CondMerg: an open source implementation in R language of conditional merging for weather radars and rain gauges observations

(Article begins on next page)

26 April 2024

CondMerg: an open source implementation in R language of conditional merging for weather radars and rain gauges observations

D. Guenzi[a]*, S. Fratianni[ab], R. Boraso[c], R. Cremonini[d]

[a] Università degli Studi di Torino, dipartimento di Scienze della Terra, via Valperga Caluso 35 – 10125 Torino (Italy)

[b] Centro interdipartimentale sui rischi naturali in ambiente montano e collinare NatRisk, via Leonardo da Vinci 44 – 10095 Grugliasco (TO, Italy)

[c] CSP Innovazione nelle ICT, strada del Lionetto 6 – 10146 Torino (Italy)

[d] ARPA Piemonte, dipartimento Sistemi Previsionali, via Pio VII 9 – 10135 Torino (Italy)

*Corresponding author. Tel.: +39 011 670 5172, e-mail: diego.guenzi@unito.it

Abstract

Monitoring of extreme events requires accurate measurement of rainfall intensities and merging weather radar data with ground information is a very common technique used to obtain the required precision. In order to do this, several methods exist but very few open source implementations are available. CondMerg is the first open source software developed in R language implementing the conditional merging method and some other experimental variants based on it. It is a cross-platform software, easily adaptable to different needs, optimized for batch processing of multiple events but also usable in near real time applications. For its execution it requires two inputs: a CSV file with rain gauges measurements and a geo-referred TIF file with weather radar quantitative precipitation estimations; main outputs are TIF files with merged observations although the code also returns information about cross-validation, with scatter plots and indexes. All TIF files are ready to be managed by common GIS software for easy visualization and analysis. Use of the program is very simple: execution can be interactive or non-interactive and, in both cases, it just requires to set some parameters at the beginning of the program and run it. The code has been tested on different extreme rain events occurred in the Piedmont region (northwestern Italy) showing improved accuracy of reconstructed rainfall fields.

Keywords

Open source, conditional merging, weather radar, rain gauges network, R project, k-fold cross-validation

1. Introduction

Analysis, monitoring and forecasting of extreme events requires accurate measurements of rainfall intensities (Zandonadi et al. 2016). Several methods and instruments exist to estimate rainfall intensities (Allamano et al. 2015, Barrett 1970, Messer et al. 2006, Rabiei et al. 2013) but the most common approaches are based on weather radars (Cremonini et al. 2014) and rain gauges (Acquaotta et al. 2016). Weather radars are particularly good in obtaining the rainfall field structure, with both high spatial and temporal resolution (Doviak and Zrnic 1984). However, due to several uncertainties (Jordan et al. 2000, Vignal and Krajewski 2001), they lack in precise rainfall intensities estimations. A common technique used to reconstruct the rainfall field with greater accuracy is based on the comparison of weather radar observations with ground measurements given by rain gauges, which are spatially scattered but very accurate. This association is an essential step in many research fields including, for example, weather forecasting (Cremonini and Bechini 2003, Ehret 2002, Legorgeu et al. 2012) and hydrological modeling (Borup et al. 2015, Cole and Moore 2008, Goormans and Willems 2013). In order to do this association, we can choose among a wide variety of methods that are considered as state-of-the-art (Delobbe et al. 2008, Goudenhoofdt and Delobbe 2009). Currently, *kriging with radar-based error correction* or, simply, *conditional merging* (Ehret 2002, Sinclair and Pegram 2005) is considered the most suitable method due to the high quality of results and because of its simplicity and computational efficiency.

In this study, we present the first free and open source implementation of the classical conditional merging method, with the addition of some experimental variants and a cross-validation system. This software, called *CondMerg* and written in R language (R Development Core Team 2011), is available online[1] under GNU GPL v3 license (Free Software Foundation 2007) and it can be used by the community both as a starting point for further customizations and as a standalone program (see Online Resource 1).

This paper is organized as follows: in chapter 2 we present a short overview of conditional merging and k-fold cross-validation methods while in chapter 3 we specify the necessary pre-requisites in order to run the software described in chapter 4. In chapter 5 we describe a use case where CondMerg has been used and in chapter 6 we summarize our work and introduce some future perspectives.

2. Methodology

CondMerg aims to help many professional figures like engineer and researchers in different fields (climatologists, meteorologists, radar specialists, hydrologists, modelers and many others) by combining radar data with rain gauges

---

[1]   https://github.com/UniToDSTGruppoClima/CondMerg

information. To reach this objective, it uses the conditional merging method, which is based on ordinary kriging process.

Ordinary kriging (Isaaks and Srivastava 1989, Krige 1951, Oliver and Webster 1990) is a regression method mainly used in geostatistical analysis that allows interpolation of values minimizing mean squared error. The idea is that if we have a known measure in some points, it is possible to determine the value of other neighboring points where there are no measures available. This spatial interpolation is based on the autocorrelation (or spatial dependence) of a variable, following the principle that closest points are more similar than distant ones. The unknown value in a particular point is calculated with a weighted average of the known values. The weights depend on the spatial relationship between the measured values in the neighborhood of the unknown point. To calculate the weights, a semivariogram is used: a graph that relates the distance between two points and the value of semivariance between the measurements taken in these two points; this semivariogram exhibits, both in qualitative and quantitative manner, the degree of autocorrelation.

