NDVI Resampling using R-based tools

Xavier Rotllan-Puig1,2,✉, Federico Gianoli1, Pier Lorenzo Marasco1, and Michael Cherlet1

11/06/2020

1 Joint Research Centre – European Commission. Directorate D – Sustainable Resources. Unit D6 – Knowledge for Sustainable Development & Food Security Unit. Via Enrico Fermi 2749. I-21027 Ispra (VA), ITALY  
2 ASTER-Projects. Barri Reboll, 9, 1r. 08694 Guardiola de Berguedà (Barcelona), SPAIN

✉ Correspondence: [Xavier Rotllan-Puig <[xavier.rotllan.puig@aster-projects.cat](mailto:xavier.rotllan.puig@aster-projects.cat)>](mailto:xavier.rotllan.puig@aster-projects.cat)

**Abstract**

# Introduction

The Copernicus Global Land Service (CGLS; <https://land.copernicus.eu/global/>) is a component of the Land Monitoring Core Service (LMCS) of Copernicus, the European flagship programme on Earth Observation. CGLS systematically produces and distributes time series of global bio-geophysical products on the status and evolution of the land surface, at different spatial resolutions. These products are used to monitor the vegetation, the water cycle, the energy budget and the terrestrial cryosphere.

The CGLS vegetation-related products (i.e. NDVI, FAPAR…), based on PROBA-V observations, have been distributed at 1km and 333m until June, 2020. As of this date, the new products are no longer provided at 1km resolution. However, the users who might be interested on continuing their 1km time series can use a resample of the 333 m products.

The Joint Research Centre (European Commission) provides different tools to help the users make their own resampling exercices. For example, a Notebook with R code and some explanations can be found at <https://nbviewer.jupyter.org/github/VITObelgium/notebook-samples/blob/master/datasets/probav/ResampleTool_R_notebook.ipynb>. In addition, Python code to run in QGIS can be found in …**Fede??**.

In this document we present some comparisons of the results obtained with these two resampling tools with the original CGLS products at 1km resolution, as well as among them.

# Materials and methods

## Data sets

The analysis has been done using a subset of the 10-daily Normalized Difference Vegetation Index (NDVI) global product. The selected images (i.e. 1km and 33m resolution) were taken the May 01, 2020, and the working maps were cut between the coordinates (DD) xmin = -18.58, xmax = 51.57, ymin = 28.5, ymax = 62.95. Figure 1 shows the 333m NDVI working map used for the resampling.

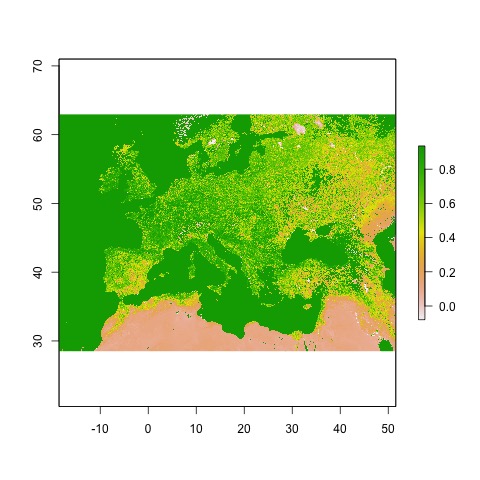


Figure 1: NDVI working map at 333m resolution

The original Global Land product files usually can be downloaded as a netCDF4 file. They can often contain specific values for invalid pixels (flagged values), which need to be dealt with. In the case of the NDVI products, digital values in the netCDF (DN) > 250 are flagged and need to be converted into NA. When the netCDF files are read in as a *raster* object, the digital values are scaled into real NDVI values automatically. After that, all pixels with NDVI values > 0.92 (= DN 250 x scale + offset) were set to NA.

The information regarding flagged values, as well as other supporting information, can be seen in the netCDF file metadata. In addition, more details on the CGLS global products can be found in their Product User Manual documentation at <https://land.copernicus.eu/global/products/>.

## Resample method

There are several approaches to resample data from a finer to a coarser resolution. They could be grouped into area-based aggregation methods and point-based interpolation methods (e.g. Bilinear and Nearest Neighbour), and can be applied depending on the data type, etc.

The aggregation method used in this assessment groups rectangular matrix of pixels of the finer resolution image to create a new map with larger cells. In this case, as we wanted to resample from 333m to 1km, a factor of 3 was implemented (i.e. a matrix of 3x3 pixels).

On the one hand, for the R-based method, the function *aggregate()* of the package *raster* was used. *aggregate()* can perform the calculation using different functions. While the default is the average (*mean()*) it can work also with *modal()*, *max()*, *min()* or even with *ad hoc* functions programmed by the user. For NDVI products the most suitable method is the average, but we also included the condition that at least 5 out of the 9 pixels had to be non-missing (i.e. not NA).

On the other hand, for the QGIS method…. **Fede**

## Metrics and plots

In order to assess the performance of the resampling methods, besides maps of the results, three well known and widely used metrics and a scatterplot were produced. The metrics are:

* Pearson correlation coefficient (Pearson’s *r*)
* Root-mean-square error (RMSE)
* Mean absolute error (MAE)

The R code used to perform the assessment reported in this document can be seen at <https://github.com/xavi-rp/NDVI_resample>.

# Results and discussion

Table 1 shows the three metrics calculated to evaluate the comparisons of the two resampling methods among them and with the original CGLS NDVI product at 1 km resolution.

