



WRaINfo: An Open Source Library for Weather Radar INformation for FURUNO Weather Radars Based on Wradlib

SOFTWARE
METAPAPER

ALICE KÜNZEL

KAI MÜHLBAUER

JULIA NEELMEIJER

DANIEL SPENGLER

*Author affiliations can be found in the back matter of this article

u[ubiquity press

ABSTRACT

WRaINfo is a software for real-time weather radar data processing developed by the Helmholtz Innovation Lab FERN.Lab, a technology and innovation platform of the German Research Centre for Geosciences Potsdam (GFZ). WRaINfo is specifically designed for processing X-band weather radar data of FURUNO devices. The modules of the package allow to read and process raw data of the WR2120 and WR2100. For this purpose, many functions of the library wradlib are used and adapted. The processing is controlled by a configuration file, main functionalities include formatting, attenuation correction, clutter detection, georeferencing and gridding of the data. This allows the construction of reproducible, automatic data processing chains. The package is written in the Python programming language. The source code is publicly available on GitLab. Compiled versions are also available on PyPi. The package is distributed under the Apache 2.0 license.

CORRESPONDING AUTHOR:

Alice Künzle

Helmholtz-Centre Potsdam,
German Research Centre for
Geosciences (GFZ), Potsdam,
Germany

alicek@gfz-potsdam.de

KEYWORDS:

WRaINfo; python; weather radar; X-band; quality control; real-time; data processing; precipitation estimation; attenuation correction; clutter detection; georeferencing and gridding; meteorology; remote; sensing

TO CITE THIS ARTICLE:

Künzel A, Mühlbauer K, Neelmeijer J, Spengler D 2023 WRaINfo: An Open Source Library for Weather Radar INformation for FURUNO Weather Radars Based on Wradlib. *Journal of Open Research Software*, 11: 9. DOI: <https://doi.org/10.5334/jors.453>

(1) OVERVIEW

INTRODUCTION

Raw data of weather radars can provide useful visual information about the spatial distribution of precipitation events. But in order to use weather radar data for quantitative studies, the raw data have to be processed in order to account for typical error sources such as ground clutter, uncertainties in polarimetric variables and in the relationship between radar reflectivity and rain rate as well as attenuation of the radar signal.

A ground-based weather radar not only detects reflectivities of water droplets in the atmosphere but also of all non-meteorological objects. These include wind parks, buildings, mountains, aircraft and bird flocks. These echoes are called (ground) clutter and must first be removed from the raw data. For this purpose, clutter maps are created over a long period of time to remove the non-meteorological echoes from the raw data.

The radar signal may be attenuated by heavy precipitation, so that only weak reflectivities are detected within the heavy rain cell and behind it. To correct this, an attenuation correction must be performed.

Weather radars don't measure precipitation directly. Thus, the amount of precipitation (R) is estimated from the clutter and attenuation corrected reflectivity (Z). The relationship is called "Z-R Relationship".

The software package WRaINfo has been developed on top of the software package wradlib [1] to process and correct all formerly effects of data originating from the ground-based dual-polarized X-band weather radar system (WR2120) of FURUNO.

Clutter detection, attenuation correction and precipitation estimation are based on polarimetric variables that are recorded in addition to reflectivity by dual-polarized weather radar systems. Table 1 provides a summary about the characteristics of measured variables.

DATA FORMAT

Per default, FURUNO weather radars generate binary files. For each elevation angle one scnx (WR2120)/scn (WR2100) file per scan interval is generated. The scnx/scn files are provided with a 16-bit resolution.

COMPARISON TO EXISTING SOFTWARE

WRaINfo heavily depends on functionalities implemented in wradlib [3], which is a package providing many functions for processing weather radar data in general. WRaINfo, however, has been developed to handle data coming particularly from FURUNO weather radar devices. This allows to reuse functionalities like clutter detection, precipitation estimation and georeferencing from wradlib. Additionally, a specific function for reading the binary raw data (scn/scnx data) of FURUNO has been implemented in wradlib and is used in WRaINfo.

WRaINfo incorporates X-band specific attenuation correction methods, as the implementation in wradlib is dedicated for C-band weather radars [4–6], which are found not to be suitable for X-band devices. As a consequence, the Z-PHI method [7–9] was implemented in WRaINfo for attenuation correction. To enable completely automated processing, process chains for the generation of clutter maps and precipitation products from the FURUNO raw data were also integrated in WRaINfo.

POLARIMETRIC VARIABLE	DEFINITION	SHORT CHARACTERISTICS	APPLICATION
ZH Horizontal reflectivity	reflectivity in the horizontally polarized channel	Increases with horizontal particle dimension, number concentration, density and water content.	–
ZV Vertical reflectivity	reflectivity in the vertical polarized channel	Increases with vertical particle dimension, number concentration, density and water content.	–
ZDR Differential reflectivity	difference between ZH and ZV	Increases with density, flattening and water content and is proportional to the hydrometeor concentration.	hydrometeor classification, clutter detection
PHIDP Differential phase shift	cumulative phase shift between H and V polarized waves	Is a measure of the attenuation integrated along the path.	attenuation correction, precipitation estimation
RHOHV Correlation coefficient	correlation coefficient between ZH and ZV	Is a measure of the diversity of hydrometeors in the radar volume and the data quality. RHOHV is close to 1 for most hydrometeors and significantly less than 1 for non-meteorological scatters.	clutter detection, hydrometeor classification
KDP Specific differential phase	derivative of PHIDP	Increases with density, flattening and water content and is proportional to the hydrometeor concentration.	attenuation correction, precipitation estimation

Table 1 Short characteristics of polarimetric variables [2].

Due to the development of xradar [10], other software packages can also be integrated in the future, so that tools from the Python ARM Radar Toolkit [11] can also be implemented in the future.

WRaINfo has reached a mature stage in the last months, which allows the operational use of the FURUNO weather radar devices. The software can be used for both radar types (WR2120 & WR2100) of FURUNO.

OVERVIEW OF WRAINFO

In summary WRaINfo provides the following functionality:

- raw data reading and returned as xarray dataset
- optional: HDF5 data reading
- application of clutter detection, phase processing and attenuation correction with optimized parameters for X-Band weather radars
- estimation of precipitation amounts with Z-R Relationship
- georeferencing and gridding of data
- operational use: processing chain to generate clutter maps over long period of time followed by entire data processing

The software has been developed by the Helmholtz Innovation Lab FERN.Lab [12], the technology and innovation platform for remote sensing for a sustainable use of resources at GFZ Potsdam, in cooperation with the University of Bonn. The following goals are pursued:

1. Testing the functionality of the software by increasing the coverage of unit tests.
2. Contribute to wradlib so that the software developed by FERN.Lab can be made available to a wider audience.
3. Facilitate the use of weather radar data for farmers based on WRaINfo.

IMPLEMENTATION AND ARCHITECTURE

Used programming language

To allow the close connection to wradlib, WRaINfo has been developed in Python 3.6, which is a widely used programming language and has a large standard library, supplemented by a variety of open source packages. Developments have been embedded within the open source cookiecutter python package template [13]. With this template it is possible to automatically generate software documentation based on Sphinx [14] and to perform quality control of the software using Pylint [15] and pytest. WRaINfo was created using the cookiecutter template adapted by FERN.Lab [16].