Conditional merging CM uses the radar field to estimate the error associated with ordinary kriging on rain gauges and tries to correct it. First, radar values R at each gauge site g are used to produce a radar-based kriging field $K_{R(g)}$. Then, this field is subtracted from the original radar field R to obtain an error field; finally, the error field is added to the gauge-based kriging field $K_{G(g)}$ (Goudenhoofdt and Delobbe 2009). It is important to notice that a convective cell could be missed by the rain gauges but seen by radar. This is not a problem for conditional merging method since it preserves the spatial variability given by the radar, just working on intensities. Final formula is expressed by equation (1):

$$CM = R - K_{R(g)} + K_{G(g)} \tag{1}$$

Relying on equation (1) we have also implemented two variants on that method, which use the same logic but do not work on error fields (subtraction) but on bias fields (division). The first method is called *bias field conditional merging BFCM* and consist in taking the original radar field, dividing it by the radar-based kriging field and multiplying the result by the gauge-based kriging field, as expressed by equation (2):

$$BFCM = \left(\frac{R}{K_{R(g)}}\right) K_{G(g)} \tag{2}$$

This method introduce some minor problems related to division by zero or even indeterminate forms, but the approach we adopted in these cases is simply to assume that the final value is 0. The second method is called *mean conditional merging MCM* and it is the mean between conditional merging and bias field conditional merging, as shown by equation (3):

$$MCM = \frac{CM + BFCM}{2} = \frac{R - K_{R(g)} + K_{G(g)} + \left(\frac{R}{K_{R(g)}}\right) K_{G(g)}}{2} \tag{3}$$

Since all methods presented here rely on spatialization algorithms applied on a set of points ($K_{R(g)}$ or $K_{G(g)}$), having a dense network of rain gauges gives more accurate results, even if it is not mandatory nor there is a minimum number of gauges required. This is true because an accurate variography (and the relative semivariogram applied to the kriging method) could greatly compensate a sparse network (Journel and Huijbregts 1978, Pannatier 1996).

To verify and validate performances of merging methods, usually, a cross-validation technique is used since it is uncommon to have an independent verification network (Goudenhoofdt and Delobbe 2009). In that case, the complete rain gauges dataset is split in two parts: a training set and a test set; merging is calculated on the training set and results are validated against the test set. The choice of training set and test set defines the type of cross-validation. In our study, we used a k-fold cross-validation method (Arlotte and Celisse 2010, Payam et al. 2009) with a customizable k parameter, letting the user choose the accuracy of the validation. In fact, in k-fold cross-validation the complete dataset is randomly partitioned into k samples (possibly equal sized). On each iteration of the algorithm, a single sample is used as test set while all remaining groups are used as training set. This process is then repeated k times, always using a different test set; in the end, the k results obtained from the iterations can be combined to produce a single estimation. It is very interesting that using a k value equal to the number of elements in the complete dataset is equivalent of doing a leave-one-out cross-validation, moving from a non-exhaustive cross-validation method to an exhaustive one, thus increasing the accuracy of the validation (Arlotte and Celisse 2010).

3. Program pre-requisites and requirements

Since R is an interpreted language, CondMerg program requires it to be executed, resulting in easy portability on different types of platforms and operating systems. It has been successfully tested under R versions 3.0.2 (Frisbee Sailing), 3.1.3 (Smooth Sidewalk) and 3.2.4 (Very Secure Dishes), both under Windows and Linux, but it is likely to work even on later versions. In addition to its portability, we chose R because it is quite spread among main stakeholders of this application, it can handle large amounts of data and it can carry out time-efficient mathematical manipulations, generating standard graphical outputs that could be imported easily in all GIS software.

To execute CondMerg, in addition to R core functions, some packages have to be installed in the environment; these packages are *rgdal*, *xlsx*, *automap*, *raster*, *plyr*, *mblm* and *doParallel*. It is also important to meet the dependencies of previously mentioned packages and these are *rJava*, *xlsxjars*, *sp*, *foreach*, *iterators*, *parallel* and an installed Java VM.

4. CondMerg design and implementation

Despite CondMerg does not offer a true graphical user interface (GUI), it is very simple to use and requires little user interaction. All parameters are configurable inside the source code (see Online Resource 1) in a single section at the

beginning of the file. These parameters include the path of output and input files (in case of non-interactive run), the possibility of excluding or keeping all precipitation data smaller than 1 mm in operations concerning ratios (to avoid the growth of rain values), the k value in the k-fold cross-validation method ($0 \leq k \leq$ number of rain gauges) and the desired percentile of data to be used for drawing the regression line that does not keep into account the outliers. Once these parameters are set, CondMerg can be run: depending on the type of execution (interactive or not – i.e.: from RStudio or from Rscript), the program can ask to select the path where input data can be found or use the hardcoded one.

4.1. Input and output files

The program processes in parallel all CSV files contained in the specified path starting with *RainGauges_* and followed by a string defining a date and/or an hour (or everything else used to identify the data). Every single CSV file has a list of rain gauge identification codes with their geographical coordinates and the value of the measured precipitation within the chosen time period (that represents the constant temporal resolution of your data - see Online Resources 2, 3 and 4). In addition, the program needs a geo-referred TIF file for each CSV, which contains raw weather radar data (such as quantitative precipitation estimations - QPE) for the same time period used in the corresponding CSV file. The TIF file name has to start with the suffix *Radar_* and has to continue with the same string defining a date and/or an hour used in the corresponding CSV (see Online Resources 5, 6 and 7). By default, it has to use the Coordinates Reference System (CRS) based on WGS84/UTM 32N (since it has been tested mainly in that zone), but the source code is easily modifiable to adopt other CRSs. It is mandatory that all rain gauges are located inside the rectangular area covered by the QPE and it is also better if that area is slightly bigger than the area that is going to be analyzed. This because the kriging (and, consequently, all merging methods) could be less precise on the borders, where the network is normally less dense and there are zones that are not covered by rain gauges. Note that CondMerg must be used as a final step in the processing of radar information, since it does not check for input data quality. This means that all corrections (clutter, beam blocking, attenuation…) and conversions computed with Z-R relationship should be applied first. For our algorithm, this is called "raw radar data" since it represents the starting information we have to refine even if QPE, from another point of view, is not the real raw radar product.

