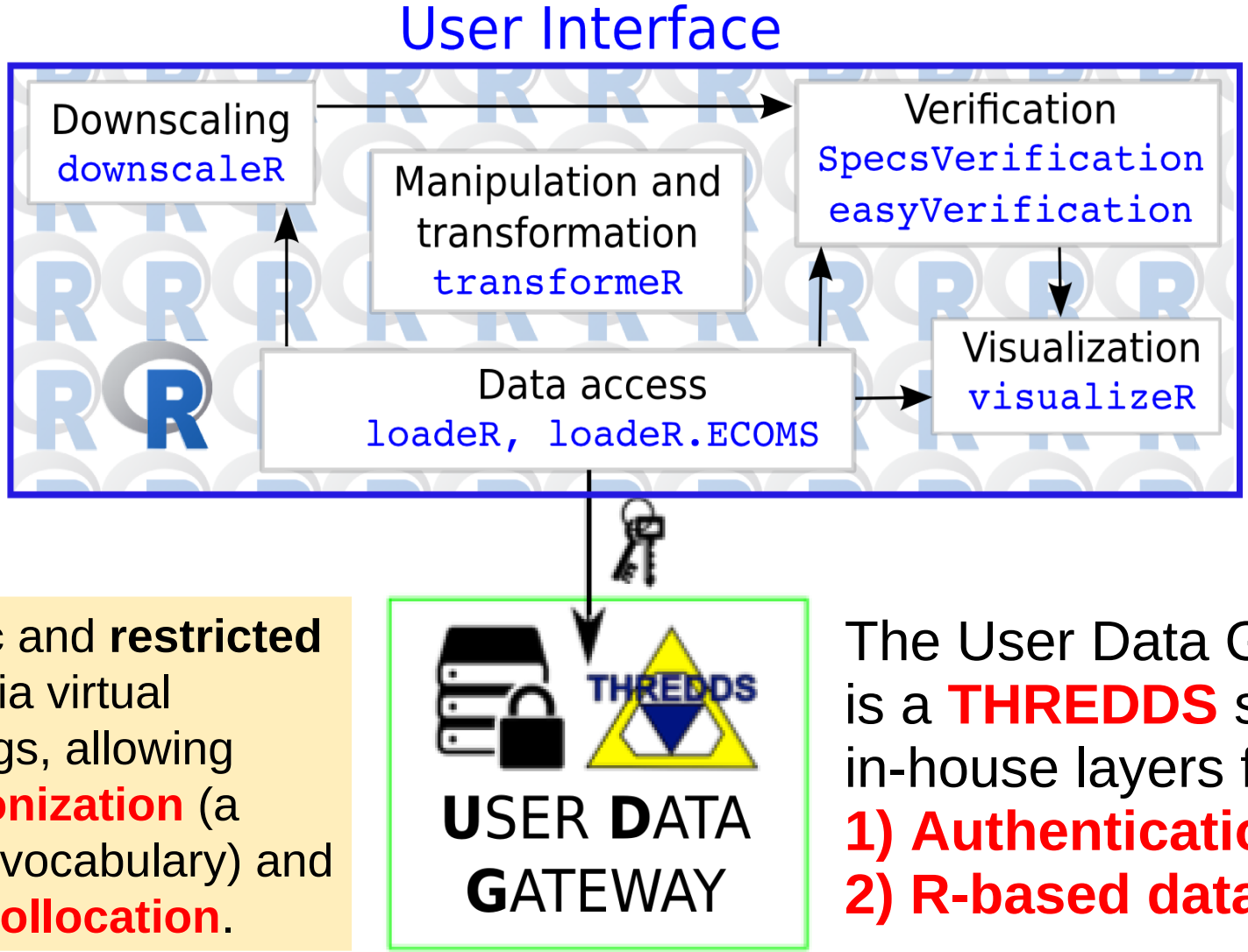
4. Key links

EUPORIAS

SPECS

2012-2016

An R-based integrated framework for (remotely) accessing and processing climate data



Existing **visualization**, **validation** and **downscaling** packages are transparently linked to UDG using **common data structures**.

Public and **restricted data** via virtual catalogs, allowing **harmonization** (a single vocabulary) and **data collocation**.

The User Data Gateway (UDG) is a **THREDDS** server with two in-house layers for:

1) Authentication

2) R-based data access.

Contents

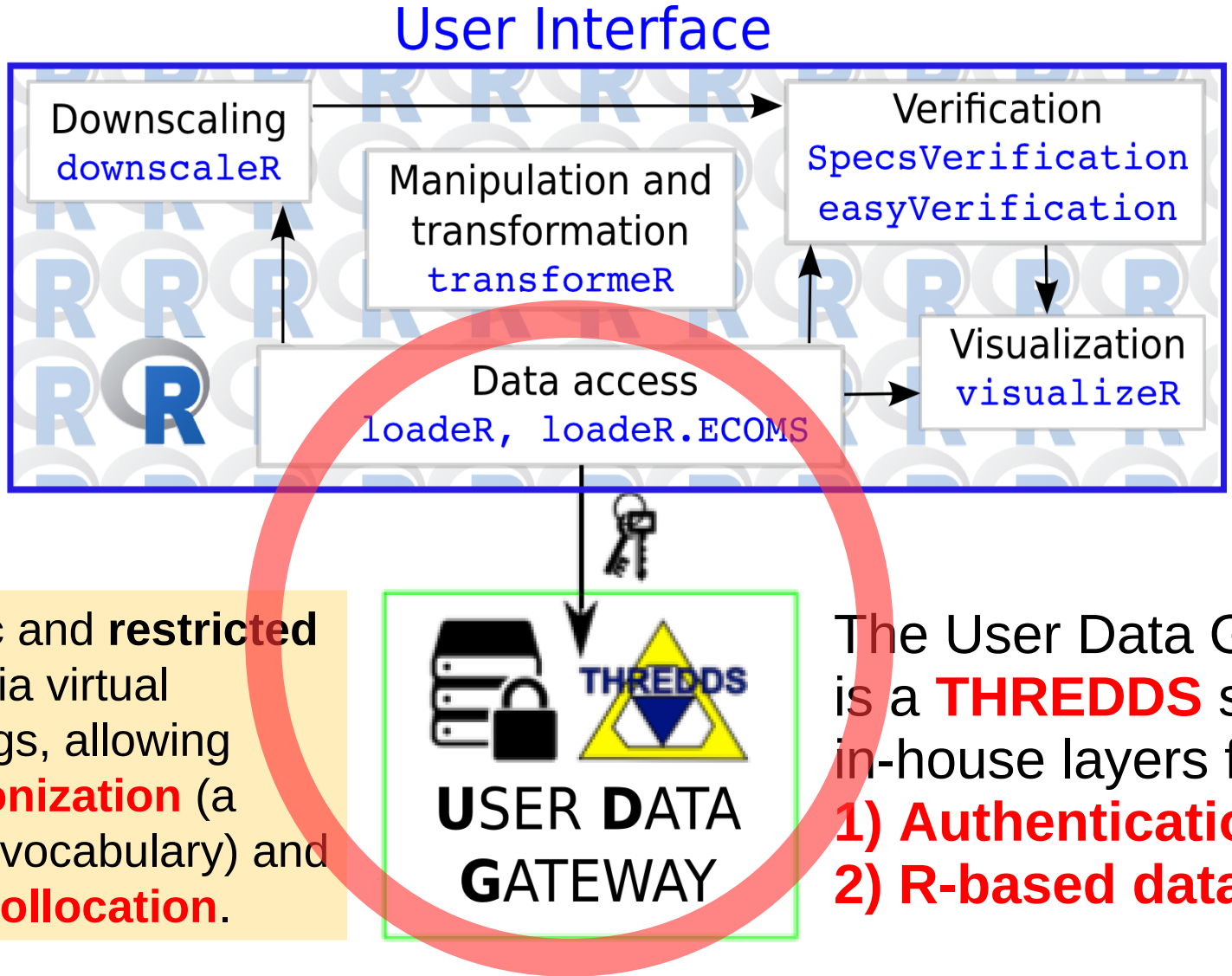
1. **The ECOMS User Data Gateway (UDG)**
   - **Harmonized access to virtual datasets**
   - Transparent access using R: examples (`loadeR.ECOMS`)
2. R package `downscaleR` for downscaling
   - Worked bias correction example
   - Worked downscaling example
3. Integration with other R tools
   - Verification (`easyVerification`)
   - Forecast skill visualization (`visualizeR`)
4. Key links

EUPORIAS

SPECS

2012-2016

An R-based integrated framework for (remotely) accessing and processing climate data



Existing **visualization**, **validation** and **downscaling** packages are transparently linked to UDG using **common data structures**.