Configuration

The package uses a configuration file. The entire contents of this file can be loaded with the `wrainfo.reader.read_config_file()` function. For this purpose, the file path to

the configuration file and “selection = None” must be specified. Before the other modules of WRaINfo can be used, all directories and settings in the configuration file have to be defined. Table 2 contains an overview of the configuration file and a description of the parameters to be defined.

Modules

WRaINfo contains eight modules, which are subdivided according to the respective functionality (Figure 1). The individual modules are described below.

a) Reader

The reader module (`reader.py`) contains functions to read the configuration file and generate file lists. The function `wrainfo.reader.create_list()` loads FURUNO raw data for a defined period of time into a list. The file list is compared with an error file list. The error file list contains manually defined erroneous files and these are removed from the created file lists in order to exclude these data from the processing. The two individual file lists are merged into one. Per default, `scn/scnx` format is supported as input data format, optionally data can be provided in HDF5 format. A file list with all raw data of the defined period is returned.

`Scn/scnx` files are processed by the function `wrainfo.reader.read_single_file()`. A raw dataset is read and returned as an xarray dataset. Figure 2 shows the most important polarimetric variables of a FURUNO raw dataset, which are read by the function.

Raw reflectivity (ZH) provides information on the spatial distribution of a recorded precipitation event extending from north to south-east (Figure 2).

By means of the polarimetric variable RHOHV, non-meteorological (ground clutter) and meteorological echoes can be distinguished in the raw reflectivity (ZH). If the cross correlation coefficient (RHOHV) is greater than 0.9, it is assumed that mostly precipitation occurs [17]. Thus, Figure 2 shows that the individual pixels with high reflectivity, such as in the west and south, are ground clutter. KDP allows conclusions to be drawn about the size and concentration of raindrops. Thus, high positive values in KDP correlate with high reflectivity, which can be seen in the north (Figure 2).

The polarimetric variable PHIDP is a measure of the attenuation. Strong attenuation is visible in the north (Figure 2). Even if the attenuation is not strong, an attenuation correction must be made.

b) Clutter detection

The clutter detection module (`clutter.py`) contains functions to detect clutter and then remove it from the raw data. Furthermore, a clutter map can be created with this module.

A clutter map can be generated using the `wrainfo.clutter.create_clutter_map_sequential()` function. To do this, a file list must first be created that contains all raw

PARAMETER	DESCRIPTION	DATA FORMAT [RANGE IF NECESSARY]	DEFAULT VALUES (IF NECESSARY)
Read data			
raw_data_directory	path to raw data	str	-
subfolder_structure_raw_data	regex format of subfolders in the raw data directory	str	Y/m/d/
error_flist_directory	path to file list with faulty files	str	-
monthly_clutter_directory	path to monthly clutter maps	str	-
subfolder_structure_clutter_directory	date format of subfolders in the monthly clutter directory	str	Y/m/
static_clutter_directory	path to the static clutter maps	str	-
static_cmap	complete file path to the selected clutter map		-
Save data in output directory			
output_directory_tar_gz_files	output directory for compressed raw data	str	-
output_path_processed_files	output directory for processed data	str	-
output_path_error_flist	output directory for file list with faulty files	str	-
Settings for processing data			
radar_location_identifier	Abbreviation for location and type of weather radar	str	-
epsg_code	EPSG Code of the desired coordinate reference system	integer	32633
nb_pixels	Resolution in pixels of the georeferenced data	integer	1400
dimensions	dimension of the data, two or three dimensional	list	["azimuth", "range", "time"]
scan_interval	scan interval of the weather radar in seconds	integer	300
re_index_parameters	Correction of the original azimuth angles	list [0–360°]	[0.25, 360, 0.5]
moments_in_processed_files	Moments which are to be stored in the processed data	list	['RATE', 'DBZH', 'VRAD', 'ZDR', 'KDP', 'PHIDP', 'RHOHV', 'WRADH', 'FUZZ', 'DBZH_no_clutter', 'DBZH_CORR', 'PREC_ZR']

Table 2 Description of the parameter used in the configuration file.

data over a defined time period. This can be done using the `wrainfo.reader.create_filelist()` function.

The created file list is one of the function parameters. The parameter `threshold` specifies that only positive values of the uncorrected reflectivity, which are in the boolean dataset of the clutter map, are considered. If the status parameter is set to true, then the processing progress of the clutter map is obtained as a percentage. Only the reflectivity values are extracted from the raw data and added. Ground clutter is located at the points where high reflectivities occur in each dataset. Afterwards a boolean dataset is created where the added reflectivity values must be greater than the threshold. The clutter maps generated with WRaINfo contain a dataset with the summed reflectivities and another boolean dataset (1 = clutter, 0 = no clutter).

Figure 3 shows a clutter map for the elevation angle of 0.5°. Pixels which are identified as clutter, are marked in yellow. These are wind parks and high buildings in the town, where the weather radar is located.

For clutter detection the fuzzy echo classification from `wradlib` is used [18, 19]. The function `wrainfo.clutter.fuzzy_echo_classification()` needs as input a two dimensional (azimuth and range) raw dataset of the weather radar and the clutter map as xarray dataset. The raw dataset may additionally contain time as a dimension. Then the parameter `"dims"` must be adjusted accordingly.

For the fuzzy echo classification all polarimetric variables (RHOHV, PHIDP, ZDR, VRAD) as well as the generated clutter map (CMAP) are used as described in Vulpiani et al. [19].