Concerning outputs, the program creates three customizable folders where it saves the results. The first one is for geo-referred TIF files (see Fig. 1 and Fig. 5 for some examples) that represents the most important output of the program.
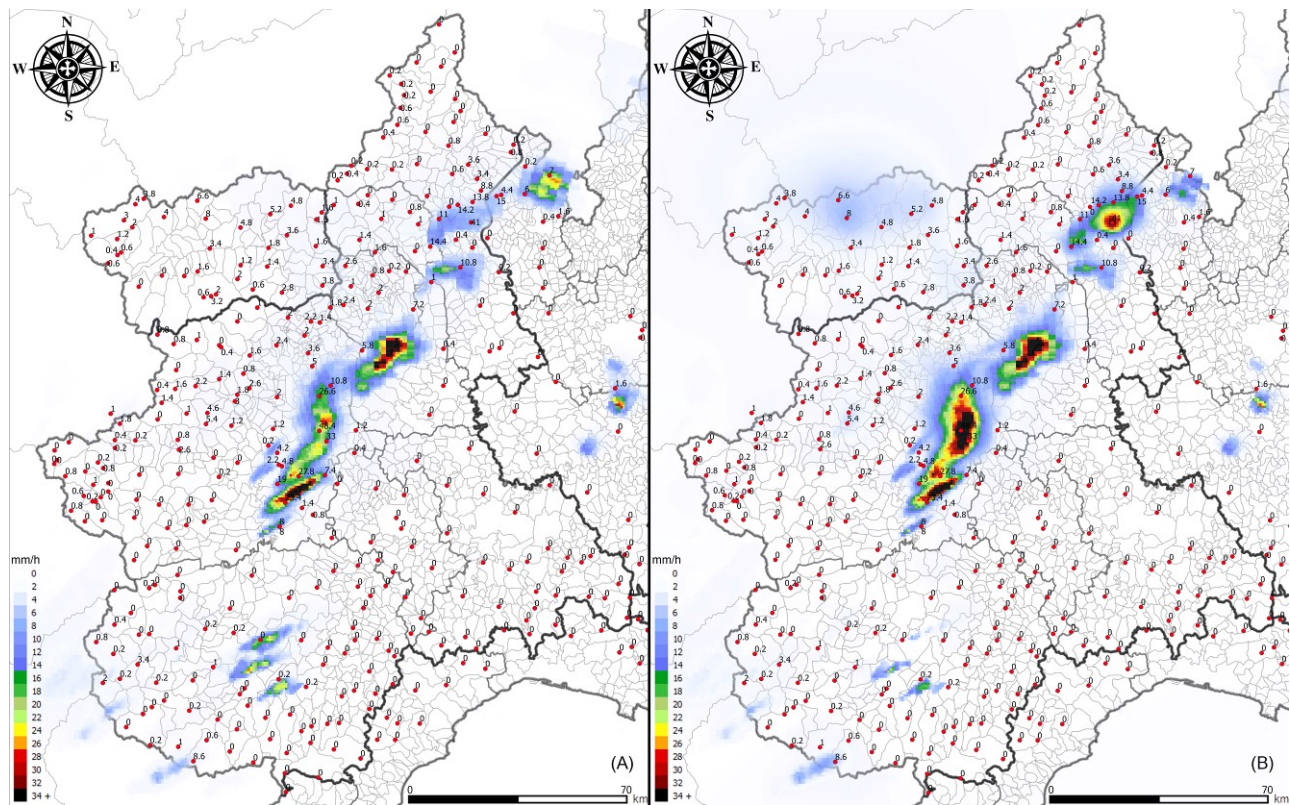

Fig. 1: raw radar (A) versus conditional merging (B) hourly precipitation data, loaded in QGIS with false colors (information coming from a rain event in the NW part of Italy, 28 July 2014 h15.00-16.00)

Here, there are raster products of merging processes like conditional merging, ordinary kriging of rain gauges and the two other experimental merging methods described in chapter 2, in addition to a CSV file that contains the information used for the statistics computation (see chapter 4.2 and 4.3). The second folder is reserved for raster files containing the mathematical differences between raw radar data and merging products, used to see immediately the differences between the three merging algorithms. The last folder, that is created only if parameter $k > 1$ (which means that k-fold is enabled), is for the cross-validation and contains both tables and charts. The tables are XLSX and CSV files about indexes on the different algorithms adopted, while the charts are scatter plots about raw radar data, merged products (using the complete dataset, without cross-validation) and cross-validated merged products (putting together the results of all k test sets to obtain a single, fully-validated set) compared with rain gauges information. Moreover, there are also scatter plots about the cross-validated data compared with the not validated one.

A complete list of input and output file names can be seen on Fig. 2. Online Resources 2, 3, 4, 5, 6 and 7 show three couples of input files (three CSVs and three TIFs) that could be used to test and practice with the program, as discussed in chapter 5.
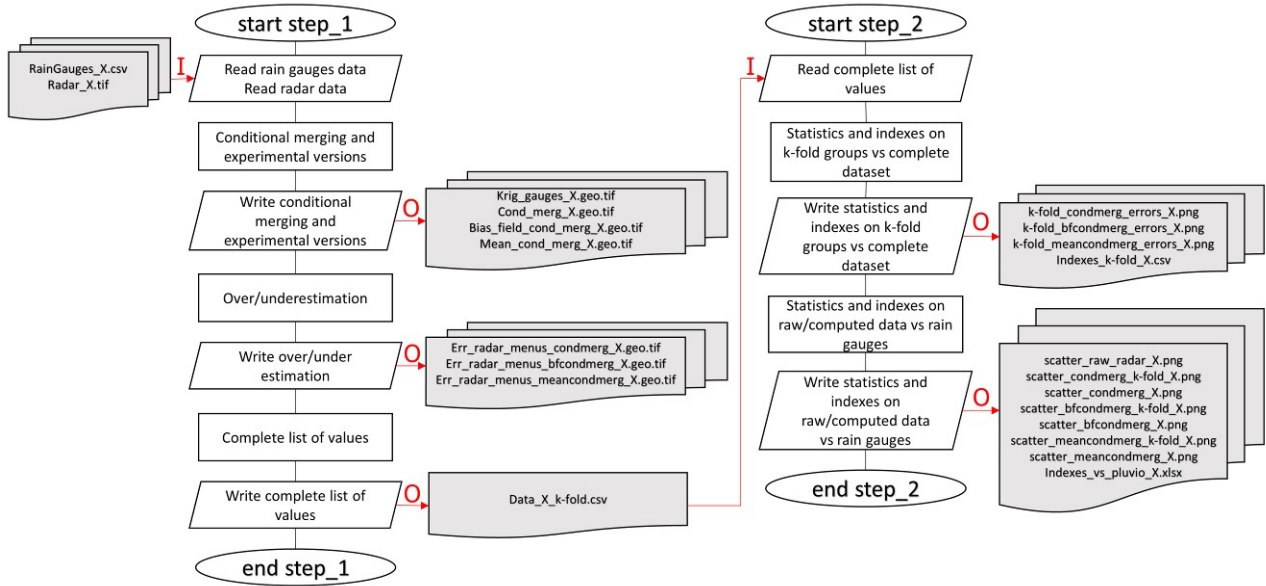
Fig. 2: simplified workflow of CondMerg program, with input and output files

