Geophysical Research Abstracts
Vol. 16, EGU2014-6308, 2014
EGU General Assembly 2014

wradlib - an Open Source Library for Weather Radar Data Processing

Thomas Pfaff (1), Maik Heistermann (2), and Stephan Jacobi (2)

(1) University of Stuttgart, Institute for Modeling Hydraulic and Environmental Systems, Department of Hydrology and Geohydrology, Stuttgart, Germany (thomas.pfaff@iws.uni-stuttgart.de), (2) University of Potsdam, Institute of Earth and Environmental Sciences, Potsdam, Germany

Even though weather radar holds great promise for the hydrological sciences, offering precipitation estimates with unrivaled spatial and temporal resolution, there are still problems impeding its widespread use, among which are:

- almost every radar data set comes with a different data format with public reading software being available only rarely.
- standard products as issued by the meteorological services often do not serve the needs of original research, having either too many or too few corrections applied. Especially when new correction methods are to be developed, researchers are often forced to start from scratch having to implement many corrections in addition to those they are actually interested in.
- many algorithms published in the literature cannot be recreated using the corresponding article only. Public codes, providing insight into the actual implementation and how an approach deals with possible exceptions are rare.
- the radial scanning setup of weather radar measurements produces additional challenges, when it comes to visualization or georeferencing of this type of data.

Based on these experiences, and in the hope to spare others at least some of these tedious tasks, *wradlib* offers the results of the author's own efforts and a growing number of community-supplied methods.

wradlib is designed as a Python library of functions and classes to assist users in their analysis of weather radar data. It provides solutions for all tasks along a typical processing chain leading from raw reflectivity data to corrected, georeferenced and possibly gauge adjusted quantitative precipitation estimates. There are modules for data input/output, data transformation including Z/R transformation, clutter identification, attenuation correction, dual polarization and differential phase processing, interpolation, georeferencing, compositing, gauge adjustment, verification and visualization.

The interpreted nature of the Python programming language makes *wradlib* an ideal tool for interactive data exploration and analysis. Based on the powerful scientific python stack (numpy, scipy, matplotlib) and in parts augmented by functions compiled in C or Fortran, most routines are fast enough to also allow data intensive re-analyses or even real-time applications.

From the organizational point of view, *wradlib* is intended to be community driven. To this end, the source code is made available using a distributed version control system (DVCS) with a publicly hosted repository. Code may be contributed using the fork/pull-request mechanism available to most modern DVCS. Mailing lists were set up to allow dedicated exchange among users and developers in order to fix problems and discuss new developments.

Extensive documentation is a key feature of the library, and is available online at http://wradlib.bitbucket.org. It includes an individual function reference as well as examples, tutorials and recipes, showing how those routines can be combined to create complete processing workflows. This should allow new users to achieve results quickly, even without much prior experience with weather radar data.