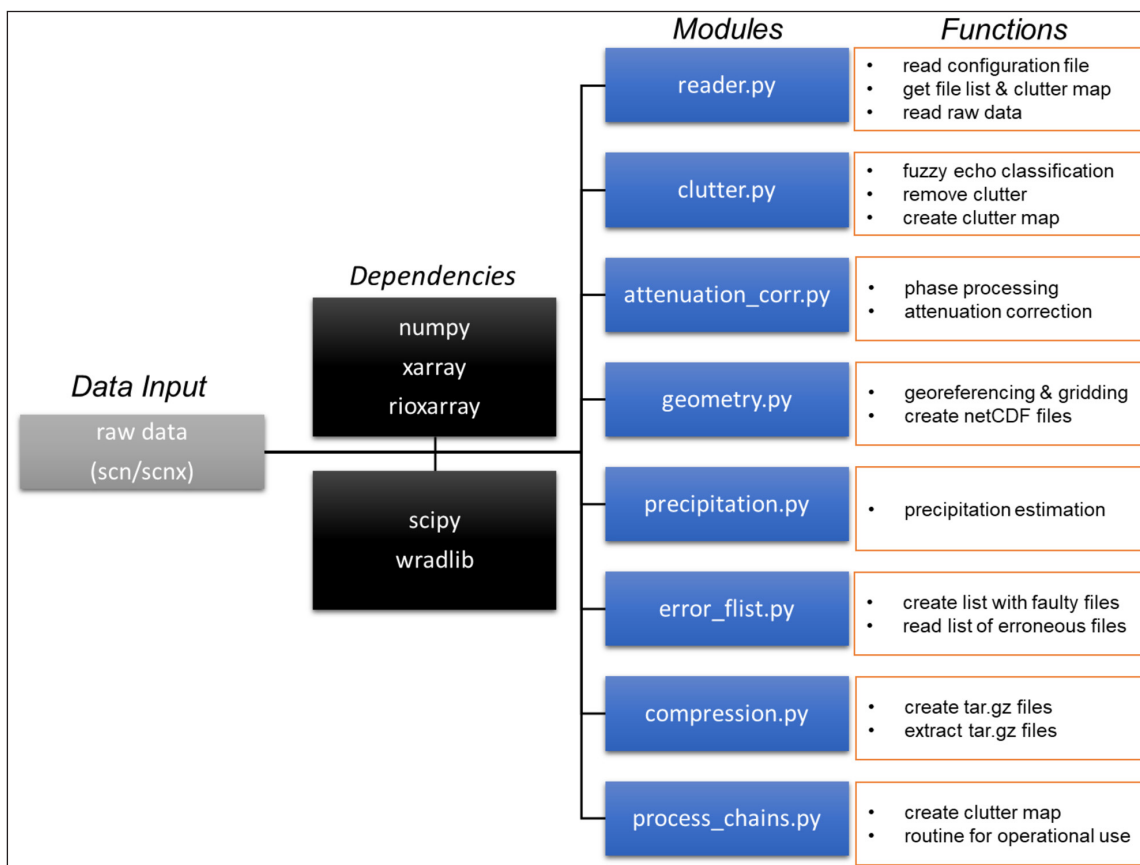


Figure 1 Software architecture of WRaInfo.

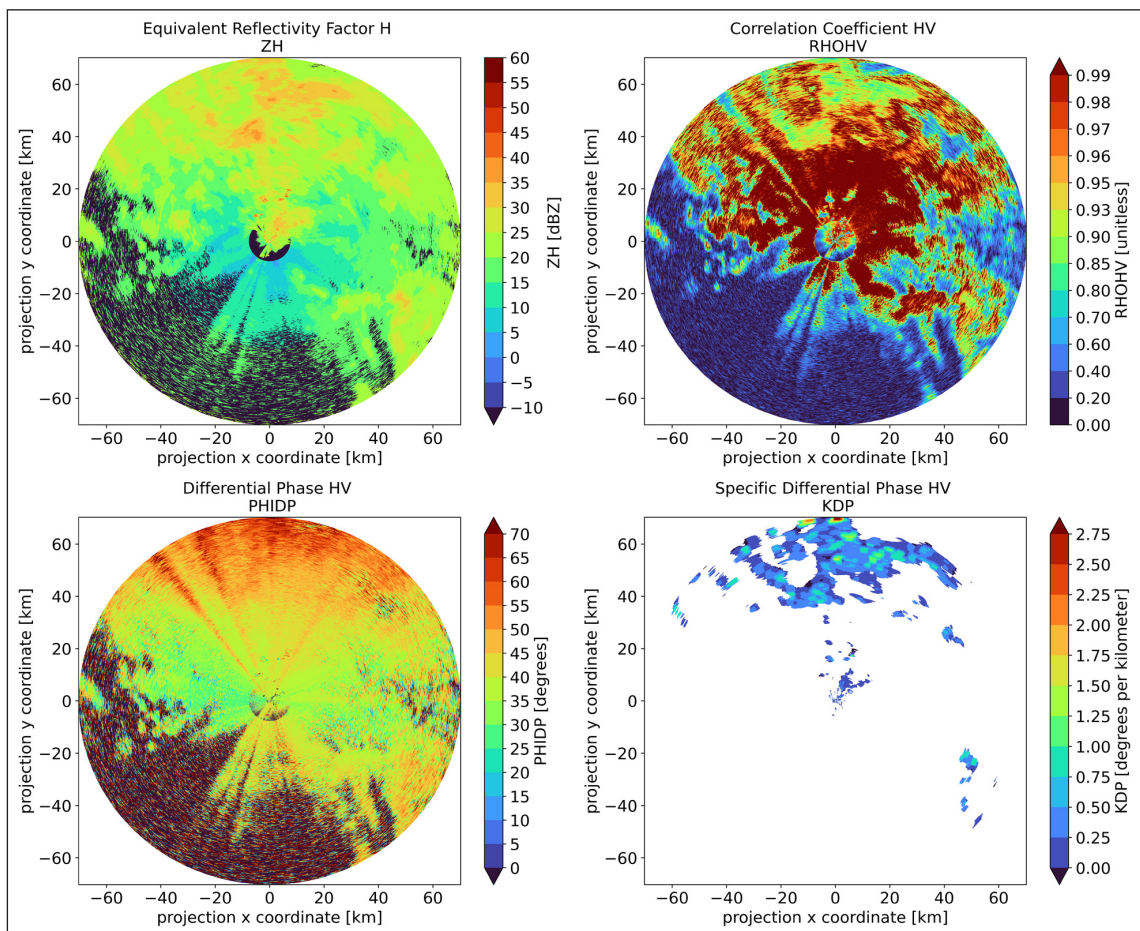


Figure 2 Polarimetric variables of raw data.

The default values for the weighting of each variable correspond to the study of Crislogo et. al [18] and can be found in Table 3.

The new variable “FUZZ” is appended to the raw dataset. This is the clutter map resulting from the fuzzy echo classification and is needed for the function *wrainfo.clutter.dbzh_no_clutter()*. Figure 4 shows the clutter map and results of the fuzzy echo classification. All radar bins that have been classified as clutter are now marked in yellow.

POLARIMETRIC VARIABLE	WEIGHT
RHOHV	0.4
PHIDP	0.1
ZDR	0.4
VRAD	0.1
CMAP	0.5

Table 3 Weights of the fuzzy echo classification procedure.

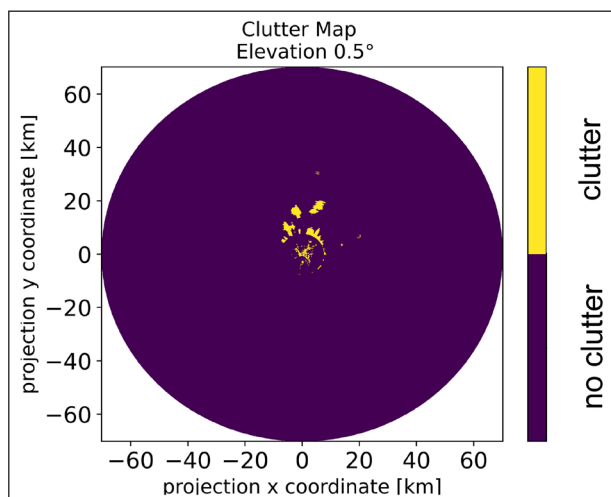


Figure 3 Clutter map.