4.2. Features and flows

CondMerg has been written to efficiently process files in batch mode (i.e.: using multi-core architectures and parallel code) and, even if it can be used on a single couple of input files and on a single core machine, it is not optimized for that. Despite this, if the needed approach is similar to the latter one, the source code could be easily modified to suit this case.

Computational complexity of CondMerg is given essentially by the ordinary kriging process it uses, which is $O(n^3)$ where n is the number of gauges you have to spatialize (Memarsadeghi et al. 2008). To give some time statistics, we executed CondMerg on a desktop CPU (a quad core Intel Core i7-4790 with Hyper Threading enabled, supporting up to 8 parallel threads) with eight couples of input files, each one with a 160.000 km^2 area covered by 384 gauges. Complete program execution, in parallel on all inputs, took less than 4 minutes to complete without doing cross-validation (with $k=0$), showing that it could be used for near real time adoption if data is acquired with a rate equal or greater than 5 minutes, in our example. If we enable the cross-validation step, the same elaboration takes around 4.3 minutes for every iteration of k (i.e.: 43 minutes with $k=10$, 3.6 hours with $k=50$, 11 hours with $k=150$…).

The software works in two steps (see Fig. 2): in the first one, the program reads input files, applies conditional merging and the experimental bias field conditional merging and mean conditional merging. After that, the software computes mathematical differences between raw radar data and merging products. This entire first step is iterated $k+1$ times since it has to be done inside the k-fold cross-validation process, which splits the dataset in k groups (the test sets) using a nondeterministic approach. Parameter k is user-defined (as specified in chapter 4) and could be intended as an indicator of the accuracy of the cross-validation process (greater is k, greater the accuracy). Only the first iteration is used to

generate real output files, all other iterations are just recorded inside the CSV file that is used for statistical analysis. This CSV file contains the list of values in rain gauges points that are used for all merging methods, both with the complete dataset and with the cross-validated values, obtained from different test sets joined in a single and fully validated data set. This file is the input for the second step of the program, which consists in the computation of statistics, indexes and scatter plots described in chapter 4.3. It is kept on disk just like all other outputs, even if it is an internal file not intended for use by common users, because it could be interesting for debug reasons or for specific values analysis.

## 4.3. Statistics, indexes and plots

On every resulting dataset described by the previously explained CSV file (raw radar, conditional merging, cross-validated conditional merging, bias field conditional merging, cross-validated bias field conditional merging, mean conditional merging and cross-validated mean conditional merging), CondMerg computes many indexes that can be found in the XLSX file and can be used for the statistical evaluation of the applied methods. These indexes are the mean error, mean absolute error, relative mean absolute error, mean bias, root mean square error, standard deviation, coefficient of determination R^2 and the A and B parameters of the simple linear regression $y = Ax + B$; the same indexes are also applied comparing the cross-validated data with the not validated one. In every single index, the best value obtained is highlighted in yellow and the best cross-validated value obtained is highlighted in green; if no yellow highlight exists, than it coincides with the green one. All these information are also visible in scatter plots produced by the software (see Fig. 3 for some examples), where data above the previously configured percentile (see chapter 4) is highlighted in red. On these plots, three different regression lines are shown: the one in red uses the complete dataset (that is also recorded inside XLSX file); the one in black excludes the outliers and the last one, which is in dark yellow, is the one derived from Theil-Sen estimator (Sen 1968, Theil 1950). All these plots could greatly help in visual interpretation and validation of obtained results.
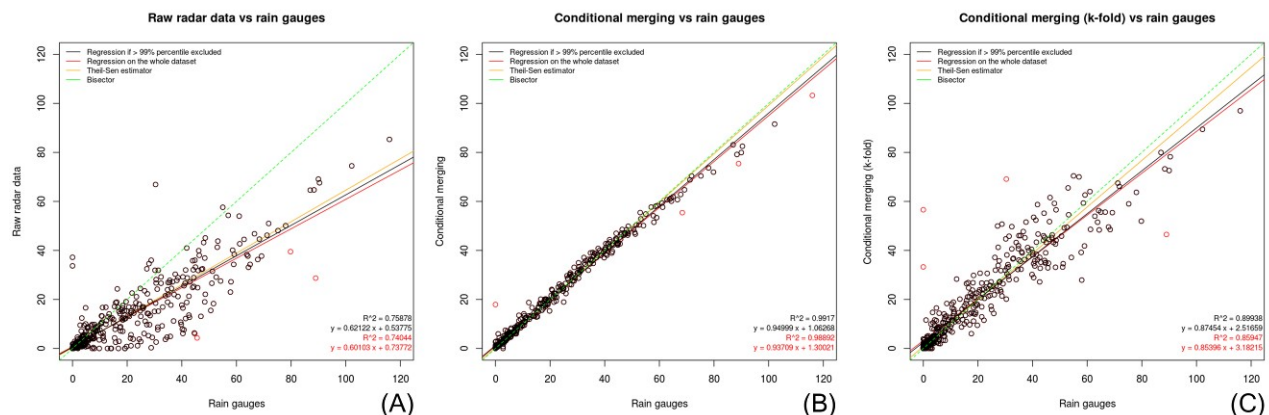
Fig. 3: scatter plots of raw radar data (A), conditional merging with complete dataset (B) and cross-validated conditional merging (C) information, referred to the same event described on Fig. 1 but on daily precipitation information