Public and **restricted data** via virtual catalogs, allowing **harmonization** (a single vocabulary) and **data collocation**.

The User Data Gateway (UDG) is a **THREDDS** server with two in-house layers for:

1) Authentication

2) R-based data access.

ECOMS-UDG datasets

ECOMS-UDG provides **harmonized** access to daily data.

- **observations** (WFDEI),
- **reanalysis** (NCEP-R1, ERA-Interim)
- **seasonal forecasting**, including hindcasts from state-of-the-art models: ECMWF-System4, NCEP-CFSv2, UKMO-Glosea5.

User-tailored design (SPECS and EUPORIAS) including the variables typically needed for impact studies, mostly at **surface** level: **precip, temp, wind speed, humidity, radiations, SLP**, but also **upper-air** information at 1000,850,700,500,300,200 mb (for statistical downscaling).

→ Link to available variables and datasets

ECOMS-UDG provides **harmonized** access to locally stored daily data.

https://meteo.unican.es/trac/wiki/udg/ecoms/dataserver/catalog

https://meteo.unican.es/trac/wiki/udg/ecoms/dataserver/catalog

Search

Wiki | Blog | Timeline | View Tickets | Search

wiki: udg / ecoms / dataserver / catalog

Up | Start Page | Index | History | Restore Form

The **ECOMS UDG** collects and provides information (mainly at 6-hourly and/or daily resolution) for a reduced number of variables from a number of datasets (seasonal hindcasts, reanalysis and observations) obtained from different data providers. The following list of variables has been identified according to the user's needs, receiving feedback from EUPORIAS WP22 (climate information indices, CIIs), WP23 (impact models), WP21 (calibration and downscaling) and SPECS WP61 (pilot applications) and WP52 (calibration and downscaling). See the section on the assessment of user's needs for more details.

Note that the **R names** below correspond to the vocabulary names used in the R data access package for homogenization purposes. Note that, data homogenization and aggregation (i.e. daily means from 6h data) is only provided through the R data access package.

In order to specify the particular **temporal frequency/aggregation** available for the variables in the different datasets, the following codes are used in the table below: **6h** (6-hourly instantaneous data). **12h** (12-hourly instantaneous data). **24h** (24-hourly instantaneous data). **DM** (daily mean value). **DX** (daily maximum value). **DN** (daily minimum value). **DA** (daily accumulated data). **DAr** (accumulated since the initialization time – `runtime`) NOTE: The R package performs deaccumulation on a daily basis to match the standard definition. **fx** (static field)

In the table below, boldface codes (e.g. **6h**) indicate variables already available through the ECOMS UDG. ***Italics*** are used for work in progress (variables to be included in the next update). **e** indicates that a variable exists in the original dataset but it is not planned to be included yet in ECOMS-UDG; **blanks** indicate that the variables do not exist in the original dataset. Codes ended by **(*)** indicate variables which do NOT exist in the dataset, but are derived/approximated from other available ones through the R data access package. For more details on the particular approximations used see the conversion formulae. Variables ended by **(#)** indicate daily aggregated values obtained from the corresponding original 3-hourly data.

		Observations:	Reanalysis:		Seasonal forecasting models:						
R name	**Variable description**	WFDEI	NCEP_reanalysis1	ERA_interim	System4_seasonal_15	System4_seasonal_51	System4_annual_15	CFSv2_seasonal	SMHI-EC-EARTH_EUPORIAS	Glosea5_seasonal_12	Glosea5_seasonal_24
					Surface variables						
tas	Near-Surface air temperature	**DM**	**6h**	**DM**	**6h/DM**	**DM**		**6h**		**DM**	**DM**
tasmax	Daily Maximum Near-Surface Air Temperature	**DX(#)**	**6h**	**DX**	**DX**	**DX**	**DX**	**6h**		**DX**	**DX**
tasmin	Daily Minimum Near-Surface Air Temperature	**DN(#)**	**6h**	**DN**	**DN**	**DN**	**DN**	**6h**		**DN**	**DN**
tp	Total precipitation amount	**DA**	**6hA**	**DA**	**DAr**	**DAr**	**DAr**	**6h**		**DA**	**DA**
psl	Sea Level Pressure		**6h**	**DM**	**6h**	**6h**	**12h**	**6h**		**DM**	**DM**
ps	Surface air pressure	**DM**		e	**6h(*)**			**6h**			
wss	Wind speed (at 10m)	**DM**		e	**6h(*)**	e	e	**6h(*)**			
tdps	2m Dewpoint Temperature			e	**6h**	e					
huss	Surface (2m) specific humidity	**DM**	**6h**	e	**6h(*)**			**6h**			

Contents

1. The ECOMS User Data Gateway (UDG)
   - Harmonized access to virtual datasets
   - **Transparent access using R: examples (`loadeR.ECOMS`)**

2. R package `downscaleR` for downscaling
   - Worked bias correction example
   - Worked downscaling example

3. Integration with other R tools
   - Verification (`easyVerification`)
   - Forecast skill visualization (`visualizeR`)

4. Key links

EUPORIAS

SPECS

2012-2016

loadeR

This allows creating a one-stop entry point for datasets, aggregating multiple files from the same (or different) resources.

Further details and worked examples:
→ https://github.com/SantanderMetGroup/loadeR/wiki

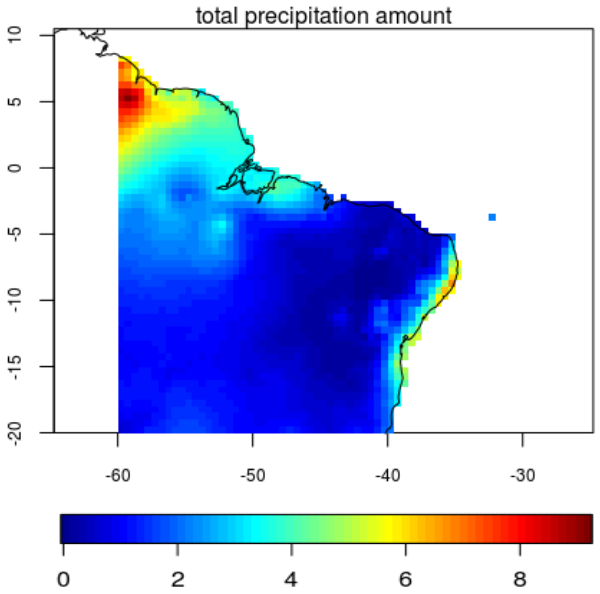
LoadeR.ECOMS

Is the an extended version of loadeR to access data from the ECOMS-UDG.

Further details and worked examples:
→ https://meteo.unican.es/trac/wiki/udg/ecoms/RPackage

Defining obs/reanalysis data chunk

```
library(loadeR.ECOMS)
loginUDG(username = 'jDoe', password = '******')
wfdei <- loadECOMS(dataset = "WFDEI",
                   var = "tp",
                   lonLim =  c(-60,-30),
                   latLim = c(-20,10),
                   season = 7:9,
                   years = 1995:2009)
```



Define verification times
Season: JAS
Period: 1995-2009

Any other OpeNDAP server can be accessed with the loadeR R package .

However, no harmonization will be available and a knowledge of the dataset (name of variables, etc.) will be required.

Defining a prediction data chunk

Seasonal Forecast (prediction)

Initialization times ("runtimes")

Verification times ("forecast times")

seasons

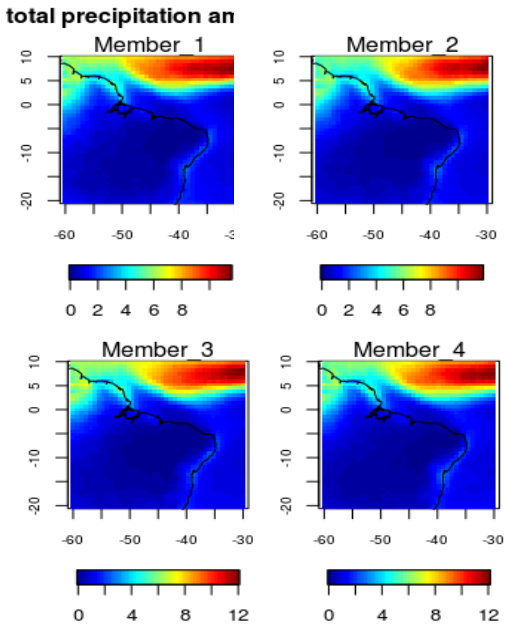
Target → season

```
cfs <- loadECOMS(dataset = 'CFSv2_seasonal',
                 var = 'tp',
                 lonLim =  c(-20, 10),
                 latLim = c(2,35),
                 members = 1:4,
                 leadMonth = 1,
                 season = 7:9,
                 years = 1995:2009)
```