To remove the pixels masked as ground clutter from the raw reflectivity, the *wrainfo.clutter.dbzh_no_clutter()* function is used. The output dataset of the fuzzy echo classification is needed for this function. In addition, the static clutter map has to be loaded. The ground clutter is excluded from the raw reflectivity using the clutter map and the previously created variable “FUZZ”. All pixels that have been identified as clutter are flagged as NaN values. Figure 5 shows the raw reflectivity and the clutter corrected reflectivity. The returned xarray dataset gets a new variable “DBZH_no_clutter”.

c) Attenuation correction

After the clutter detection is done, the attenuation correction follows. This is accomplished by the *wrainfo.attenuation_corr.phase_zphi()* function and processes the phase PHIDP. The function needs the data array of PHIDP from the clutter corrected xarray dataset. The parameters “start_range” and “range” of the moving window have to be defined. Within the function a binary array (True = rain, False = not-rain) rolling a range-long sum from PHIDP is generated. Afterwards the first occurrence of the maximum (from front and back) of PHIDP binary array is searched for. This gives the indices of the centres of the rolling window. Then min, max, median and mean of these are calculated. Finally, the true start and end indices of the phase PHIDP are calculated (+/- half window length). The returned dataset additionally contains the first and last values (min, max, median, mean) of the phase PHIDP (Figure 6).

In the Z-PHI method, the rain profile along the radar beam is derived from the reflectivity profile (ZH). The required boundary condition is the total attenuation of the radar beam, which is derived from PHIDP.

In the following function *wrainfo.attenuation_corr.zphi()* the specific attenuation (AH) is calculated using an optimized algorithm based on the ZPHI-Method [7,

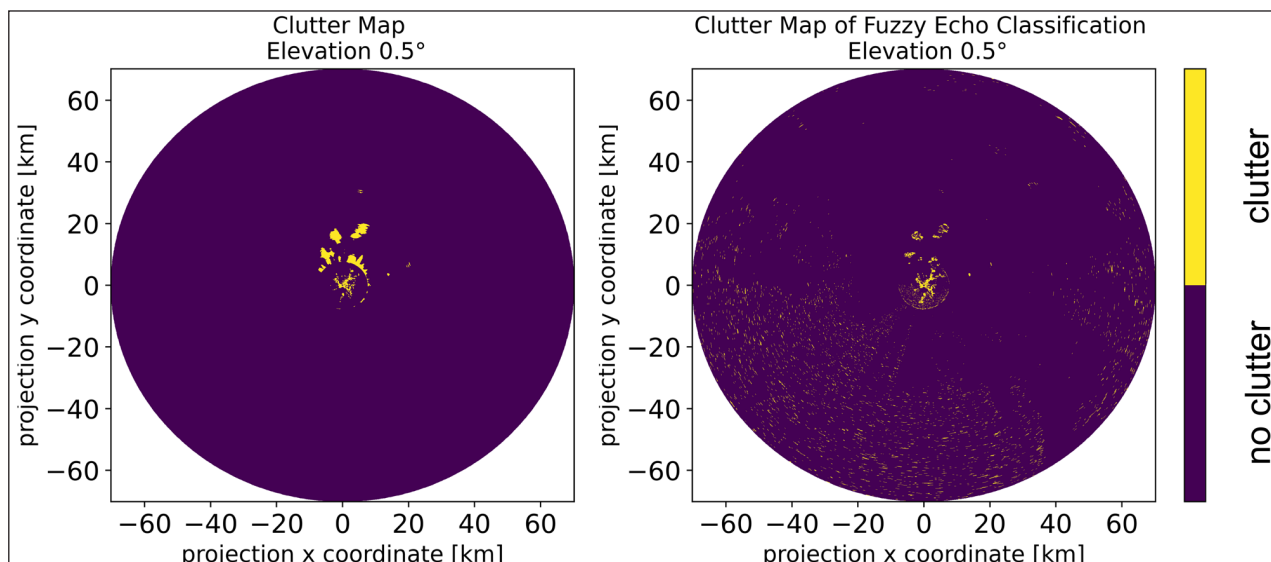


Figure 4 Comparison of the clutter map and results of the fuzzy echo classification.

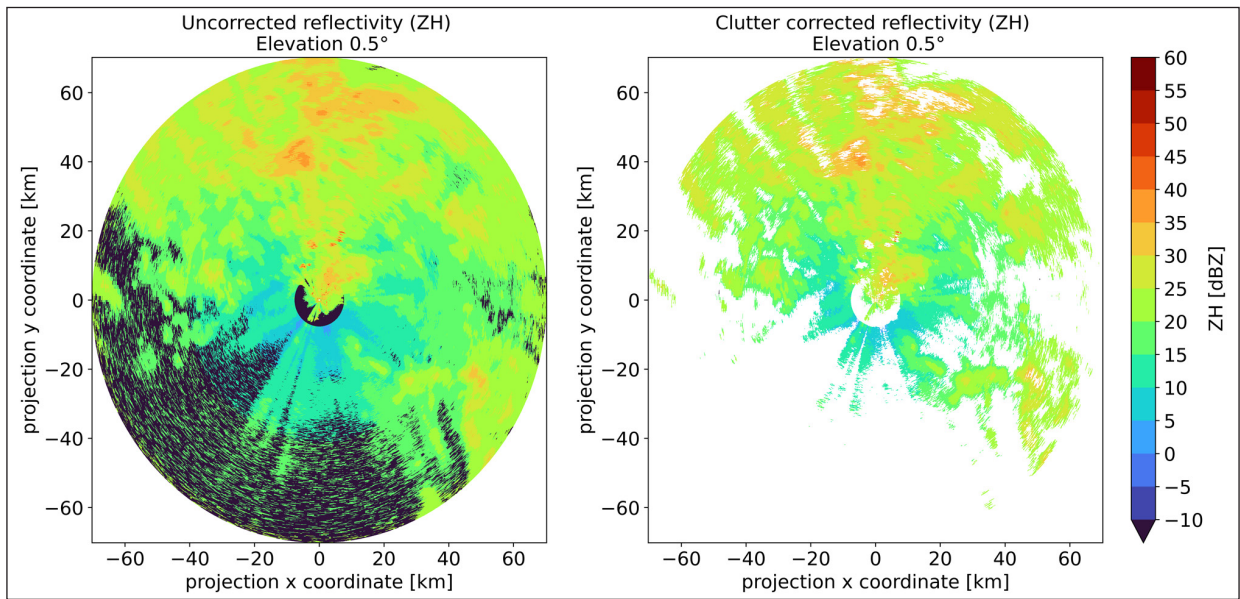


Figure 5 Comparison of raw reflectivity and clutter corrected reflectivity.