5. Application of CondMerg to a case study

To understand how CondMerg works, we present here a practical application: on 28 and 29 July 2014, a cyclonic flow coming from the north of France arrived on the Piedmont region, in northwestern Italy (see Fig. 4 and Online Resource 8). This caused a low-pressure area with rainstorms and hails near Turin (the main city of Piedmont) but rapid low-pressure departure toward east ensured that precipitation did not persist too much over the same area. Despite of this, mean rain values had been measured between 30 and 40 mm in 12 hours, with peaks around 120 mm; other peaks had been measured up to 64 mm/h, 89.8 mm in 3 hours and 115.8 mm in 6 hours, leading to an estimated return period greater than 50 years (ARPA Piemonte, 2014). In this extreme event, radar QPE and rain gauges information did not match most times, so we had to reconstruct the rainfall fields using CondMerg (e.g.: see central zone of Fig. 1 A, where rain gauges measured 48.4 mm/h and 33 mm/h while radar showed values of 24.2 mm/h and 16.4 mm/h, respectively).
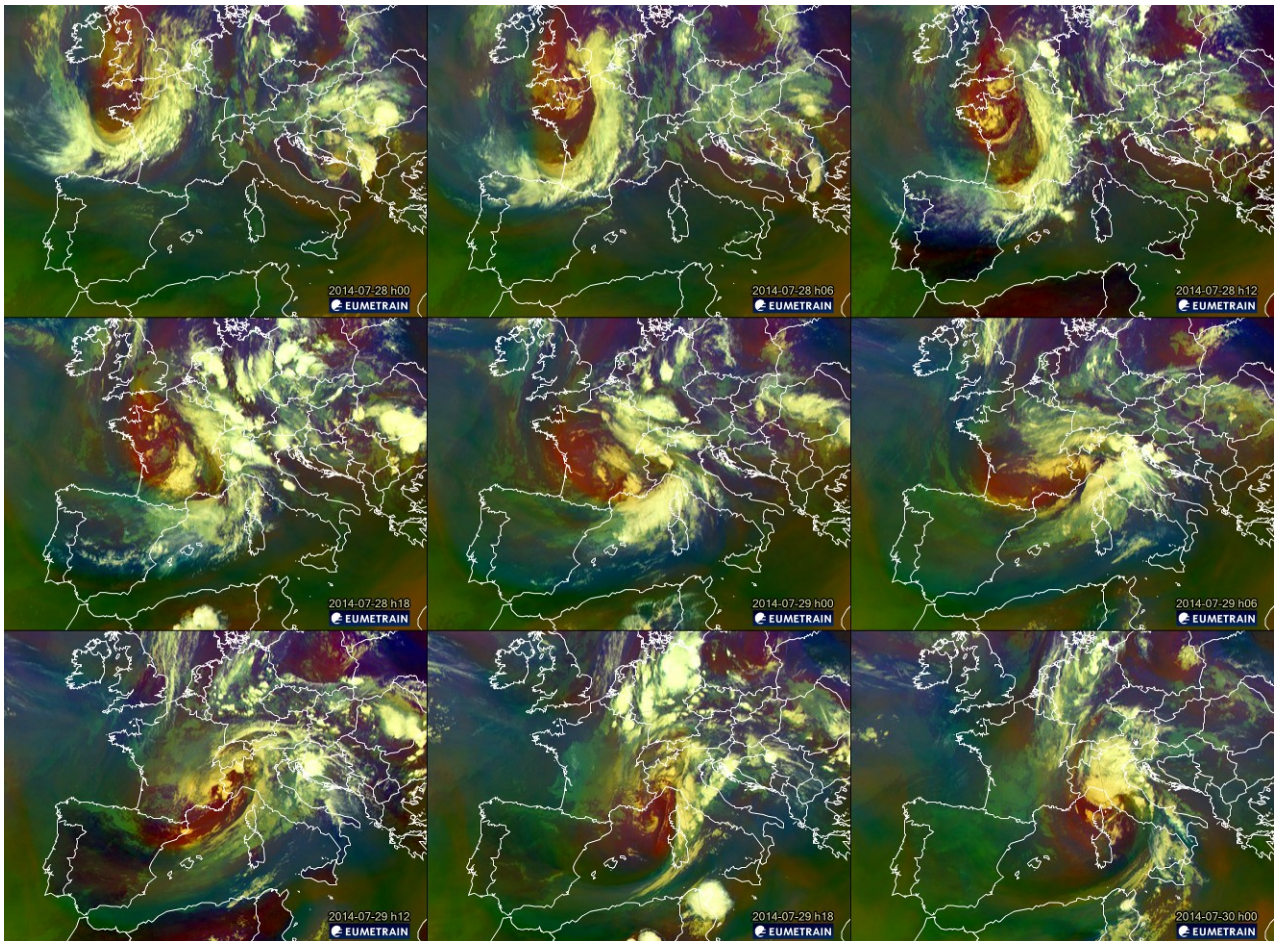

Fig. 4: satellite information composition of the extreme event of 28 and 29 July 2014 about Airmass (an RGB composite based upon data from infrared and water vapor channels from Meteosat Second Generation), taken from Eumetrain - http://www.eumetrain.org

Since we just want to analyze three main aspects of the extreme event and, in particular,

1. daily information about the 28 July on the whole Piedmont (see Online Resources 2 and 5),

2. hourly information from 14.00 to 15.00 just for a small part of the territory (see Online Resources 3 and 6) and

