

# Rainfall Filtered Autoregressive Model (RainFARM) precipitation downscaling

*Jost von Hardenberg (ISAC-CNR)*

*26/03/2019*

## Introduction

It becomes often necessary to bridge the scale gap between climate scenarios of precipitation obtained from global and regional climate models and the small scales needed for impact studies, in order to measure uncertainty and reproduce extremes at these scales. An effective approach is provided by stochastic downscaling techniques (Ferraris et al. 2003a,b), in particular so-called full-field weather generators, which aim at generating synthetic spatiotemporal precipitation fields whose statistical properties are consistent with the small-scale statistics of observed precipitation, based only on knowledge of the large-scale precipitation field. These methods allow for example to estimate uncertainties at small scales in rainfall scenarios, by generating large ensembles of synthetic small-scale precipitation fields.

In particular the *Rainfall Filtered Autoregressive Model* (**RainFARM**; Rebora et al. 2006a,b) is a stochastic downscaling procedure based on the nonlinear transformation of a linearly correlated stochastic field, generated by small-scale extrapolation of the Fourier spectrum of a large-scale precipitation field. Developed originally for downscaling at weather timescales, the method has been adapted for downscaling at climate timescales by D'Onofrio et al. 2014 and recently improved for regions with complex orography in Terzago et al. 2018.

An efficient implementation of RainFARM is provided for CSTools by the `CST_RainFARM()` and `RainFARM()` functions.

## Downscaling seasonal precipitation forecasts with RainFARM

### Preliminary setup

In order to run the examples in this vignette, the *CSTools* package and some other support R packages need to be loaded by running:

```
library(s2dverification)
library(maps)
devtools::install_git('https://earth.bsc.es/gitlab/external/cstools', branch = 'RainFARM')
library(CSTools)
```

We use the `Load` function from the **s2dverification** package, to load a seasonal precipitation forecast from the ecmwfS5 forecast. There are 25 ensemble members available. We select a 4x4 pixel cutout at resolution 1 degree in a smaller area lon=[6,9], lat=[44,47] covering North-Western Italy and three starting dates on November 1st for years 2010, 2011 and 2012, selecting forecast times for the following month of March:

```
infile <- list(path = '../medscope/nwalps/data/$VAR_NAME$_$START_DATE$_nwalps.nc')
data <- Load('prlr', exp = list(infile), obs = NULL,
            sdates = c('20101101', '20111101', '20121101'),
            leadtimemin = 121, leadtimax = 151, lonmin = 5, lonmax = 9, latmin = 44,
            latmax = 47, storefreq = "daily", sampleperiod = 1, nmember = 25,
            output = "lonlat")
```

This gives us a CSTools object `data`, containing an element `data$data` with dimensions:

```
dim(data$data)
dataset member  sdate  ftime  lat  lon
      1      25      3    31    4    4
```

Please notice that RainFARM (in this version) only accepts square domains, possibly with an even number of pixels on each side, so we will need to select an appropriate cutout.

## Standard downscaling without climatological weights

Our goal is to downscale with RainFARM these data from the resolution of 1 degree (about 100 km at these latitudes) to 0.05 degrees (about 5 km) using the `CST_RainFARM()` function. This means that we need to increase resolution by a factor `nf=20`. RainFARM can compute automatically its only free parameter, i.e. the spatial spectral slope, from the large-scale field (here only with size 4x4 pixel, but we suggest to use at least 8x8 pixel in a real application). In this example we would like to compute this slope as an average over the *member* and *ftime* dimensions, while we will use different slopes for the remaining *dataset* and *sdate* dimensions (a different choice may be more appropriate in a real application). To obtain this we specify the parameter `time_dim=c("member", "ftime")`. The slope is computed starting from the wavenumber corresponding to the box, `kmin=1`. We create 5 stochastic realizations for each dataset, member, starting date and forecast time with `nens=5`. The command to donwscale and the resulting fields are:

```
data_down <- CST_RainFARM(data, nf=20, kmin = 1, nens = 5,
                          time_dim = c("member", "ftime"))

dim(data_down$data)
  dataset  member realization  sdate  ftime  lat  lon
      1      25           5      3     31   80   80
str(data_down$lon)
num [1:80] 5.53 5.58 5.62 5.67 5.72 ...
str(data_down$lat)
num [1:80] 47.5 47.4 47.4 47.3 47.3 ...
```

The function returns an array `data_down$data` with the additional “realization” dimension for the stochastic ensemble with 5 members. The longitudes and latitudes have been correspondingly interpolated to the finer resolution.