Define runtime
May
Initializations

Define members
First 4

Define verification times
Season: JAS
Period: 1995-2009



The ECOMS-UDG wiki:

https://meteo.unican.es/trac/wiki/udg/ecoms/dataserver

User registration

Available datasets and variables

Exploration via Web

APIs for Data Access

The loadeR.ECOMS wiki:

https://meteo.unican.es/trac/wiki/udg/ecoms/RPackage

Installation and Versions

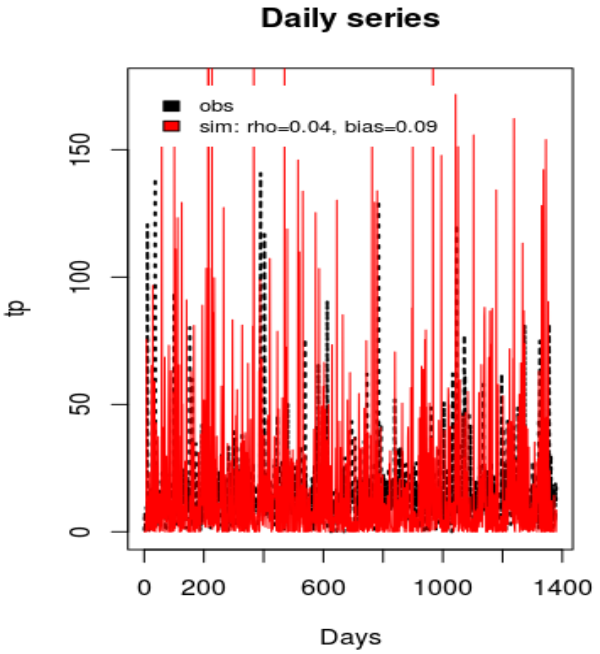
Authentication

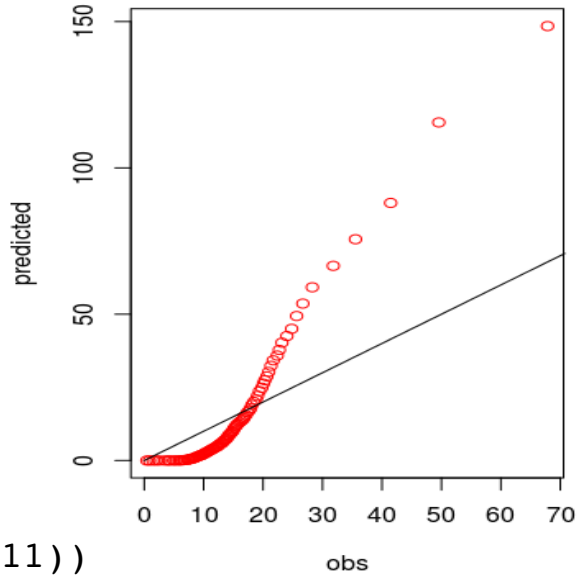
Data Homogeneization

Examples

Data loading...

```
library(loadeR.ECOMS)
loginUDG("username", "password")

#OBSERVATIONS
wfdei <- loadECOMS(dataset = "WFDEI",
                   var = "tp",
                   lonLim = c(-18, -9),
                   latLim = c(9, 16),
                   season = 7:9,
                   time = "DD",
                   aggr.d = "sum",
                   Years = 1995:2009)
#SEASONAL FORECAST
cfs <- loadECOMS(dataset = "CFSv2_seasonal",
                 var = "tp",
                 lonLim = c(-18, -9),
                 latLim = c(9, 16),
                 Season = 7:9
                 time = "DD",
                 aggr.d = "sum",
                 years = 1995:2009,
                 leadMonth = 1,
                 members = 1:5)

library(downscaleR)
quickDiagnostics(wfdei, cfs , members = 1, location = c(-15,11))
```





Contents

1. The ECOMS User Data Gateway (UDG)
   - Harmonized access to virtual datasets
   - Transparent access using R: examples (`loadeR.ECOMS`)

2. **R package `downscaleR` for downscaling**
   - Worked bias correction example
   - Worked downscaling example

3. Integration with other R tools
   - Verification (`easyVerification`)
   - Forecast skill visualization (`visualizeR`)

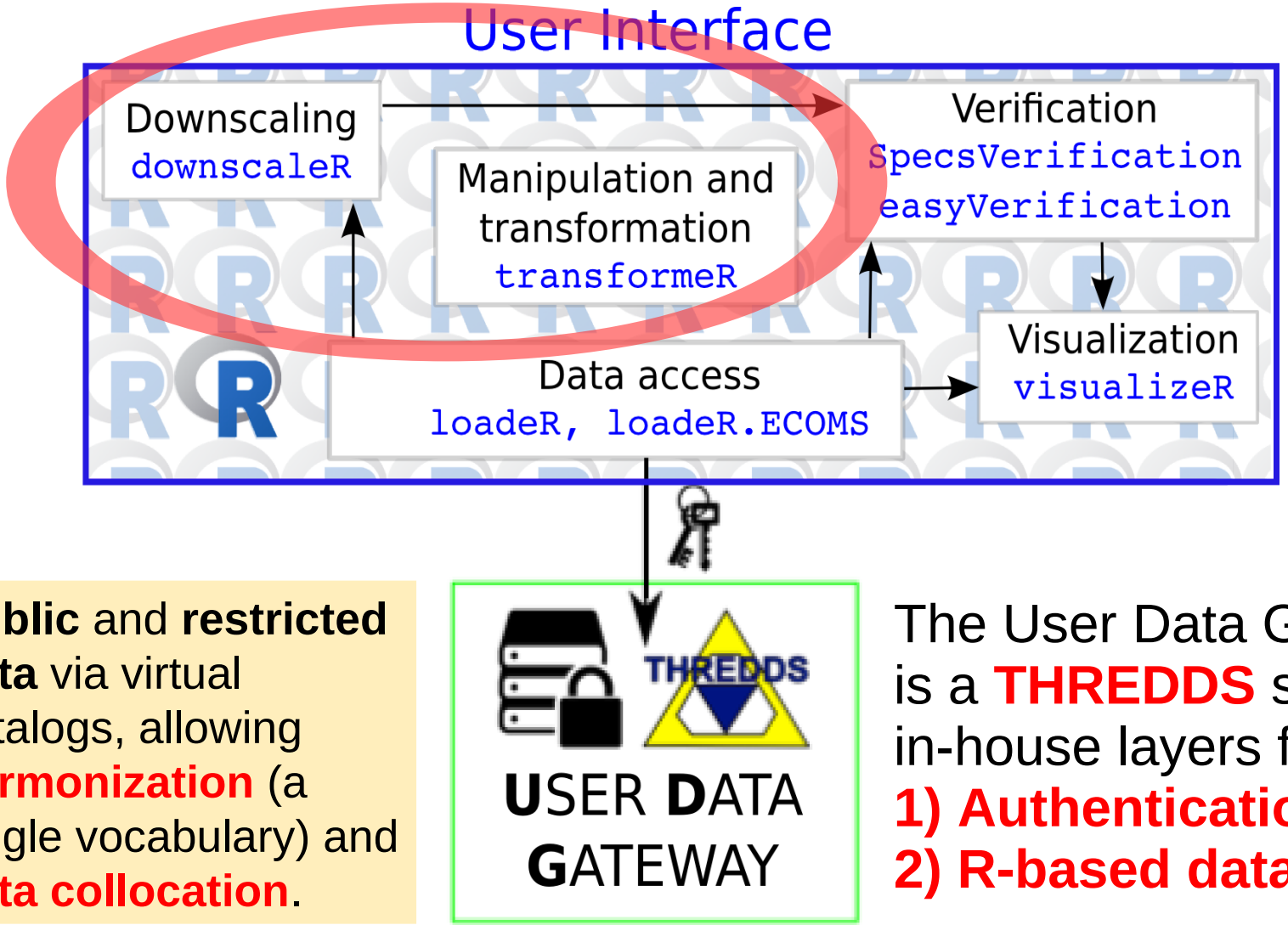
4. Key links

EUPORIAS

SPECS

2012-2016

An R-based integrated framework for (remotely) accessing and processing climate data in the era of climate services



Existing **visualization**, **validation** and **downscaling** packages are transparently linked to UDG using **common data structures**.

Public and **restricted data** via virtual catalogs, allowing **harmonization** (a single vocabulary) and **data collocation**.