3. hourly information from 15.00 to 16.00 for the whole territory (see Online Resources 4 and 7),

we prepared the corresponding 6 input files (3 couples of TIF/CSV files) as described in chapter 4.1, keeping in mind to insert also some rain gauges that are located just outside of the study area, to avoid possible artifacts or reduction in precision near the borders of the image. After that, we had to set the values of variables in the section *Hardcoded and editable parameters* (see Online Resource 1) as described in chapter 4 and, finally, we ran the script. Since we chose to run it under RStudio, we had an interactive execution having CondMerg ask for the path of input files. After specifying the folder in the selection window, the program starts its computations; in the end, three new subfolders are present in that folder, as described in chapter 4.1. Looking inside these folders, we can find 21 TIFs, 30 PNGs, 3 XLSXs and 6 CSVs (that are 7 TIF files, 10 PNG files, 1 XLSX file and 2 CSV files for each input couple). All TIF files could be opened and visualized in GIS software (like QGIS or ArcGIS) and represent the most important output of the program (see Fig. 1 and Fig. 5 for some examples). PNGs are all scatter plots that could be used for data comparison and validation; they can be divided into two classes: those that start with the *k-fold* string before the merging method name and those that do not have it. First ones compare the cross-validated data with the complete, not validated dataset while the latter ones compare the merging products with rain gauges data. From the latter class, we have scatter plots of the merging products both with the complete dataset (see Fig. 3 B) and with cross-validated data only (see Fig. 3 C); indexes related to this group can be found in the XLSX file (see Table 1). Concerning CSV files, normally they can be ignored, since they represent debug output (see chapter 4.2) or indexes on test sets compared with the complete dataset (that are less significative than the one presented in the XLSX file).
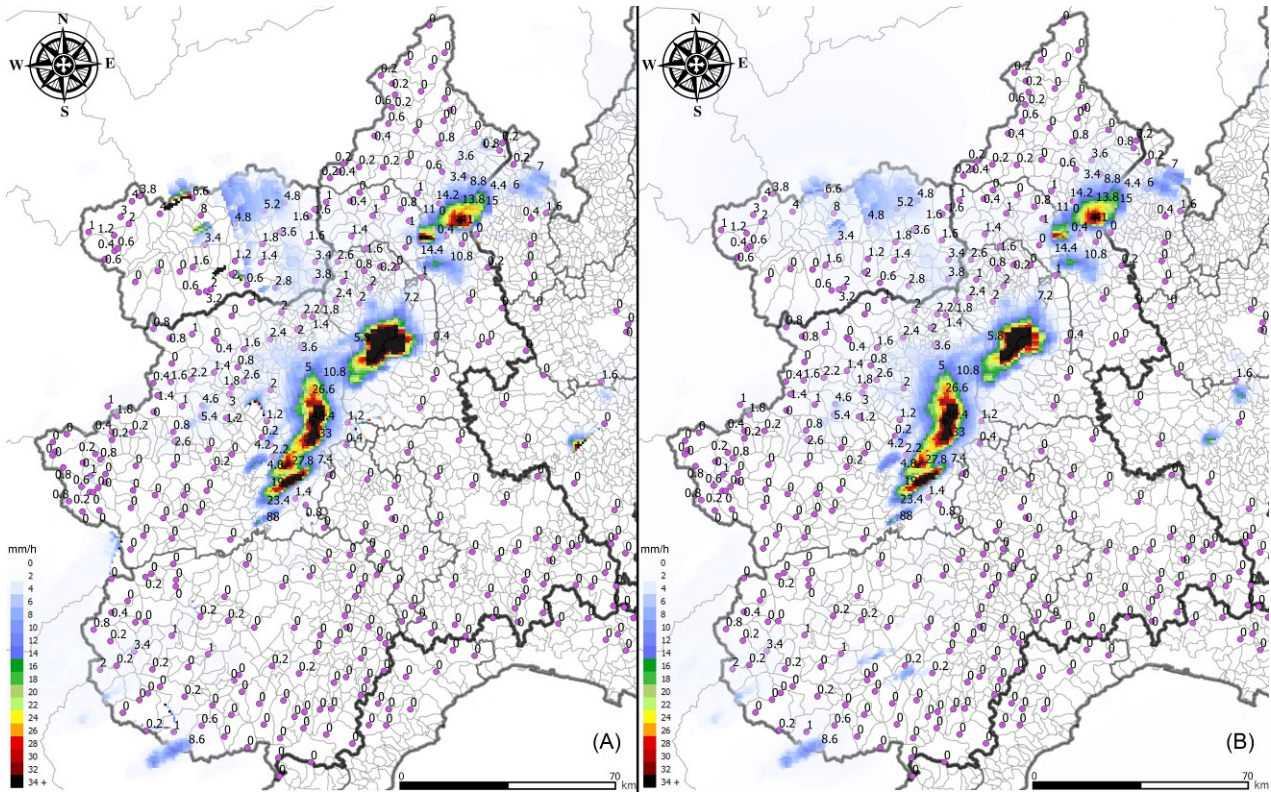
Fig. 5: experimental bias field conditional merging (A) versus experimental mean conditional merging (B) hourly precipitation data, loaded in QGIS with false colors (information coming from a rain event in the NW part of Italy, 28 July 2014 h15.00-16.00)

Analyzing these files is possible to conclude that, in this case study, every merging method computed better-fitting results than the raw radar data. For example, let us focus on the hourly information from 15.00 to 16.00 for the Piedmont region (see Fig. 1, Fig. 5 and Table 1). If we start considering errors (mean error, mean absolute error, relative mean absolute error and root mean square error), the conditional merging method offers the best results, with a reduction of errors that starts from 88.7% reaching 99.6% (nearly eliminating it). Obviously, these values are impressively high because they use the same dataset for training and test (the complete dataset cases discussed previously); if we look at conditional merging with k-fold cross-validation, improvements are still very interesting, starting from a reduction of errors of 10% and going up to 87.5%. A similar argument can be used for bias field conditional merging and for mean conditional merging (with some small exception for the mean error): in bias field conditional merging we also obtained some small increase in errors (from 1.5% to 8%) compared to raw data, but it showed also the best result for cross-validated methods regarding mean error. The other three best results regarding error indexes are obtained by mean conditional merging.