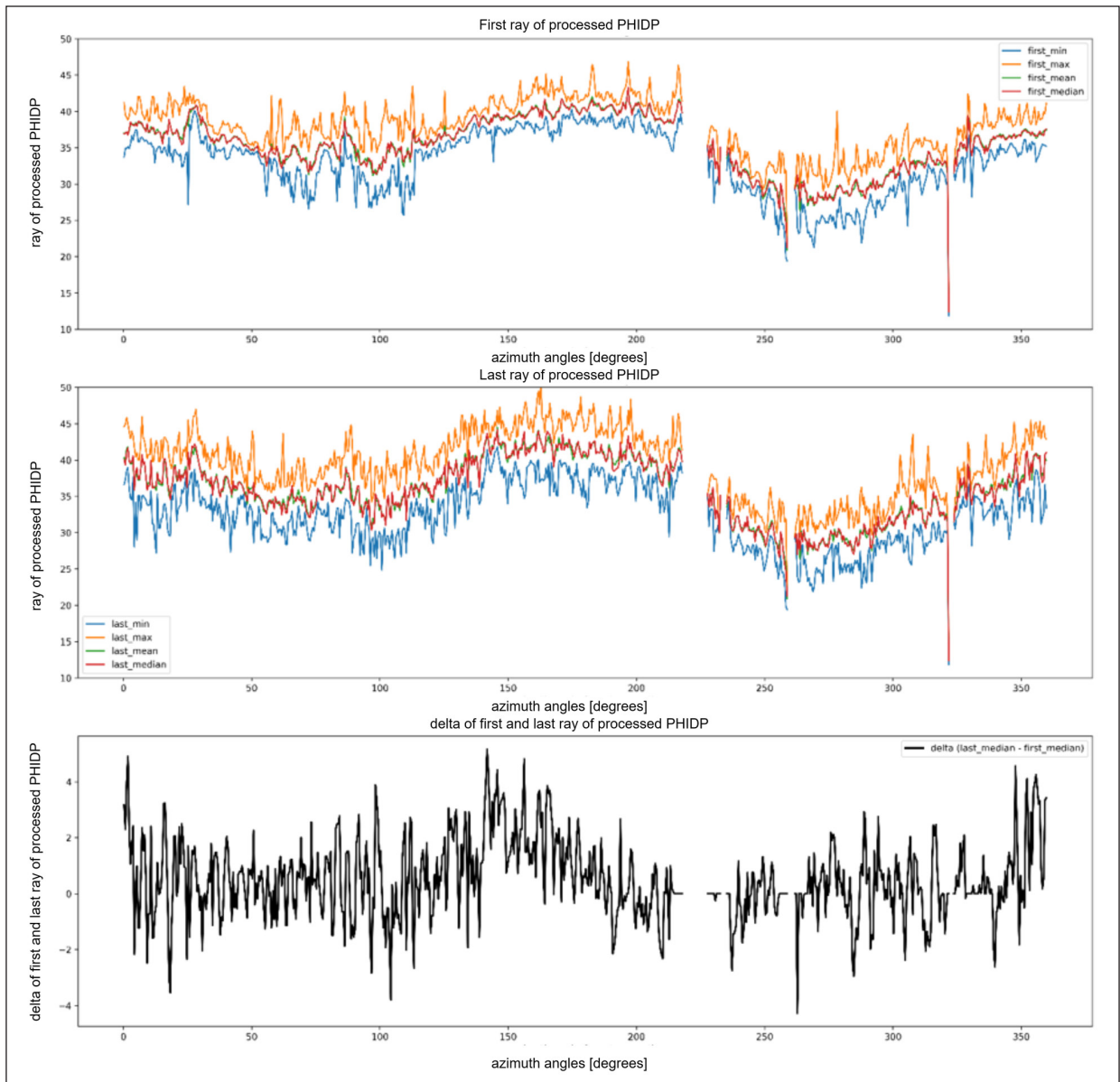


Figure 6 Results of the processed phase PHIDP.

8]. The function needs the clutter corrected reflectivity and the added dataset of the function *wrainfo.attenuation_corr.phase_zphi()* as function parameters. To define the boundary conditions, within the function *wrainfo.attenuation_corr.zphi()* first $\Delta\Phi_{DP}$ is calculated (eq. 1) from the parameters “last_median” and “first_median” (Figure 7). In eq. 1 r_1 and r_2 represent the integration bounds.

These are contained in the returned xarray dataset of the function *wrainfo.attenuation_corr.phase_zphi()*. First any $\Delta\Phi_{DP}$ values less than 0 are set to 0. Then the function of $\Delta\Phi_{DP}$ ($f\Delta\Phi_{DP}$) is calculated (eq. 2). According to eq. 3 we prepare attenuated reflectivity to the power of $b = [Z_a(r)]^b$, where b is a constant parameter (usually within 0.6–0.9 at microwave frequencies) and the integral of that from each range bin to maximum range [9]. The calculation of that integral is done per ray by cumulative trapezoidal integration over the range dimension and following subtraction from its maximum value. α is calculated as $\alpha = \frac{A}{KDP}$ and is assumed as constant or temperature dependent [9, 20]. A stands for the specific attenuation.

$$\Delta\Phi_{DP} = \Phi_{DP}(r_2) - \Phi_{DP}(r_1) \tag{1}$$

$$f\Delta\Phi_{DP} = 10^{0.1 \cdot b \cdot \alpha \cdot \Delta\Phi_{DP}} - 1 \tag{2}$$

For performance reasons eq. 3 is rearranged as

$$A_H(r) = \frac{[Z_a(r)]^b}{\frac{0.46b \int_{r_1}^{r_2} [Z_a(s)]^b ds}{f(\Delta\Phi_{DP})} + 0.46b \int_{r_1}^{r_2} [Z_a(s)]^b ds} \tag{3}$$

The integral (r_1, r_2) is extracted from the above integral by selection using the first index provided from *wrainfo*.

attenuation_corr.phase_zphi() function. In advance KDP is derived from AH and further PHIDP is recalculated from KDP.

The ZPHI-based AH, KDP, and PHIDP are merged into one xarray dataset which is finally returned.

Figure 7 shows the recalculated polarimetric variables PHIDP and KDP, respectively, and the respective raw dataset as well as the calculated specific attenuation (AH).

Subsequently, the attenuation correction of the reflectivity is performed with the recalculated PHIDP (eq. 4). Figure 8 shows a comparison of the reflectivity before and after attenuation correction. α depends on temperature and drop size distribution. The range of values for X-Band weather radars is between 0.14 – 0.35 dB/deg. α is mostly assumed to be constant ($\alpha = 0.28$) and corresponds to the climatic mean. Certain uncertainties, such as variability in summer and winter, cannot be excluded. Thus, 0.28 is used for α in eq. 5.

$$ZH_{corr} = ZH_{raw} + \alpha \cdot \Phi_{DPrecalc} \tag{4}$$

$$ZH_{corr} = ZH_{raw} + 0.28 \cdot \Phi_{DPrecalc} \tag{5}$$

d) Precipitation estimation

The precipitation module (*precipitation.py*) is used to estimate precipitation rates from clutter and attenuation corrected reflectivity based on the Z-R relationship. This is an empirical relationship between the reflectivity (Z) of the weather radar and the precipitation rate (R). In equation 6 a and b are the relationship parameter and depend on the rainfall type (stratiform or convective). Marshall and Palmer [21] defines default values for a and b are: $a = 200$ and $b = 1.6$.