The User Data Gateway (UDG) is a **THREDDS** server with two in-house layers for:

1) Authentication

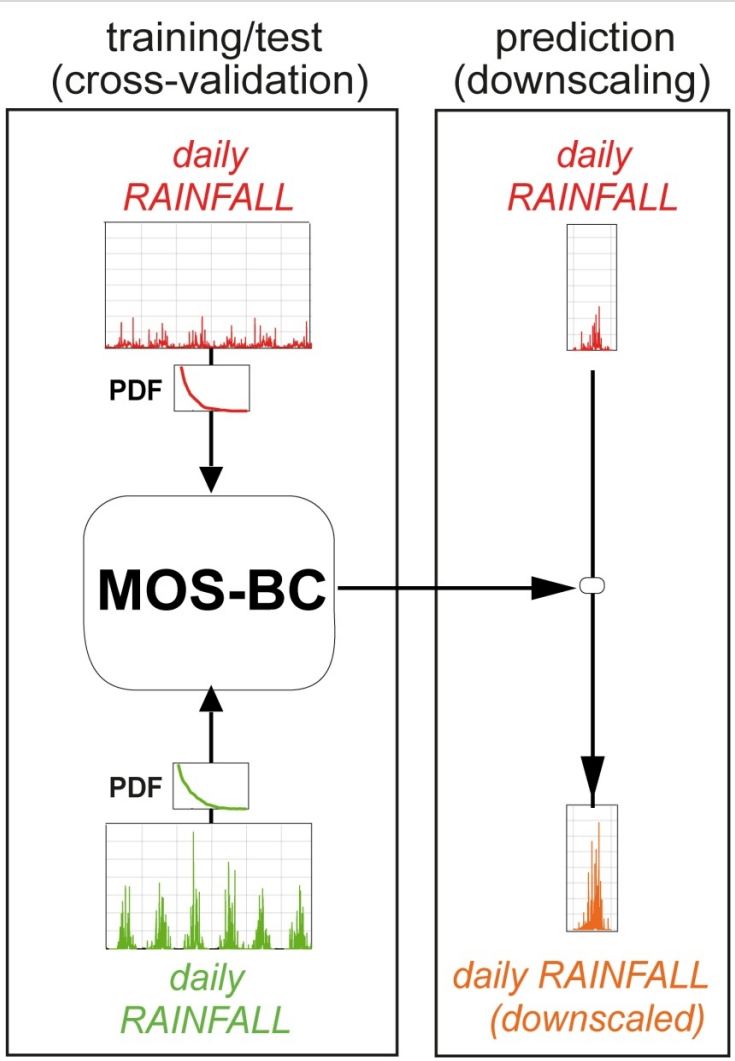
2) R-based data access.

- `transformeR` extends the `loadeR` and `loadeR.ECOMS` capabilities for **data manipulation**
  - Regridding/interpolation, subsetting and aggregation
  - PCA/EOF analysis

→ https://github.com/SantanderMetGroup/transformeR

- `downscaleR` has been designed to work with daily data (seasonal predictions, multidecadal pro*j*ections).
  - **Bias correction**/adjustment (including cross-validation): (Local) Scaling, qq-mapping (various forms), parametric.
  - **Perfect-prog downscaling** (including cross-validation): Analogs, regression (linear and generalized linear)

→ https://github.com/SantanderMetGroup/downscaleR

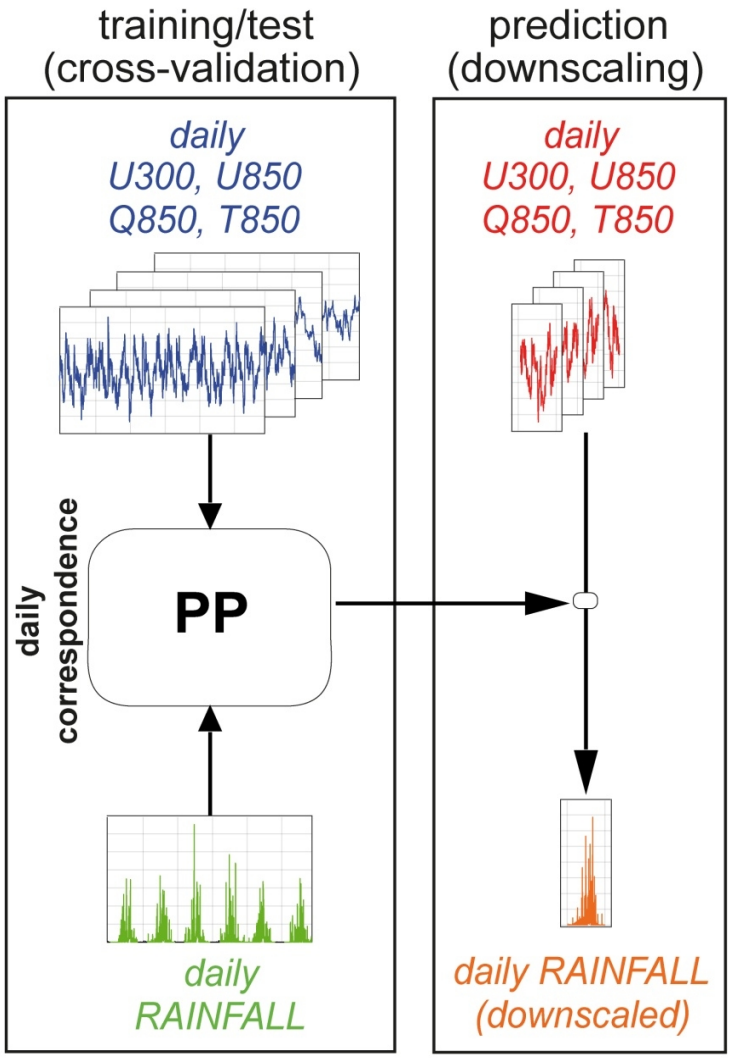


Bias Correction (BC)

vs

Perfect Prognosis (PP)

The PP method can improve the SKILL of the forecast target variable (daily rainfall)



	rainfall	large-scale
Reanalysis		
Seasonal predictions	X	
Observations (local)	X	

	rainfall	large-scale
Reanalysis		X
Seasonal predictions		X
Observations (local)	X	

Contents

EUPORIAS

SPECS

2012-2016

Bias Correction...

```
?biasCorrection


#method local scaling
cal <- biasCorrection(y = wfdei,
                      x = cfs,
                      newdata = cfs,
                      precipitation = TRUE,
                      method = "scaling",
                      scaling.type = "multiplicative")
quickDiagnostics(wfdei, cfs, cal, members = 1, location = c(-15, 11))
```

Bias Correction...

```
library(downscaleR)
?biasCorrection


#method local scaling
cal <- biasCorrection(y = wfdei,
                      x = cfs,
                      newdata = cfs,
                      precipitation = TRUE,
                      method = "scaling",
                      scaling.type = "multiplicative")
quickDiagnostics(wfdei, cfs, cal, members = 1, location = c(-15, 11))

#method eqm

cal <- biasCorrection(y = wfdei,
                      x = cfs,
                      newdata = cfs,
                      precipitation = TRUE,
                      method = "eqm",
                      n.quantiles = 100,
                      wet.threshold = 0,
                      extrapolation = "constant")

quickDiagnostics(wfdei, cfs,  members = 1, cal, location = c(-15, 11))
```

Bias Correction...

```
...

#method eqm

cal <- biasCorrection(y = wfdei,
                      x = cfs,
                      newdata = cfs,
                      precipitation = TRUE,
                      method = "eqm",
                      n.quantiles = 100,
                      wet.threshold = 0,
                      extrapolation = "constant")

quickDiagnostics(wfdei, cfs,  members = 1, cal, location = c(-15, 11))

cfs2 <- interpGrid(cfs, getGrid(wfdei))
quickDiagnostics(wfdei, cfs2,  members = 1,  location = c(-15, 11), type = "interannual")
```

Bias Correction...

```
...

#method eqm

cal <- biasCorrection(y = wfdei,
                      x = cfs,
                      newdata = cfs,
                      precipitation = TRUE,
                      method = "eqm",
                      n.quantiles = 100,
                      wet.threshold = 0,
                      extrapolation = "constant")

quickDiagnostics(wfdei, cfs,  members = 1, cal, location = c(-15, 11))

cfs2 <- interpGrid(cfs, getGrid(wfdei))
quickDiagnostics(wfdei, cfs2,  members = 1,  location = c(-15, 11), type = "interannual")

quickDiagnostics(wfdei, cfs, cal,  members = 1,  location = c(-15, 11), type = "interannual")
```