Moving to the analysis of the mean bias, we can always see that all merging methods offered better results compared to raw radar data, with conditional merging obtaining best ones. If we narrow the analysis just to the cross-validated methods, we have bias field conditional merging showing best results.

Concerning standard deviation, all merging methods offered better results than raw radar data but, in this case, the best is measured by conditional merging, both with the complete dataset and with cross-validated information.

The coefficient of determination showed that the best result is reached by conditional merging but analyzing only the cross-validated methods, mean conditional merging offered greater results. Again, we can see that nearly all computed values are better than raw radar.

In the end, we analyzed *A* and *B* coefficients of the regression. In this case, we noticed that *B* coefficient (which is less significant than the other one) shows that all merging methods have a regression that is very near the origin of the axis but, when we introduce cross-validation, this is not always true and in some cases like this, raw radar data could have a better value. Regarding the *A* parameter, instead, the global observations discussed here are true again, with conditional merging showing as the best method but leaving the first place to the bias field when cross-validation is introduced.

Algorithm	ME	MAE	RMAE	MB	RMSE	SD	R2	A	B
Raw radar data	-0,489	1,033	0,660	1,415	3,154	5,243	0,595	0,510	0,279
Conditional merging (k-fold)	0,061	0,929	0,593	0,945	2,783	5,053	0,667	0,657	0,598
Conditional merging	-0,002	0,115	0,073	0,974	0,356	5,072	0,996	0,955	0,068
Bias field conditional merging (k-fold)	0,006	1,049	0,670	1,046	3,417	5,070	0,549	0,716	0,451
Bias field conditional merging	-0,111	0,255	0,163	0,956	0,841	5,107	0,971	0,950	-0,032
Mean conditional merging (k-fold)	-0,072	0,872	0,557	1,161	2,492	5,094	0,737	0,686	0,420
Mean conditional merging	-0,061	0,177	0,113	1,054	0,511	5,091	0,990	0,953	0,013

Table 1: information obtained from XLSX file produced by CondMerg containing different indexes on hourly data from 28 July 2014, h15.00 to 16.00 for the Piedmont region. Analyzed indexes are mean error ME, mean absolute error MAE, relative mean absolute error RMAE, mean bias MB, root mean square error RMSE, standard deviation SD, coefficient of determination R2 and A and B parameters in simple linear regression y = Ax + B. These indexes are computed on raw radar information, conditional merging, bias field conditional merging and mean conditional merging (with both complete dataset and cross-validated data – indicated by k-fold string in parenthesis). In original XLSX file, best results of each index are highlighted in yellow (concerning the whole column) and in green (best result among cross-validated data).

6. Conclusions and future work

This study describes CondMerg, the first free and open source R program used to compute the conditional merging method starting from radar and rain gauges data. This program has been tested on some extreme events in the northwestern part of Italy and has shown ease of use, efficiency and quickness, requiring basic knowledge of R language. In addition to the conditional merging method, it computes two experimental variants of that algorithm that need further testing and validation, maybe on a larger number of events. In our tests, all three methods reported statistical values that were better than the ones calculated on the raw radar data, confirming that a merging step is

necessary to improve the rainfall field estimation from weather radar and gauges observations (Goudenhoofdt and Delobbe 2009). Being an open source software is very important for this program since it could be easily modified and improved directly from the community of users, fitting most needs of engineer, researchers and other professional figures who work on conditional merging. In this sense, users are encouraged to report bugs, feature requests or changes they made to the code to better fit their needs either directly to the corresponding author or by using GitHub platform. If the software becomes popular, it could also be refactored as an R package and submitted to CRAN for easier and broader adoption.

Appendix A. Supplementary material and software files

Supplementary data and the program associated with this article can be found both in the online version of the article at http://link.springer.com/article/10.1007/s12145-016-0278-y and at https://github.com/UniToDSTGruppoClima/CondMerg

References

Acquaotta F., Fratianni S., Venema V. (2016) Assessment of parallel precipitation measurements networks in Piedmont, Italy. International Journal of Climatology, doi: 10.1002/joc.4606

Allamano P., Croci A. and Laio F. (2015) Toward the camera rain gauge. Water resources research, 51:1744-1757 doi: 10.1002/2014WR016298

Arlot S. and Celisse A. (2010) A survey of cross-validation procedures for model selection. Statistics surveys, 4:40-79

ARPA Piemonte (2014) Rapporto tecnico eventi temporaleschi luglio 2014 (Italian only) Available online at https://www.arpa.piemonte.it/pubblicazioni-2/relazioni-tecniche/analisi-eventi/eventi-2014/rapporto-eventi-temporaleschi-luglio-2014/at_download/file Accessed on 19 June 2016

Barrett E. C. (1970) The estimation of monthly rainfall from satellite data. Monthly weather review, 98:322–327 doi: 10.1175/1520-0493(1970)098<0322:TEOMRF>2.3.CO;2

Borup M., Grum M., Linde J. J. and Mikkelsen P. S. (2015) Dynamic gauge adjustment of high-resolution X-band radar data for convective rain storms: model-based evaluation against measured combined sewer overflow. Journal of Hydrology (in review), 687-699 doi:10.1016/j.jhydrol.2016.05.002

Cole S. J. and Moore R. J. (2008) Hydrological modelling using raingauge and radar-based estimators of areal rainfall. Journal of Hydrology, 358:159-181 doi:10.1016/j.jhydrol.2008.05.025

Cremonini R. and Bechini R. (2003) Rainfall estimation in north-western Italy, using two polarimetric C-band radars and a dense real-time gauge network. 3rd GPM Workshop, Noordwijk