$$Z = a \cdot R^b \tag{6}$$

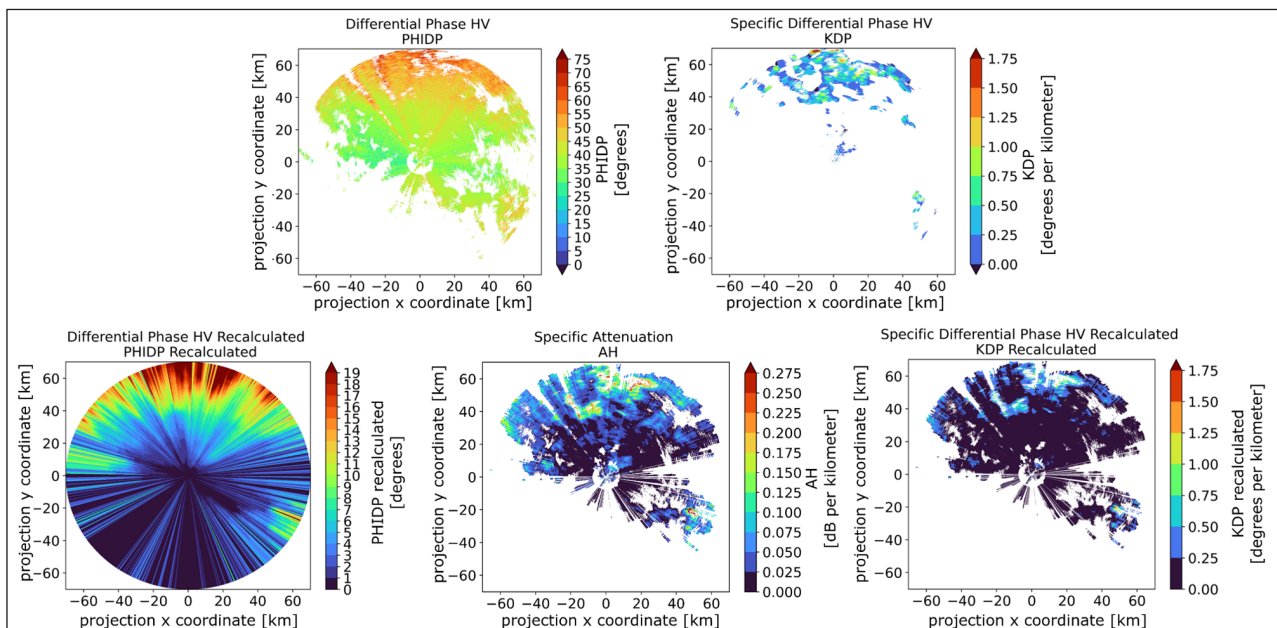


Figure 7 ZPHI based recalculated PHIDP, KDP and specific attenuation.

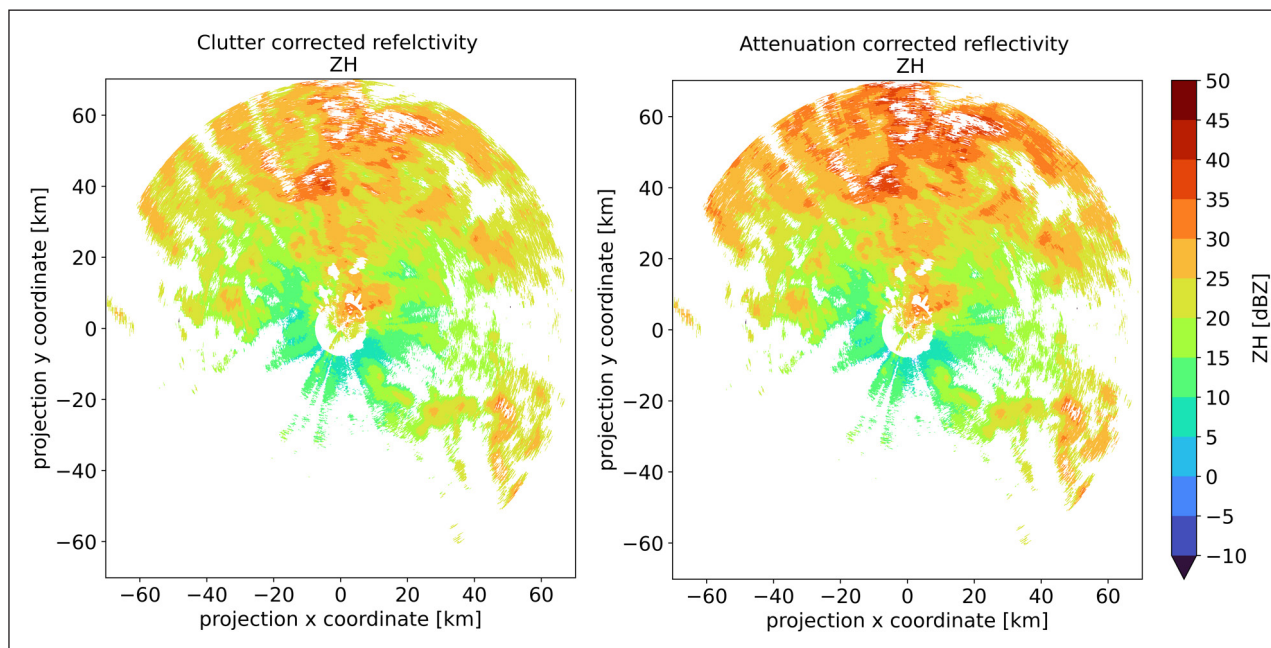


Figure 8 Reflectivity before and after attenuation correction.

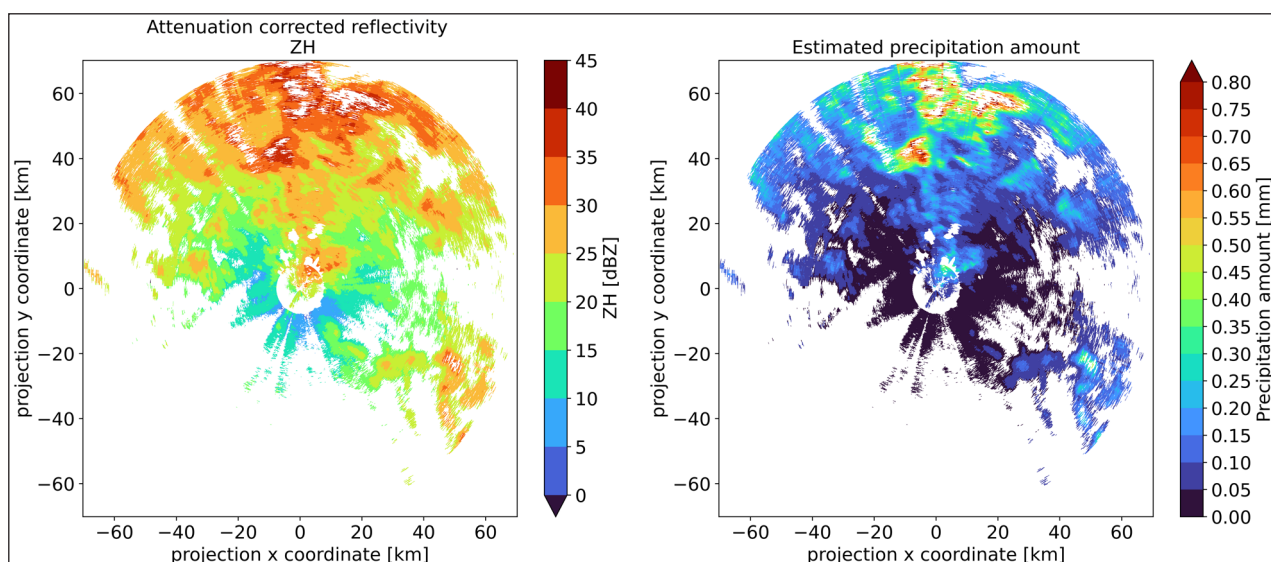


Figure 9 Attenuation corrected reflectivity and estimated precipitation.