Alternatively we could have used the “reduced” function `RainFARM` which accepts directly a data array (with arbitrary dimensions, provided a longitude, a latitude and a “time” dimension exist) and two arrays to describe longitudes and latitudes:

```
downscaled <- RainFARM(data$data, data$lon, data$lat, nf=20, kmin = 1, nens = 5,
                      time_dim = c("member", "ftime"))
```

The resulting array has the same dimensions as the `$data` element in the result of `CST_RainFARM`.

Each instant and each realization will of course be different, but let’s plot and compare the original and the downscaled data of the first ensemble member and for the first realization for a specific forecast date (Fig. 1):

```
a <- data$data[1,1,1,17,,]*86400*1000; a[a>60] <- 60
image(data$lon,rev(data$lat),t(apply(a,2,rev)),xlab="lon",ylab="lat",
      col=rev(terrain.colors(20)), zlim=c(0,60))
map("world", add=TRUE); title(main = "pr 17/03/2010 original")
a <- data_down$data[1,1,1,1,17,,]*86400*1000; a[a>60] <- 60
image(data_down$lon,rev(data_down$lat),t(apply(a,2,rev)),xlab="lon",
      ylab="lat", col=rev(terrain.colors(20)), zlim=c(0,60))
map("world", add=TRUE); title(main = "pr 17/03/2010 downscaled")
```

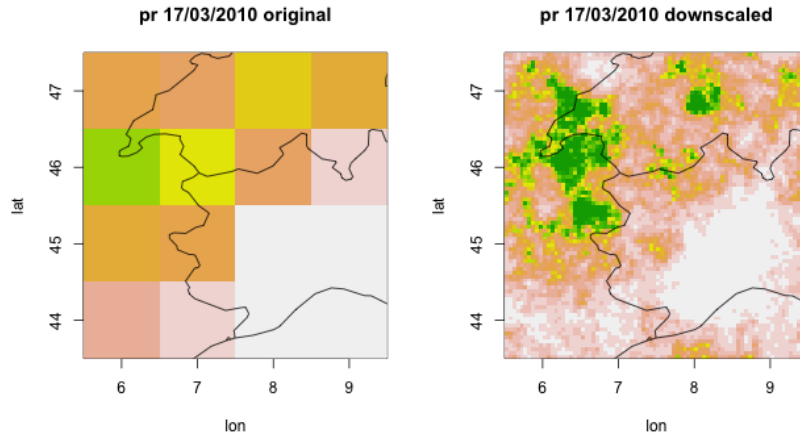


Figure 1: Comparison between an original precipitation field at 1-degree resolution (left) and the same field downscaled with RainFARM to 0.05 degree resolution (right)

RainFARM has downscaled the original field with a realistic fine-scale correlation structure. Precipitation is conserved in an average sense (in this case smoothing both the original and the downscaled fields with a circular kernel with a diameter equal to the original field grid spacing would lead to the same results). The downscaled field presents more extreme precipitation peaks.

### Downscaling using climatological weights

The area of interest in our example presents a complex orography, but the basic RainFARM algorithm used does not consider topographic elevation in deciding how to distribute fine-scale precipitation. A long term climatology of the downscaled fields would have a resolution comparable to that of the original coarse fields and would not resemble the fine-scale structure of an observed climatology. If an external fine-scale climatology of precipitation is available, we can use the method discussed in Terzago et al. (2018) to change the distribution of precipitation by RainFARM for each timestep, so that the long-term average is close to this reference climatology in terms of precipitation distribution (while the total precipitation amount of the original fields to downscale is preserved).