Table 1:

Table continues below

|  |  |
| --- | --- |
| objects | Pearson’s r |
| orig-1km\_\_resampl-1km-RAggreg | 0.979 |
| orig-1km\_\_resampl-1km-QGIS-Aggreg | NA |
| resampl-1km-R-Aggreg\_\_resampl-1km-QGIS-Aggreg | NA |

|  |  |
| --- | --- |
| Root Mean Square Error | Mean Absolute Error |
| 0.051 | 0.031 |
| NA | NA |
| NA | NA |

## R-based method vs Python-based method

## R-based method vs original 1km product

Figure 2 shows both the original NDVI map at 1km resolution and the resampled one to 1km using the R-based method.

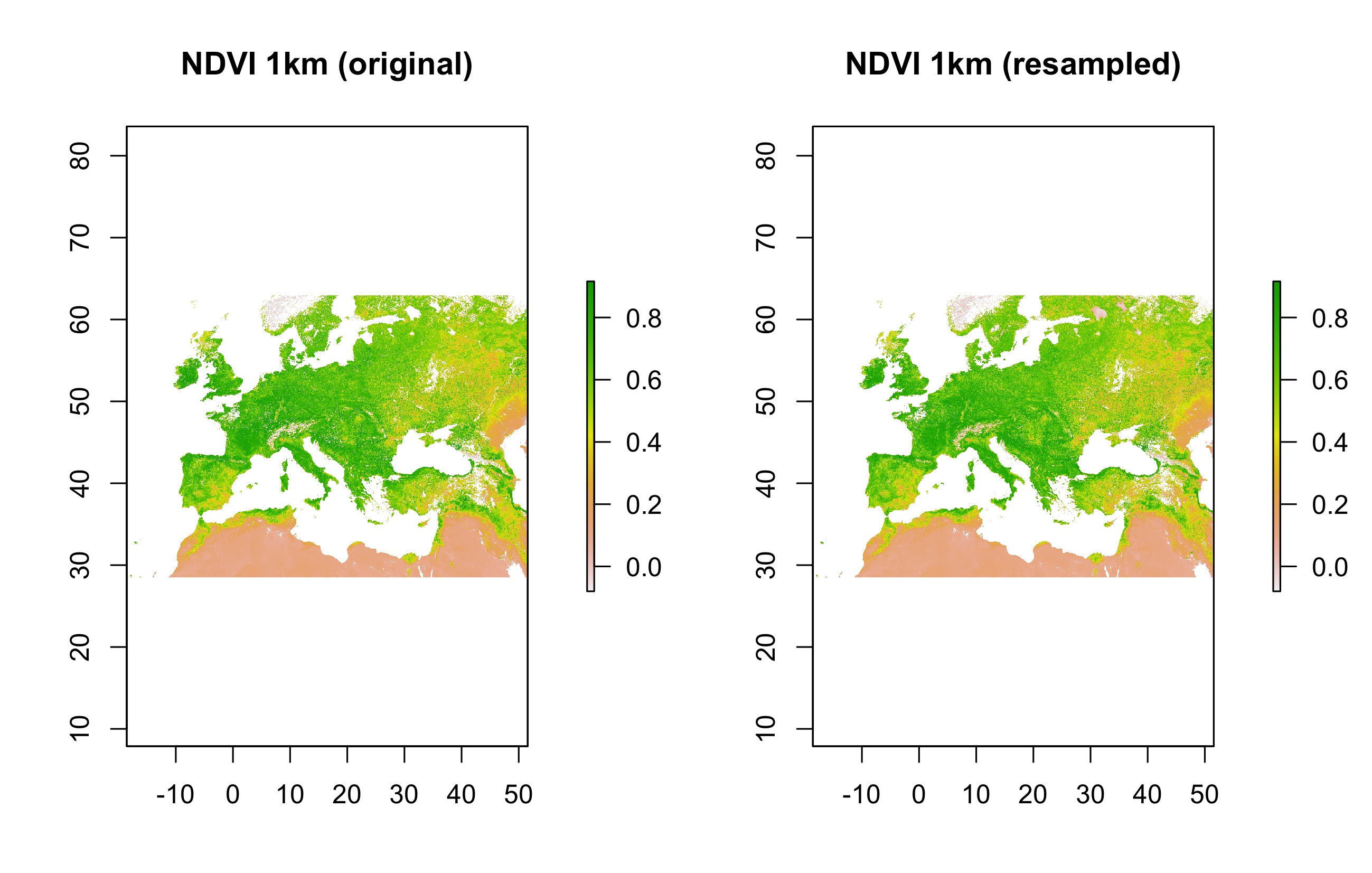


Figure 2: Original NDVI map at 1km resolution and the resampled one using the R-based method

As it can be seen in the scatterplot (Figure 3), and corroborated by the Pearson’s *r* (0.979), there was a good level of correlation among the original 1km map and the resampled one using the R method. Also RMSE and MAE reported good levels of ‘error’ among the two maps (0.051 and 0.031, respectively).

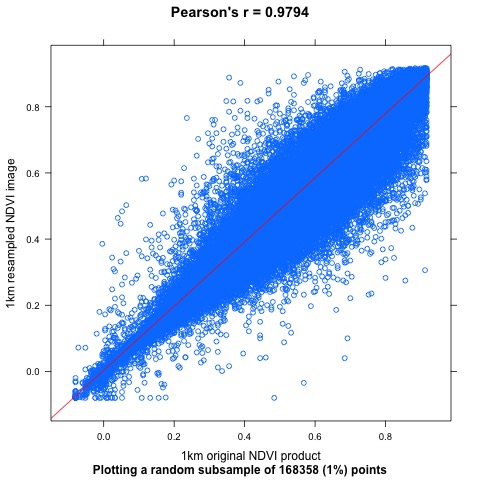


Figure 3: Scatter plot displaying a random subset (1%) of pixel values of the 1km original NDVI product against the values of the same pixels of the resampled map using the R method (blue points). Also the regression (red) line

## Python-based method vs original 1km product