The function `wrainfo.precipitation.qpe_zr()` is used for the precipitation estimation. The function needs as input parameter the returned dataset of the attenuation correction and the name of the data array, which contains the attenuation corrected reflectivity. The directory to the configuration file also has to be defined as a function parameter. In the configuration file the scan interval in seconds and the dimensions of the dataset have to be defined.

First, the precipitation intensity [mm/h] is derived from the corrected reflectivity in this function. Second, the amount of precipitation [mm] within the specified scan interval is calculated based on the precipitation intensity [mm/h]. Finally, the returned xarray dataset contains a new variable “PREC_ZR”. [Figure 9](#) shows the estimated

precipitation amount for the clutter and attenuation corrected reflectivity.

e) Georeferencing and gridding

The geometry module is used to project the polar data into a Cartesian grid and to write georeferenced radar data as netCDF files to the hard disk.

The function `wrainfo.geometry.furuno_georeferencing()` uses as input parameter an xarray dataset and the directory to the configuration file, because the target coordinate reference system, the spatial resolution in pixels and the moments, which should be georeferenced have to be defined in the configuration file. The function returns a georeference xarray dataset, which shows in [Figure 10](#).

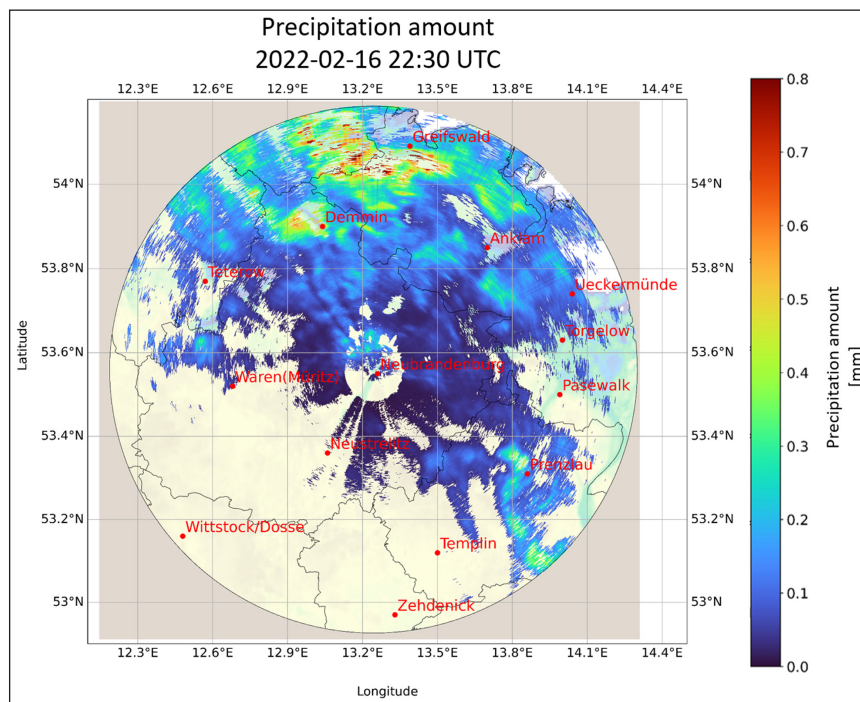


Figure 10 Georeferenced and gridded precipitation data with WRaInfo.

The writing of (georeferenced) radar data as netCDF files to the hard disk is done with the function `wrainfo.geometry.furuno_sweep_to_netcdf()`. This function has the same parameters as the function for georeferencing. Additionally, the data type can be defined. This refers to the level of data processing. The output directory has to be specified in the configuration file.

f) Error and compression module

The error file list (`error_flist.py`) and compression (`compression.py`) module allow the data management of FURUNO data on the hard disk. With the former it is possible to exclude erroneous files from the data processing. With the compression module it is possible to store raw data per day in a compressed tar.gz file.

g) Process chains

WRaInfo contains the module `process_chains.py`, which is based on the modules `reader.py`, `clutter.py`, `attenuation_corr.py`, `precipitation.py` and `geometry.py` (Figure 1).

The module `process_chains.py` shows how the operational mode can be set up to process the raw data of the FURUNO weather radars and subsequently generate georeferenced precipitation products with any temporal resolution. In this paragraph, the individual process chains are described in detail which allow an operational workflow.

GENERATION OF GROUND CLUTTER MAPS

To remove ground clutter from the raw data, monthly clutter maps are generated for each elevation angle over a period of at least 3 months. For this purpose the function `wrainfo.process_chains.clutter_chain()` is used.

The start time and the number of days over which the clutter map should be created must be defined. The stop time is then calculated within the function by adding the start time and the number of days. First, a file list is created for all raw data of the specified elevation angle for the period between start and stop time. Then the processing of the clutter map is done with the function `wrainfo.clutter.create_clutter_map_sequential()` as described in chapter b) clutter detection.

The output filename of the clutter map is created, which contains start and end time of the processed clutter map as well as the elevation angle. The clutter maps are stored as netCDF datasets in the specified output path “`monthly_clutter_directory`” in the configuration file (Table 2).

The function `wrainfo.process_chains.static_cmap()` is used for creating a clutter map from the last three clutter maps for the selected elevation angle. Within the function the last three generated clutter maps are added and saved as netCDF dataset. The created clutter map covers a period of 9 months and is used in the function `wrainfo.process_chains.wr_routine_furuno()`.

If the available data covers only a time series shorter than 9 months, the directory to a clutter map with another arbitrary time period can be specified in the configuration file. This is then used for the function `wrainfo.process_chains.wr_routine_furuno()`.

WEATHER RADAR ROUTINE

For the function `wrainfo.process_chains.wr_routine_furuno()`, which is a process chain to generate precipitation products from the raw data, the following parameters have to be defined in the configuration file. Details can be found in Table 2:

- subfolder_structure_raw_data
- error_flist_directory
- re_index_parameters
- raw_data_directory
- dimensions
- static_cmap
- scan_interval
- epsg_code
- nb_pixels
- moments_in_processed_files
- radar_location_identifier
- output_path_processed_files

Data reading

At first the static clutter map for the selected elevation angle is read. The boolean data array is then extracted from the dataset. This is needed later for the clutter detection. In the next step the filenames of the FURUNO raw data are stored in a list. The list always contains all files for one day, if delta = 1 was set. If delta is chosen differently, the processing is done for a different time period within the for loop.

The data processing of the raw data is accomplished like follows: first, the extension is used to check how to read the data. Second, by reading the data the reindexing of the azimuth angle is done and third, errors in the data array of the elevation angle are corrected.

Application of all corrections

The clutter detection takes place and first the complete dataset is masked with the variable RHOHV. Clutter are removed from the uncorrected reflectivity. Subsequently the phase processing is performed in which PHIDP is recalculated and the specific attenuation is determined and KDP is derived. The recalculated PHIDP is then

used for attenuation correction. After the attenuation correction, the quantitative precipitation estimation can be performed using the Z-R relationship. This is followed by smoothing and georeferencing of the dataset. Finally, the precipitation product is stored in the specified output directory as a netCDF file.