Suitable climatology files could be for example a fine-scale precipitation climatology from a high-resolution regional climate model (see e.g. Terzago et al. 2018), a local high-resolution gridded climatology from observations, or a reconstruction such as those which can be downloaded from the WORLDCLIM (<http://www.worldclim.org>) or CHELSA (<http://chelsa-climate.org>) websites. The latter data will need to be converted to NetCDF format before being used (see for example the GDAL tools (<https://www.gdal.org>)). We will assume that a copy of the WORLDCLIM precipitation climatology at 30 arcseconds (about 1km resolution) is available in the local file `medscope.nc`. From this file we can derive suitable weights to be used with RainFARM using the `RFWeights` functions as follows:

```
ww <- RFWeights("./worldclim.nc", nf=20, lon=data$lon, lat=data$lat)
```

The result is a two-dimensional weights matrix with the same `lon` and `lat` dimensions as requested. The weights (varying around an average value of 1) encode how to distribute differently precipitation in each stochastic realization of RainFARM. We call again `CST_RainFARM()`, this time passing climatological weights:

```
data_down_weights <- CST_RainFARM(data, nf=20, kmin = 1, nens = 5,
  weights = ww, time_dim = c("member", "ftime"))
```

We plot again the same realization as before in Fig. 2 (left panel). The WORLDCLIM climatology data used

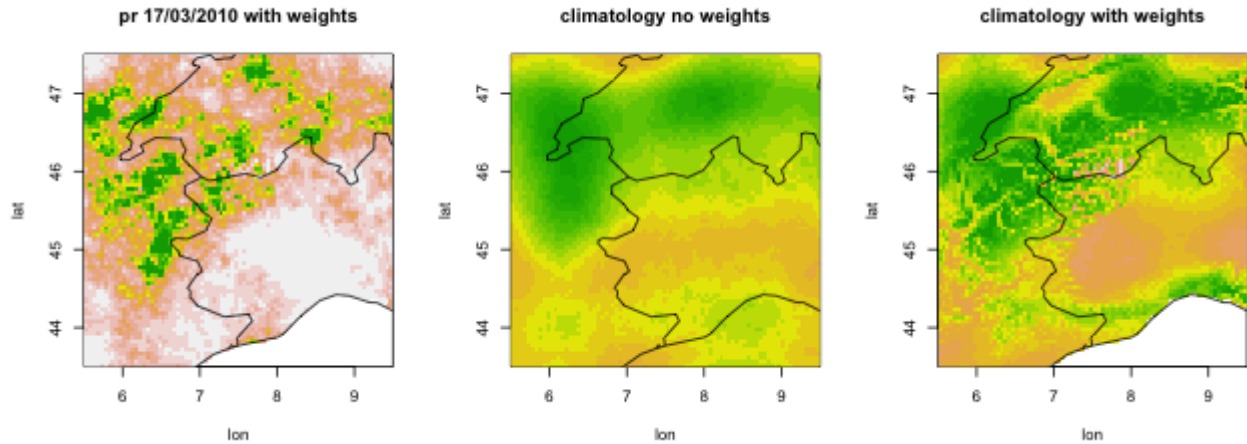


Figure 2: The same field as in Fig. 1 downscaled using climatological weights from the WORLDCLIM dataset. (left panel). The center and right panel show the climatology of the downscaled fields over all ensemble members, realizations and forecast times without and with climatological weights respectively. The center and right panels use a different colorscale than the left panel.

in this case have missing values over the ocean, so that the resulting downscaled maps have valid points only over land.

From a single realization and time it is not possible to see that a more realistic precipitation has been achieved, but this becomes clear if we compare the climatological average over all 5 stochastic ensemble realizations, over all forecast times and over all forecast times. The resulting plot in panels Fig2b,c show that the RainFARM climatology downscaled without weights presents on average a very coarse structure, comparable with that of the original fields, while when using the weights a much more realistic distribution is achieved.