Bias Correction...applying cross validation...

```
cfs1mem <- subsetGrid(cfs, members = 1)
cal1mem <- subsetGrid(cal, members = 1)

cal.cross <- biasCorrection(y = wfdei,
                            x = cfs1mem,
                            newdata = cfs1mem,
                            precipitation = TRUE,
                            method = "eqm",
                            wet.threshold = 0,
                            cross.val = "loocv")

quickDiagnostics(wfdei, cfs1mem, cal.cross, members = 15, location = c(-15, 11))
```

Bias Correction...applying a moving window...

```
?biasCorrection
cal.win <- biasCorrection(y = wfdei,
                          x = cfs1mem,
                          newdata = cfs1mem,
                          precipitation = TRUE,
                          method = "eqm",
                          wet.threshold = 0,
                          window = c(30, 20))

quickDiagnostics(wfdei, cfs1mem, cal.win, location = c(-15, 11))
```

Bias Correction...of a non-observed period...

```
y <- subsetGrid(wfdei, years = 1995:2004)
x <- subsetGrid(cfs, years = 1995:2004)
newdata <- subsetGrid(cfs, years = 2005:2009)

cal2 <- biasCorrection(y = y, x = x, newdata = newdata,
                       precipitation = TRUE,
                       method = "eqm",
                       wet.threshold = 0.01)
quickDiagnostics(wfdei, cfs, cal2, members = 1, location = c(-15, 11))
```

Contents

EUPORIAS

SPECS

2012-2016

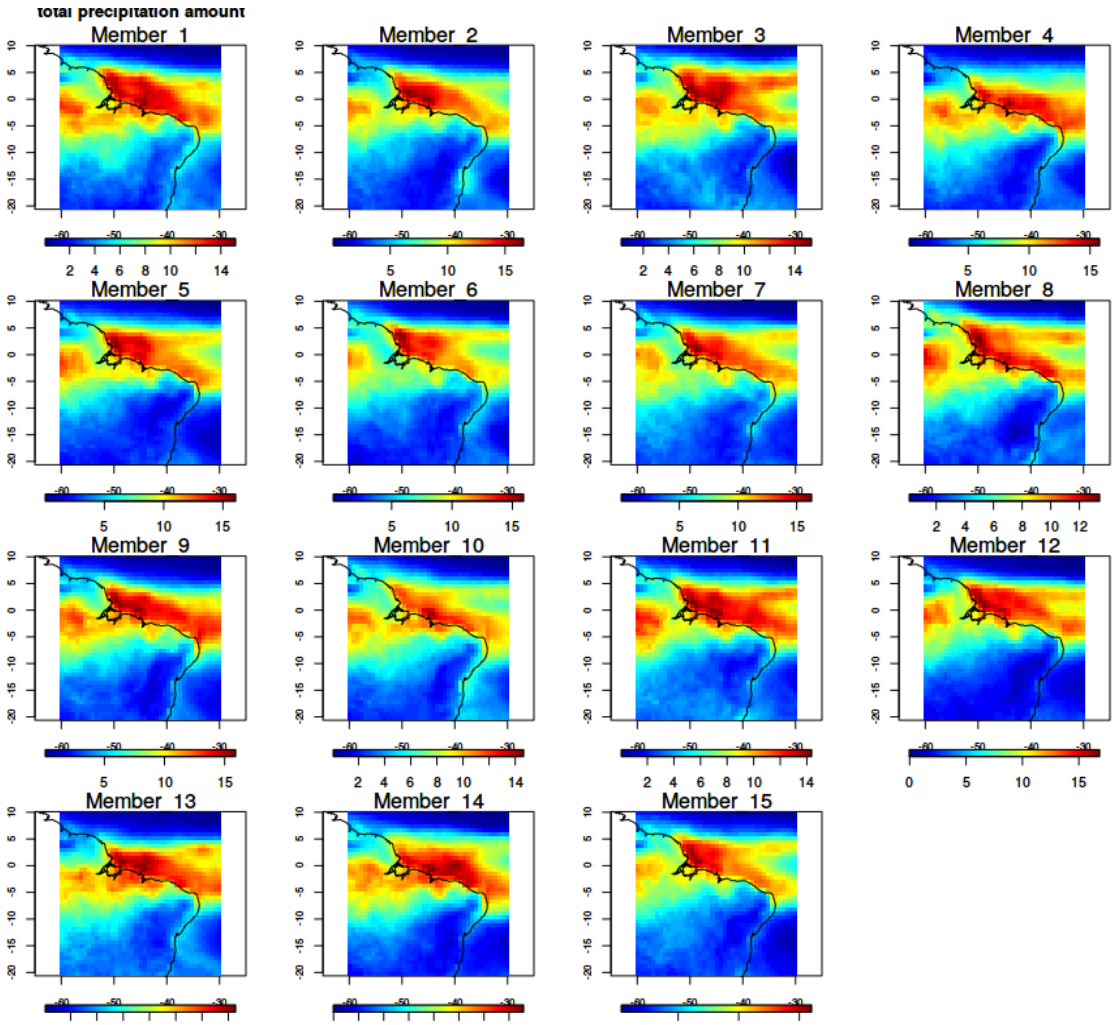
Downscaling of System4 MAM precipitation forecast over NE Brazil (January initialization, 15 members)

SPECS D52.2 http://www.specs-fp7.eu/wiki/images/d/d0/SPECS_D52.2.pdf

A relatively complex task usually involving many intermediate steps from data loading to analysis of the results...

… made EASY (10 commands)

… and **fully** REPRODUCIBLE



Data loading...

```
# target variable
predictand <- "tp"
wfdei.tp <- loadECOMS(dataset = "WFDEI" , var = predictand,
                      lonLim = c(-60,-30), latLim = c(-20,10),
                      season = 3:5, years = 1981:2010)

predictor <- c("psl", "ta@850", "hus@850")
# Loading NCEP (Predictors)
NCEP.psl <- loadECOMS(dataset = "NCEP" , var = predictor[1],
                      lonLim = c(-60,-30), latLim = c(-20,10),
                      season = 3:5, years = 1981:2010)
# Same for ta@850 and hus@850
# Loading System4 predictions (Predictors)
S4 <- loadECOMS(dataset = "System4_seasonal_15" ,predictor[1],
                lonLim = c(-60,-30), latLim = c(-20,10),
                season = 3:5, years = 1981:2010,
                leadMonth = 2)
# Same for ta@850 and hus@850 (and also tp for a posterior verification).
```

… data preprocessing ....

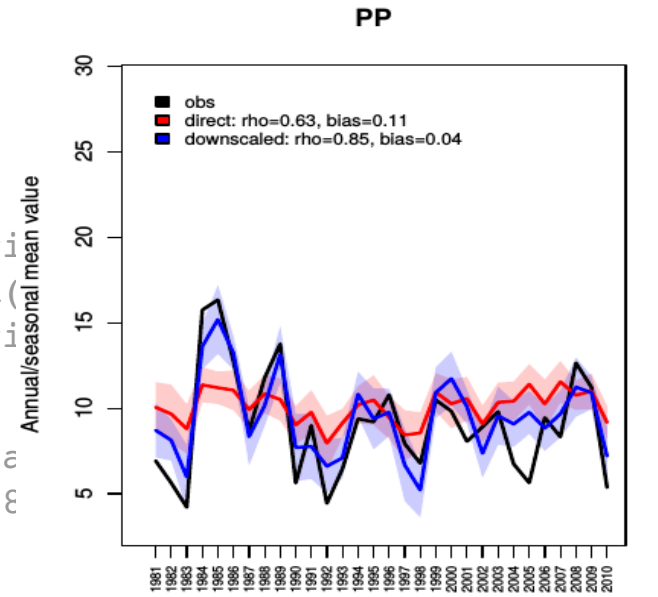
```
...
predictor <- c("psl", "ta@850", "hus@850")
# Loading NCEP (Predictors)
NCEP.psl <- loadECOMS(dataset = "NCEP" , var = predictor[1],
                      lonLim = c(-60,-30), latLim = c(-20,10),
                      season = 3:5, years = 1981:2010)
# Same for ta@850 and hus@850
# Loading System4 predictions (Predictand, precip)
S4.psl <- loadECOMS(dataset = "System4_seasonal_15" , var = 'predictor[1]',
                lonLim = c(-60,-30), latLim = c(-20,10),
                season = 3:5, years = 1981:2010,
                leadMonth = 2)
# Same for ta@850 and hus@850 (and also tp for a posterior verification).
# Interpolating S4 to the NCEP grid, and rescaling
S4.psl <- interpGrid(S4.psl, new.coordinates = getGrid(NCEP.psl))
S4.ta <- interpGrid(S4.ta, new.coordinates = getGrid(NCEP.psl))
S4.hus <- interpGrid(S4.hus, new.coordinates = getGrid(NCEP.psl))


# Predictor datasets for reanalysis and forecast data
NCEP <- makeMultiGrid(NCEP.psl, NCEP.ta850, NCEP.hus850)
S4 <- makeMultiGrid(S4.psl, S4.ta850, S4.hus850)

# Computing EOFs and PCs for reanalysis
ncep.eof <- computeEOF(NCEP, n.eofs = 15)
S4.sc <- rescaleMonthlyMeans(pred = NCEP, sim = S4)
```

