Hydrol. Earth Syst. Sci. Discuss., 9, C6400–C6404, 2013
www.hydrol-earth-syst-sci-discuss.net/9/C6400/2013/

Hydrology and Earth System Sciences Discussions

Interactive comment on "Technical Note: An open source library for processing weather radar data (*wradlib*)" *by* M. Heistermann et al.

M. Heistermann et al.

heisterm@uni-potsdam.de

Received and published: 18 January 2013

Dear Referee,

Thanks a lot for your response to our *"Technical Note: An open source library for processing weather radar data (wradlib)"* and for the many helpful comments. On the following pages, we would like to reply to the major comments (all comments related to typos and other minor issues have been dealt with in the revised manuscript).

Before, though, we would like to give one general response on some of the referee comments: Both referees asked, for a couple of aspects, for more detailed information, e.g. in terms of example code, technical guidance, or with respect to data formats.

Although the paper is submitted as a Technical Note, we are hesitant to overload it with technical detail, as it still should be brief and concise, and, most important, not be mistaken as a tutorial or quick start manual. The example code shown in the manuscript is rather intended to give a "feel" of the library usage, not to provide complete workflows. On the other hand, we agree that the information requested by the reviewers could be helpful to the reader and to potential users of the wradlib library. Thus, for some of the issues raised by the referees, we included the requested information on the wradlib documentation page (http://wradlib.bitbucket.org), mostly as topical tutorials or recipes. Please consider this only as a general remark. The specific suggestions will be found within the replies to the comments below. Please note that the referee comments are shown in **bold format**.

We believe that the revised manuscript, together with the enhanced wradlib documentation pages, is improved as compared to the first version and we are very grateful for the advice given. We hope that the present quality of the manuscript, together with the following replies, meets the standards of HESS and can now be considered for publication.

Sincerely Yours,

Maik Heistermann (on behalf of the co-authors)

At the end of the introduction BALTRAD is mentioned. I think the authors should comment here how wradlib is different, and what sets it apart from BALTRAD. The text at the beginning of Section 4 could be (partially) used for this.

We agree with the referee that a systematic comparison with BALTRAD would be interesting. However, we deliberately decided not to include such a comparison in the initial manuscript because we suspected such a comparison would be premature as both wradlib and BALTRAD are quite "young" and dynamic. Once such a comparison is published, though, it might have quite a half-life. A good example is the data formats

accepted by BALTRAD and wradlib: At the moment, BALTRAD only accepts ODIM-H5 while wradlib provides interfaces for other file formats, too. But this can change quickly on the BALTRAD side. So at this point, we would rather suggest not providing an enhanced comparison. However, if the Editor or the referee insists, we are willing to add a paragraph with a more in-depth comparison.

In Section 3, the authors should consider indicating where a complete list of available wradlib functions can be found.

In the initial manuscript, we referred to the wradlib documentation page in section 2.4 and at the end of section 4. This documentation includes a full *library reference* (i.e. documentation of the wradlib API). In the revised version, we will additionally point out this information in the introduction of section 3, as suggested by the referee.

In Section 3 there should be a reference to the wradlib manual.

As pointed out above, the documentation pages (i.e. manual) are already mentioned in sections 2.4 and 4. We think this should be sufficient.

On p.12340, lines 7-8, the authors should elaborate on the metadata that is included. Is there a standard set of metadata, does this depend on the data format, or something in between?

There is no standard metadata set returned by wradlib. The metadata returned by wradlib depends on the metadata provided with the data. wradlib tries to extract and return all available metadata, e.g. from BUFR, hdf5 or NetCDF attributes or file headers of other formats. wradlib returns the metadata as Python dictionaries which can be browsed and inspected using keywords. In the revised manuscript, we will add one sentence to explain that the returned metadata depends on the data source.

In Section 3.1, reading of radar data is discussed. It is stated that wradlib can

read both polar and Cartesian data. Can it also read multi-scan volume data files?

Basically, multi-scan volume data can be encoded in a variety of formats. This does not necessarily imply that the data has to come in one file. A typical approach would be to store the data for each elevation angle in a separate file which could be BUFR, hdf5, NetCDF or any other format. Reading volume data would then mean to read the scans for each available elevation angle and to construct from those two-dimensional polar data a three dimensional polar data array with elevation, azimuth and range as axes. Of course, polar volume data can also be stored in single files, e.g. following OPERA/ODIM conventions for BUFR or hdf5. However, the ODIM-H5 convention for example also implies that scans at different elevation angles are stored in different datasets, although the tree is *inside the file* and not in the *file system*. It will always depend on the user's preferences how to combine the data from different elevation angles to one polar volume. However, we agree that the user should be guided on how to construct three dimensional arrays from polar volumes which can e.g. be processed by the wradlib.vpr module. For this purpose, we added a recipe "Reading polar volume data" to the wradlib documentation page on http://wradlib.bitbucket.org/recipes.html.

On p.12340, line 20, the module wradlib.vpr is mentioned. However, on p.12341, lines 11-12, it is stated that this module does not exist yet. I think that this should be made clear on p.12340 as well.

In the meantime, we enhanced the module wradlib.vpr with functions and classes to create 3-dimenisonal Cartesian representations from polar volume data and to create CAPPIs and Pseudo-CAPPIs (an example is given under http://wradlib.bitbucket.org/recipes.html); however, implementations for VPR *correction* are still underway. In the revised manuscript, we have pointed this out more clearly.

Would it be possible to include the "recipe" (Example code) for making Figures

3 and 4 in Section 3.5?

Including the example code for these figures in the manuscript itself would – in our opinion – be rather confusing than helpful. However, we added a corresponding recipe for Figure 4 in the recipes section of the wradlib documentation pages, http://wradlib.bitbucket.org/recipes.html). This partly coincides with the efforts in response to referee 2 who suggested publishing a full worked example. Figure 3 results from an application building on wradlib and it will not be too illustrative to provide the underlying code.

Sections 3.6 and 3.7 deal with rain gauge data. Is wradlib able to read gauge data. If so, in which formats?

Currently, wradlib requires rain gauge data for rain gauge adjustment (wradlib.adjust) and for verification (wradlib.verify). However, there is no standard rain gage data format expected by wradlib since the spate of formats is even more overwhelming for rain gauge data than it is for weather radar data. Rain gauge data often come in some sort of ASCII table format which can be read by several numpy functions such as "genfromtxt" or "loadtxt". Rain gauge data in NetCDF or hdf5 format can be easily read by using the corresponding Python modules. We would rather not include these technical details in the manuscript, though. But if the referee knows a standard for rain gauge data which he considers as worth being included in wradlib, we will try to include an interface to that format.

Interactive comment on Hydrol. Earth Syst. Sci. Discuss., 9, 12333, 2012.