### Determining the spectral slopes

The only free parameter of the RainFARM method is represented by the spatial spectral slope used to extrapolate the Fourier spectrum to the unresolved scales. When fine-scale precipitation data for the study area are available this slope could be determined offline from a study of these data. Else this slope can be determined directly from the large-scale fields to downscale. The RainFARM functions compute automatically this parameter by default over the input dimensions specified by the `time_dim` parameter. In many practical applications a careful study of an appropriate average to be used may still be necessary. This can be achieved using the function `CST_RFSlope()`. If for example we wish to determine the average slopes over all member and forecast times, but computing separate slopes for each starting date and dataset, we can do so with:

```
slopes <- CST_RFSlope(data, time_dim=c("member", "ftime"))
dim(slopes)
  dataset  sdate
      1      3
slopes
  [,1] [,2] [,3]
[1,] 1.532351 1.664028 1.459252
```

which return an array of spectral slopes, one for each “dataset” and starting date “sdate”.

## Compacting dimensions

RainFARM creates an additional dimension in the `$data` array of the output data structure, named “realization”. This is an additional ensemble dimension used to accommodate multiple stochastic realizations (if `nens>1` is chosen), in addition to the “member” dimension of the input array (which instead is usually used to represent for example multiple forecast realizations with the same model from the same starting date). Having a separate ensemble dimension for the stochastic ensemble allows to evaluate independently uncertainty due to the small scales resolved with downscaling, compared to other sources of uncertainty such as the dependence to initial conditions gauged for example by the original “member” dimension. Since other CSTools functions support only one ensemble dimension named “member”, it may be useful in some cases to compact together these two ensemble dimensions. In the output array the “realization” dimension is placed immediately after the “member” dimension, so the user could do this after calling the function reshaping the dimensions of the array. It is also possible to use the option `drop_realization_dim` of the `RainFARM` and `CST_RainFARM` functions which has the following behaviour when set to `TRUE`:

- 1) if `nens==1`: the “realization” dimension is dropped;
- 2) if `nens>1` and a “member” dimension exists: the “realization” and “member” dimensions are compacted and the resulting dimension is named “member”;
- 3) if `nens>1` and a “member” dimension does not exist: the “realization” dimension is renamed to “member”.

## Technical issues

For maximum efficiency, the `CST_RainFARM()`, `RainFARM()`, `RFWeights()` and `RFSlope()` functions are based on an external library written in the Julia language (<https://github.com/jhardenberg/RainFARM.jl>). The `JuliaCall` R package (<https://cran.r-project.org/web/packages/JuliaCall/index.html>) is used to call the external library. In particular the function uses the `julia_setup()` function from the `JuliaCall` library, which takes a long time (up to a few tens of seconds) only the very first time it is called. Further calls to these functions will present no delay.

## Bibliography

- Terzago, S., Palazzi, E., von Hardenberg, J. Stochastic downscaling of precipitation in complex orography: A simple method to reproduce a realistic fine-scale climatology. *Natural Hazards and Earth System Sciences*, 18(11), 2825–2840, doi:10.5194/nhess-18-2825-2018 (2018)
- D’Onofrio, D.; Palazzi, E., von Hardenberg, J., Provenzale A., Calmanti S. Stochastic Rainfall Downscaling of Climate Models. *J of Hydrometeorology* 15 (2), 830-843, doi:10.1175/JHM-D-13-096.1 (2014)
- Rebora, N., L. Ferraris, J. von Hardenberg, A. Provenzale. Rainfall downscaling and flood forecasting: A case study in the Mediterranean area. *Nat. Hazards Earth Syst. Sci.*, 6, 611–619, doi:10.5194/nhess-6-611-2006 (2006a)
- Rebora, N., Ferraris, L., von Hardenberg, J., Provenzale, A. RainFARM: Rainfall downscaling by a filtered autoregressive model. *J. Hydrometeor.*, 7, 724–738, doi:10.1175/JHM517.1 (2006b)
- Ferraris, L., Gabellani, S., Parodi, U., Rebora, N., von Hardenberg, J., Provenzale, A. Revisiting multifractality in rainfall fields. *J. Hydrometeor.*, 4, 544–551, doi:10.1175/1525-7541(2003)004,0544:RMIRF.2.0.CO;2 (2003a)
- Ferraris, L., Gabellani, S., Rebora, N., and Provenzale, A.. A comparison of stochastic models for spatial rainfall downscaling, *Water Resour. Res.*, 39, 1368, <https://doi.org/10.1029/2003WR002504>, (2003b)