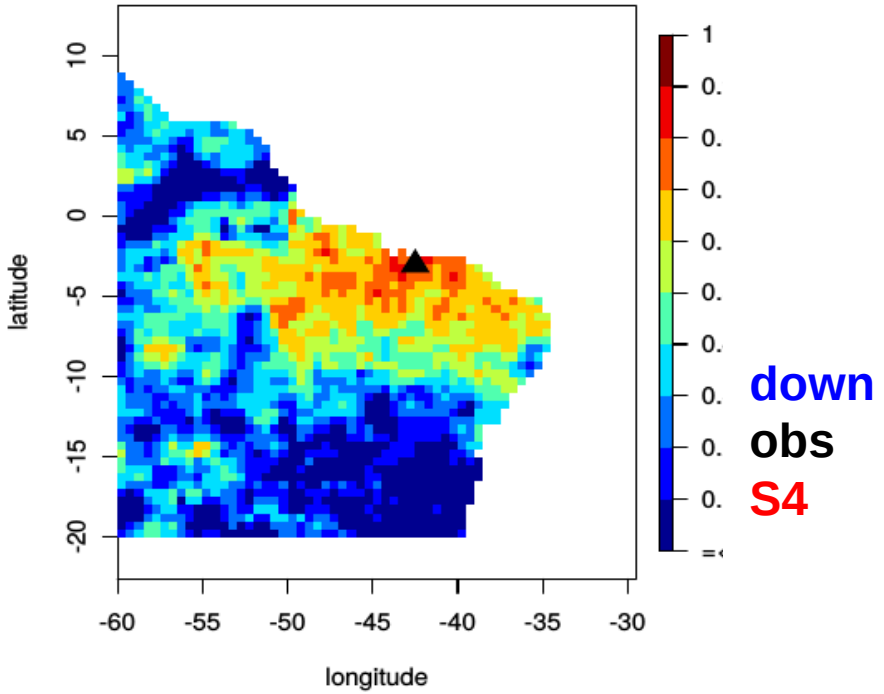
… downscaling and cross-validation ...

```
...
# Interpolating S4 to the NCEP grid, and rescaling
S4.psl <- interpGrid(S4.psl, new.coordinates = getGrid(NCEP.psl))
S4.ta <- interpGrid(S4.ta, new.coordinates = getGrid(NCEP.psl))
S4.hus <- interpGrid(S4.hus, new.coordinates = getGrid(NCEP.psl))


# Predictor datasets for reanalysis and forecast data
NCEP <- makeMultiGrid(NCEP.psl, NCEP.ta850, NCEP.hus850)
S4 <- makeMultiGrid(S4.psl, S4.ta850, S4.hus850)

# Computing EOFs and PCs for reanalysis
ncep.eof <- computeEOF(NCEP, n.eofs = 15)
S4.sc <- rescaleMonthlyMeans(pred = NCEP, sim = S4)

# Downscaling with Generalized Linear Models
down <- downscale(obs = wfdei.tp, pred = ncep.eof,
                  sim = S4.sc,
                  method = "glm",
                  n.pcs = 15,
                  parallel = TRUE)
```

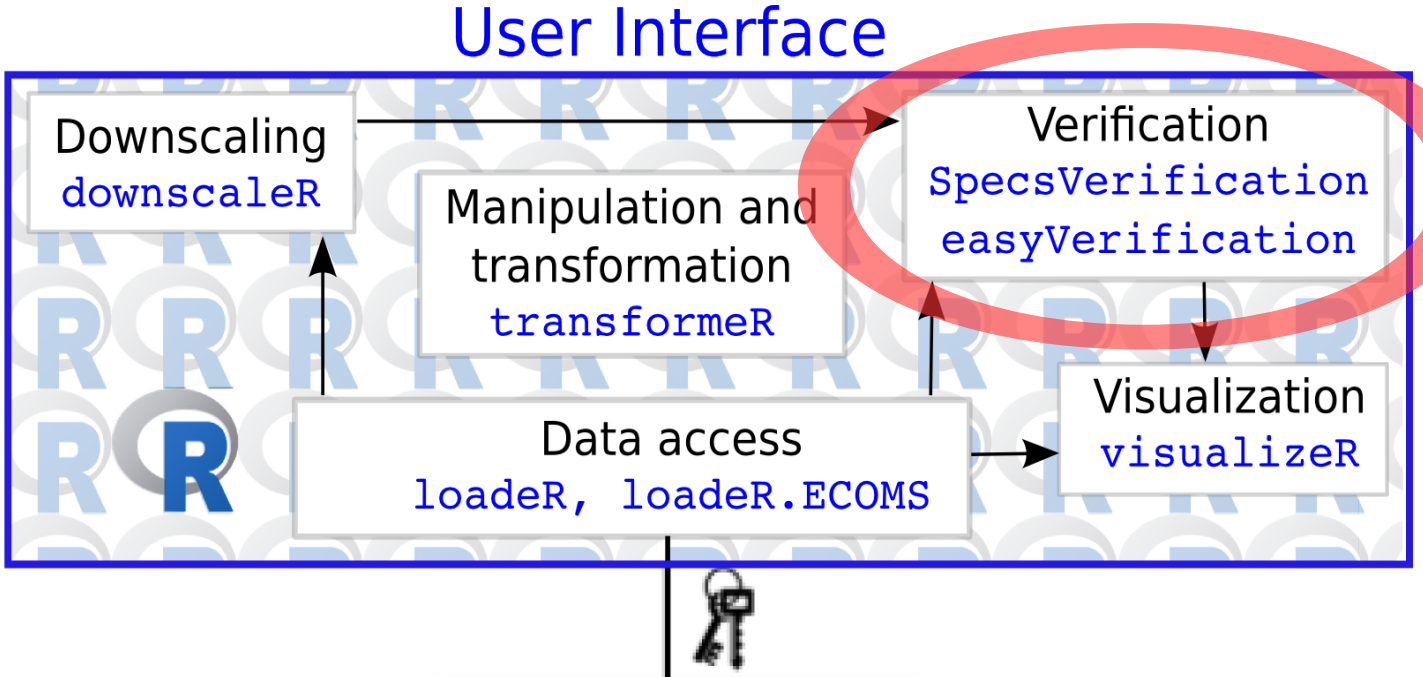
… and verification and visualization.

```
...
# Interpolating S4 to the NCEP grid, and rescaling
S4.psl <- interpGrid(S4.psl, new.coordinates = getGri
S4.ta <- interpGrid(S4.ta, new.coordinates = getGrid(
S4.hus <- interpGrid(S4.hus, new.coordinates = getGri


# Predictor datasets for reanalysis and forecast data
NCEP <- makeMultiGrid(NCEP.psl, NCEP.ta850, NCEP.hus8
S4 <- makeMultiGrid(S4.psl, S4.ta850, S4.hus850)

# Computing EOFs and PCs for reanalysis
ncep.eof <- computeEOF(NCEP, n.eofs = 15)
S4.sc <- rescaleMonthlyMeans(pred = NCEP, sim = S4)

# Downscaling with Generalized Linear Models
down <- downscale(obs = wfdei.tp, pred = ncep.eof,
                  sim = S4.sc,
                  method = "glm",
                  n.pcs = 15,
                  parallel = TRUE)
# Analysis of results
quickDiagnostics(wfdei.tp, S4.tp, down,
                 type = "interannual")
```





An R-based integrated framework for (remotely) accessing and processing climate data in the era of climate services



Existing **visualization**, **validation** and **downscaling** packages are transparently linked to UDG using **common data structures**.

Public and **restricted data** via virtual catalogs, allowing **harmonization** (a single vocabulary) and **data collocation**.

The User Data Gateway (UDG) is a **THREDDS** server with two in-house layers for:

1) Authentication

2) R-based data access.

`easyVerification` package. Common validation scores for seasonal forecasting.