Cremonini R., Tiranti D., Barbero S. (2014) The urban flooding early warning system of the Greater Turin (north-western Italy) based on weather-radar observations. Engineering geology for society and territory, 5:837-842

Delobbe L., Bastin G., Dierickx P., Goudenhoofdt E., Leclercq G., Moens L. and Thunus M. (2008) Evaluation of several radar-gauge merging techniques for operational use in the Walloon region of Belgium. Proceedings of the International Symposium on Weather Radar and Hydrology, Grenoble

Doviak R. J. and Zrnic D. S. (1984) Doppler radar and weather observations. Academic Press, San Diego

Ehret U. (2002) Rainfall and flood nowcasting in small catchments using weather radar. PhD thesis, University of Stuttgart

Free Software Foundation (2007) GNU General Public License (GPL v3). Available online at http://www.gnu.org/licenses/gpl.html Accessed on 19 June 2016

Goormans T. and Willems P. (2013) Using Local Weather Radar Data for Sewer System Modeling: Case Study in Flanders, Belgium. Journal of Hydrologic Engineering 18:269-278 doi:10.1061/(ASCE)HE.1943-5584.0000589

Goudenhoofdt E. and Delobbe L. (2009) Evaluation of radar-gauge merging methods for quantitative precipitation estimates. Hydrology and Earth system sciences, 13:195-203 doi: 10.5194/hess-13-195-2009

Isaaks E. and Srivastava R. (1989) Applied geostatistics. Oxford University Press

Jordan P. W., Seed A. W. and Austin G. L. (2000) Sampling errors in radar estimates rainfall. Journal of geophysical research, 105:2247-2257

Journel A. G. and Huijbregts C. J. (1978) Mining geostatistics. The Blackburn Press, New York

Krige D. (1951) A statistical approach to some basic mine valuation problems on the Witwatersrand. Journal of the chemical, metallurgical and mining society of South Africa, 52:119-139

Legorgeu C., Pergaud J. and Wobrock W. (2012) A comparison of two techniques combining rain-gauge and radar measurements at high resolutions and its application to precipitation forecasts verification. 7th European Conference on Radar in Meteorology and Hydrology, Toulouse

Memarsadeghi N., Raykar V. C., Duraiswami R. and Mount D. M. (2008) Efficient kriging via fast matrix-vector products. Aerospace conference, doi: 10.1109/AERO.2008.4526433

Messer H., Zinevich A. and Alpert P. (2006) Environmental monitoring by wireless communication networks. Science, 312:713 doi: 10.1126/science.1120034

Oliver M. A. and Webster R. (1990) Kriging: a method of interpolation for geographical information systems. International journal of geographic information systems, 4:313-332 doi: 10.1080/02693799008941549

Pannatier Y. (1996) Variowin: Software for Spatial Data Analysis in 2D. Springer, New York doi:10.1007/978-1-4612-2392-4

Payam R., Lei T. and Huan L. (2009) Cross-validation. Encyclopedia of Database Systems, book B:532-538 doi: 10.1007/978-0-387-39940-9_565

R Development Core Team (2011) The R Project for Statistical Computing. Available online at https://www.r-project.org/ Accessed on 19 June 2016

Rabiei E., Haberlandt U., Sester S. and Fitzner D. (2013) Rainfall estimation using moving cars as rain gauges - Laboratory experiments. Hydrology and Earth system sciences, 17:4701–4712 doi: 10.5194/hess-17-4701-2013

Sen P. K. (1968) Estimate of the regression coefficient based on Kendall's tau. Journal of the American statistical association, 63:1379-1389

Sinclair S. and Pegram, G. (2005) Combining radar and rain gauge rainfall estimates using conditional merging. Atmospheric science letters, 6:19-22

Theil H. (1950) A rank-invariant method of linear and polynomial regression analysis. Indigationes mathematicae, 12:85-91

Vignal B. and Krajewski W. F. (2001) Large-sample evaluation of two methods to correct range-dependent error for WSR-88D rainfall estimates. Journal of hydrometeorology, 2:490-504

Zandonadi L., Acquaotta F., Fratianni S. and Zavattini J. A. (2016) Changes in precipitation extremes in Brazil (Paraná River Basin). Theoretical and applied climatology, 123:741-756

Supplementary material captions

Online Resource 1: file *MultiCore_CondMerg.R*, containing the program discussed in this article

Online Resource 2: file *RainGauges_2014-07-28_daily_Piedmont.csv*, containing daily rain gauges data coming from the extreme event occurred in the Piedmont region (NW part of Italy) on 28 July 2014

Online Resource 3: file *RainGauges_2014-07-28_h14-15_TurinOnly.csv*, containing hourly rain gauges data coming from the extreme event occurred near Turin (NW part of Italy) from 14.00 to 15.00 of 28 July 2014

Online Resource 4: file *RainGauges_2014-07-28_h15-16_Piedmont.csv*, containing hourly rain gauges data coming from the extreme event occurred in Piedmont Region (NW part of Italy) from 15.00 to 16.00 of 28 July 2014

Online Resource 5: file *Radar_2014-07-28_daily_Piedmont.tif*, containing daily radar data coming from the extreme event occurred in the Piedmont region (NW part of Italy) on 28 July 2014

Online Resource 6: file *Radar_2014-07-28_h14-15_TurinOnly.tif*, containing hourly radar data coming from the extreme event occurred near Turin (NW part of Italy) from 14.00 to 15.00 of 28 July 2014

Online Resource 7: file *Radar_2014-07-28_h15-16_Piedmont.tif*, containing hourly radar data coming from the extreme event occurred in Piedmont Region (NW part of Italy) from 15.00 to 16.00 of 28 July 2014

Online Resource 8: file *Case study 2014-07-28 and 29 event.gif*, containing the animation of the Airmass RGB satellite data from Eumetrain (http://www.eumetrain.org) about the extreme event of 28 and 29 July 2014