Available documentation with worked examples:
http://www.meteo.unican.es/work/downscaler/wiki/docs/ecoms_bias_correction.pdf

EUPORIAS

Verification of seasonal forecasts from the ECOMS User Data Gateway: a worked example

Joaquín Bedia[1] and Jonas Bhend[2]

[1]Santander Met Group. University of Cantabria - CSIC (Spain)
[2]Federal Office of Meteorology and Climatology MeteoSwiss, Zurich, Switzerland

version 2 - '2015-05-18'

Annual aggregation of daily data ...

```
mean_bc <- aggregateGrid(grid = cal,
                         aggr.y = list(FUN = "mean", na.rm = TRUE))
mean_obs <- aggregateGrid(grid = wfdei,
                          aggr.y = list(FUN = "mean", na.rm = TRUE))
```

... detrending ...

```
mean_bc <- aggregateGrid(grid = cal,
                         aggr.y = list(FUN = "mean", na.rm = TRUE))
mean_obs <- aggregateGrid(grid = wfdei,
                          aggr.y = list(FUN = "mean", na.rm = TRUE))
pred <- detrendGrid(mean_bc)
obs <- detrendGrid(mean_obs)
```

... ROC Skill score with `easyVerification` ...

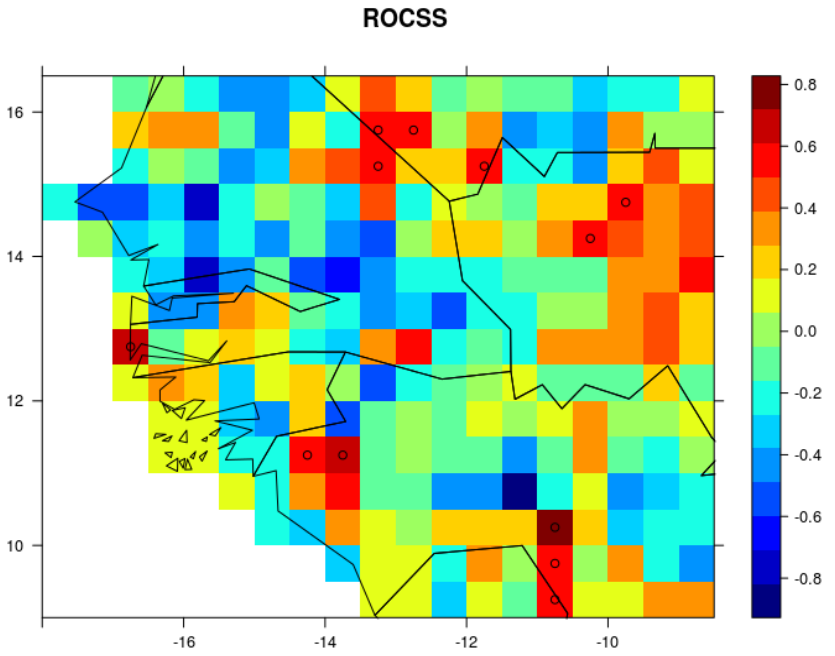
```
mean_bc <- aggregateGrid(grid = cal,
                         aggr.y = list(FUN = "mean", na.rm = TRUE))
mean_obs <- aggregateGrid(grid = wfdei,
                          aggr.y = list(FUN = "mean", na.rm = TRUE))
pred <- detrendGrid(mean_bc)
obs <- detrendGrid(mean_obs)
rocss <- easyVerification::veriApply("EnsRocss",
                                     fcst = pred[["Data"]],
                                     obs = obs[["Data"]],
                                     prob = 2/3, ensdim = 1, tdim = 2)
```

… return to the loadeR – downscaleR grid format

```
mean_bc <- aggregateGrid(grid = cal,
                         aggr.y = list(FUN = "mean", na.rm = TRUE))
mean_obs <- aggregateGrid(grid = wfdei,
                          aggr.y = list(FUN = "mean", na.rm = TRUE))
pred <- detrendGrid(mean_bc)
obs <- detrendGrid(mean_obs)
rocss <- easyVerification::veriApply("EnsRocss",
                                     fcst = pred[["Data"]],
                                     obs = obs[["Data"]],
                                     prob = 2/3, ensdim = 1, tdim = 2)
upper.tercile <- easyVeri2grid(easyVeri.mat = rocss$cat2,
                               obs.grid = obs)
```

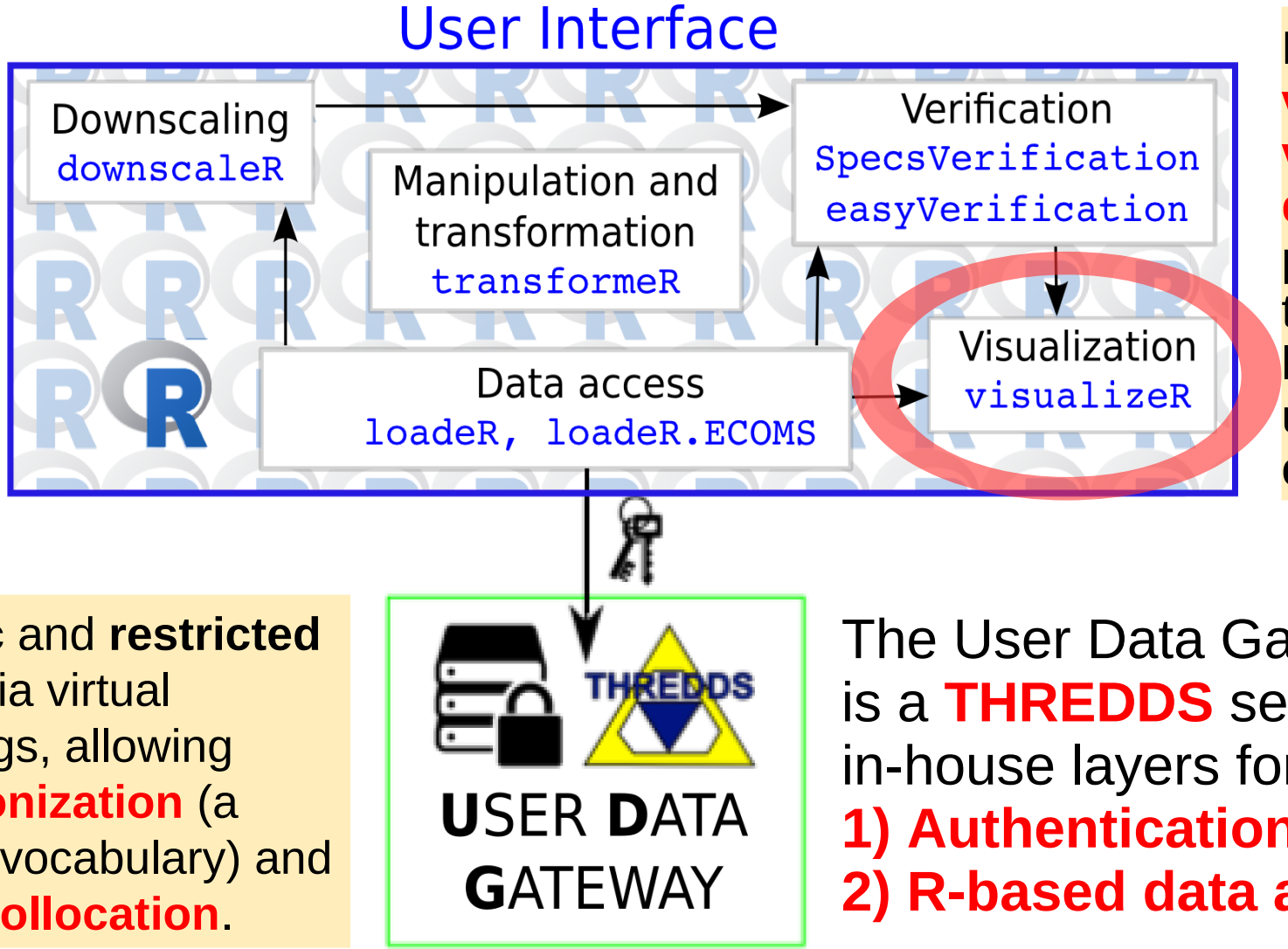
… and plot the resulting ROCSS map

```
mean_bc <- aggregateGrid(grid = cal,
                         aggr.y = list(FUN = "mean", na.rm = TRUE))
mean_obs <- aggregateGrid(grid = wfdei,
                          aggr.y = list(FUN = "mean", na.rm = TRUE))
pred <- detrendGrid(mean_bc)
obs <- detrendGrid(mean_obs)
rocss <- easyVerification::veriApply("EnsRocss",
                                     fcst = pred[["Data"]],
                                     obs = obs[["Data"]],
                                     prob = 2/3, ensdim = 1, tdim = 2)
upper.tercile <- easyVeri2grid(easyVeri.mat = rocss$cat2,
                               obs.grid = obs)
plotClimatology(upper.tercile,
                scales = list(draw = TRUE),
                backdrop.theme = "countries"
                main = "ROCSS")
```



Contents

An R-based integrated framework for (remotely) accessing and processing climate data in the era of climate services



Existing **visualization**, **validation** and **downscaling** packages are transparently linked to UDG using **common data structures**.

Public and **restricted data** via virtual catalogs, allowing **harmonization** (a single vocabulary) and **data collocation**.

The User Data Gateway (UDG) is a **THREDDS** server with two in-house layers for:

1) Authentication

2) R-based data access.

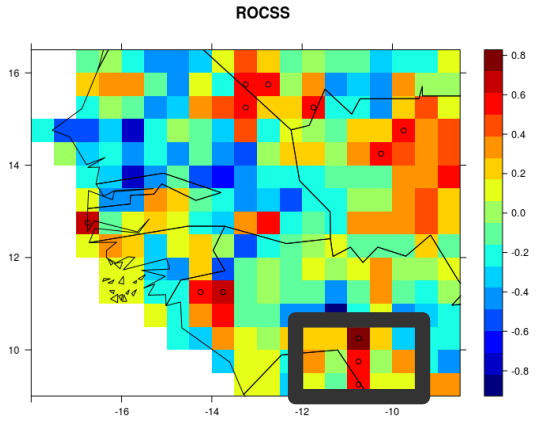
`visualizeR` package. Special verification plots for forecast skill visualization.

Documentation and worked examples in the wiki:
https://github.com/SantanderMetGroup/visualizeR/wiki

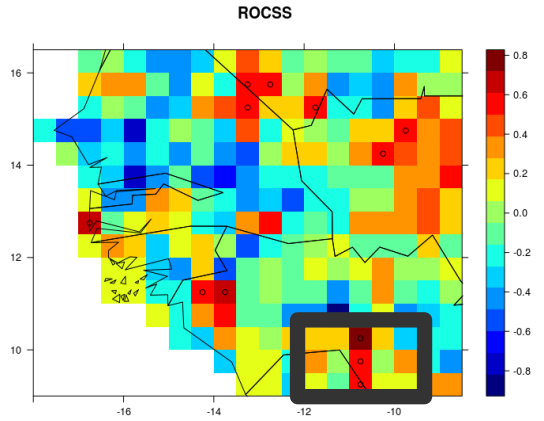
EUPORIAS

Spatial subset of a region of interest...

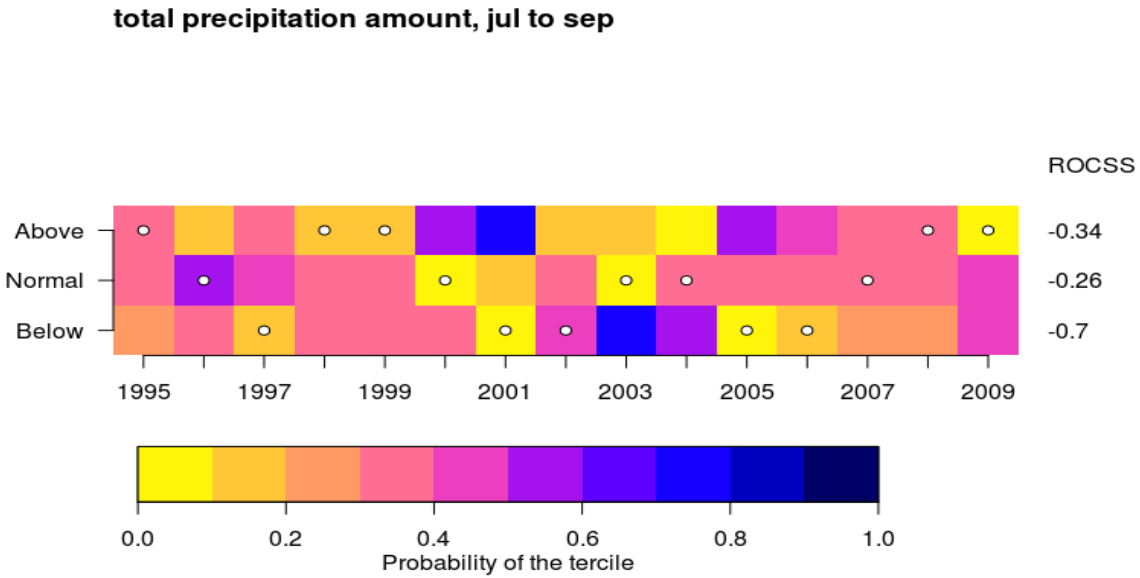
```
pred.sub <- subsetGrid(grid = cal,
                       latLim = c(9.5,10.2), lonLim = c(-11.5,-10))
obs.sub <- subsetGrid(grid = wfdei,
                      latLim = c(9.5,10.2), lonLim = c(-11.5,-10))
```



Tercile Validation Plot



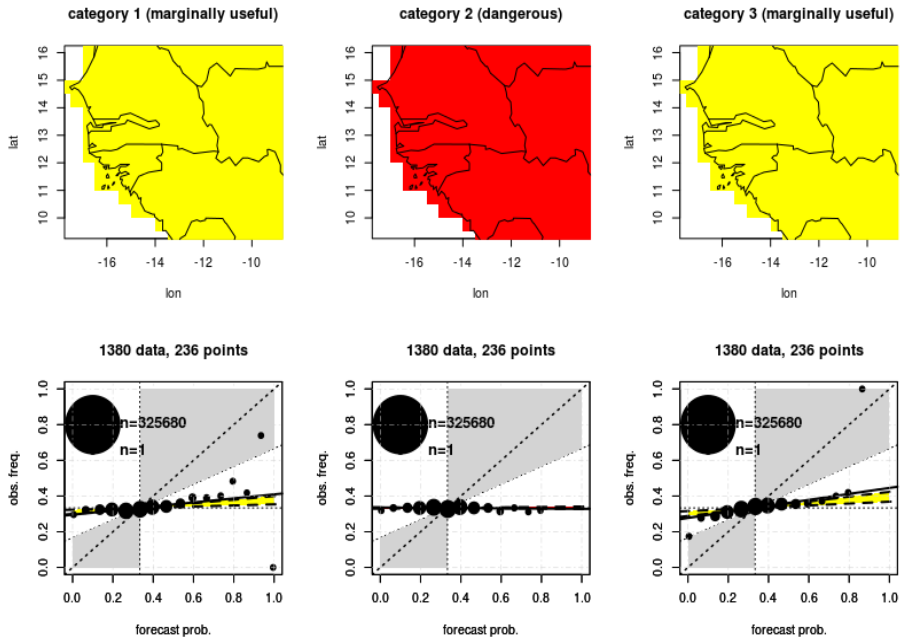

```
pred.sub <- subsetGrid(grid = pred,
                       latLim = c(40,42.5), lonLim = c(24,26.5))
obs.sub <- subsetGrid(grid = obs,
                      latLim = c(40,42.5), lonLim = c(24,26.5))
tercileValidation(pred = pred.sub, obs = obs.sub, color.pal = "ypb")
```



Implementation of tercile validation as pesented by:
Diez et al. 2011. doi:10.1111/j.1600-0870.2011.00523.x

Reliability Plots with reliability categories

```
Cfs2 ← interpGrid(cfs, new.coordinates = getGrid(wfdei))
reliabilityCategories(obs = cfs2, pred = wfdei, nbins = 3, nboot = 10)
```



Implementation of reliability as presented by:
Weisheimer & Palmer 2014. doi:10.1098/rsif.2013.1162

Conclusions

- All datasets and variables required by ECOMS users centralized in a single **OpeNDAP Server** (The ECOMS-UDG) with two extra layers for
  - 1) authentication and
  - 2) harmonization

- A suite of **R packages** fully integrated allow performing many different tasks, such as:
  - User friendly access to the ECOMS-UDG
  - Data transformation
  - Downscaling and bias correction
  - Verification
  - Data visualization
  - Specific CII calculation
  - ….

OPEN-SOURCE BENEFITS
* Reproducibility
* Customizability
* Flexibility
* Interoperability
* Auditability
...

Contents

Links

UDG wiki with instructions for registration

→ **https://meteo.unican.es/trac/wiki/udg/registration**

The ECOMS-UDG wiki

→ **http://www.meteo.unican.es/udg-wiki/ecoms**

Link to R packages (respositories and wiki-s):

→ **https://github.com/SantanderMetGroup/**