Figure 11 provides an overview of how the individual process chains in the *process_chains.py* module build on each other.

QUALITY CONTROL

In order to ensure functionality and also to be able to release high quality software, WRaINfo contains automated tests for the majority of the functions of each sub-module. For new functions that are integrated into the software, new tests are added respectively. All tests are based on pytest.

When software with newly developed functions is uploaded, an integrated CI pipeline in GitLab is used to attempt to install WRaINfo from a Docker container. If the installation is successful, all automated functional tests are completed. Additionally, the code is checked for PEP8 style using Pylint, and hyperlinks, which are available in the documentation and in README are checked for their response via a dedicated urlchecker test. Only when all tests are successfully completed, the software documentation based on Sphinx is created and deployment to Zenodo and Pypi takes place. The software documentation includes an installation guide and additionally, for each submodule, a set of examples written in Jupyter Notebooks. Using pandoc and the other requirements (nbconvert, nbsphinx, nbformat, jupyter and notebook), the “docstrings” are extracted from the source code and formatted. The Jupyter notebooks are not executed when creating the documentation. However, they can be run

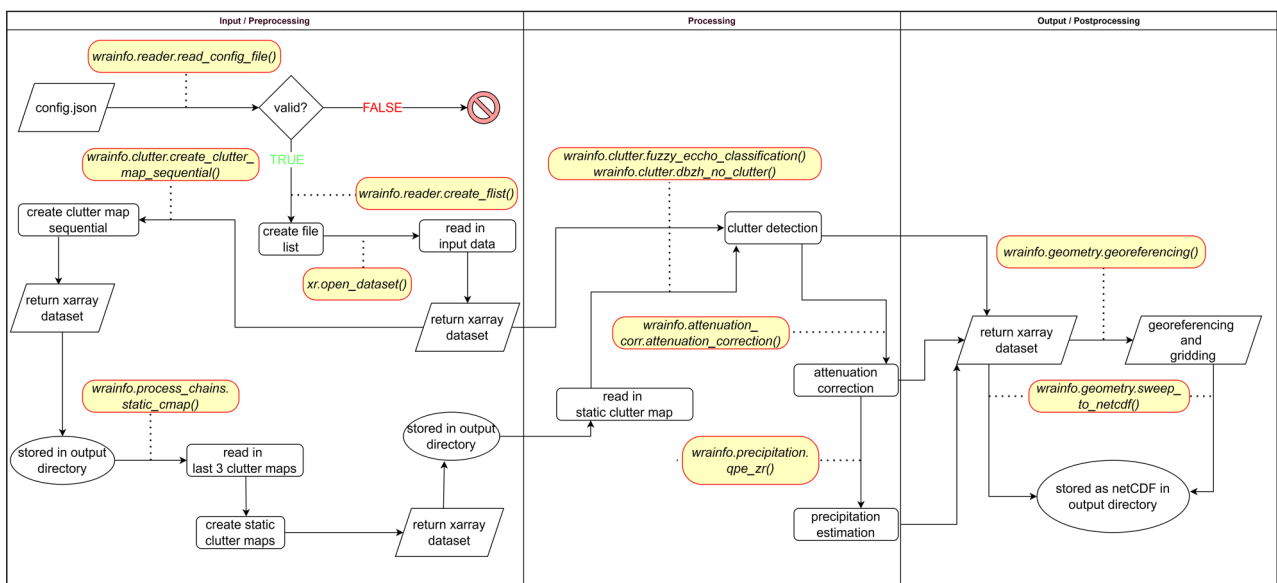


Figure 11 Flowchart of WRaINfo in operation mode.

and traced by users in a python environment themselves. For this purpose, the Git Repository contains a test folder with a few sample files. These files are also used for the automated functional tests.

With each upload, the documentation is automatically updated so that the new code and the documentation remain in sync.

(2) AVAILABILITY

OPERATING SYSTEM

GNU/Linux, Windows

PROGRAMMING LANGUAGE

WRaINfo has been written in Python 3.6 +

DEPENDENCIES

- numpy 1.9+
- rioxarray 0.11.2+
- scipy 1.0+
- wradlib 1.15+
- xarray 0.17.0+

LIST OF CONTRIBUTORS

Alice Künzel, software developer, Helmholtz-Centre Potsdam, German Research Centre for Geoscience (GFZ)

Kai Mühlbauer, software developer, University of Bonn, Institute of Geoscience – Meteorology Section

Julia Neelmeijer, project leader/supervisor, Helmholtz-Centre Potsdam, German Research Centre for Geoscience (GFZ)

Daniel Spengler, project leader/supervisor, Helmholtz-Centre Potsdam, German Research Centre for Geoscience (GFZ)

SOFTWARE LOCATION

Archive

Name: WRaINfo

Persistent identifier: [10.5281/zenodo.7521387](https://zenodo.org/doi/10.5281/zenodo.7521387)

Licence: Apache Software License 2.0

Publisher: FERN.Lab (GFZ Potsdam)

Version published: v0.9.3

Date published: 10/01/2023

CODE REPOSITORY

Name: GitLab

Identifier: <https://git.gfz-potsdam.de/fernlab/products/furuno/wrainfo>

Licence: Apache Software License 2.0

Date published: 04/10/2022

LANGUAGE

English

(3) REUSE POTENTIAL

WRaINfo is a relatively new software package. At the moment WRaINfo is used by the University of Graz and the University of Natural Resources and Applied Life Sciences Vienna. In addition, WRaINfo is already part of a thesis at the University of Graz.

WRaINfo has been written in such a way that it is not only reusable but also extensible. In the future, research institutions operating a FURUNO weather radar will be able to use WRaINfo as a basis and extend the software with their own developments. FURUNO weather radars are already operated at several locations in Germany and also in other countries, so that WRaINfo will gain in importance in the future.

In addition, it is planned to develop a software module for high-resolution precipitation information gathering for farmers, who would like to consider these measurements for management purposes. WRaINfo will be the basis for such an application development and could surely be also used for other application developments based on FURUNO weather radar data (e.g. fire monitoring, storm monitoring, etc.).

ACKNOWLEDGEMENTS

We would like to thank FURUNO Germany for providing us with a dual polarized X-band weather radar and for the close cooperation.

We would also like to thank Neubrandenburg University of Applied Sciences for allowing us to install the weather radar on their building.

FUNDING INFORMATION

1. “Initiative and Networking Fund of the Helmholtz Association”, HIL-A15 FERN.Lab, “Helmholtz Innovation Lab – FERN.Lab
2. German Federal Ministry of Food and Agriculture (on the basis of a decision by the German Bundestag), 28DE114A18, AgriSens DEMMIN 4.0

COMPETING INTERESTS

The authors have no competing interests to declare.

AUTHOR AFFILIATIONS

Alice Künzel  orcid.org/0000-0002-3633-7436

Helmholtz-Centre Potsdam, German Research Centre for Geosciences (GFZ), Potsdam, Germany

Kai Mühlbauer  orcid.org/0000-0001-6599-1034

University of Bonn, Institute of Geoscience – Meteorology Section, Bonn, Germany

Julia Neelmeijer  orcid.org/0000-0002-1165-9404

Supervisor/Project leader, Helmholtz-Centre Potsdam, German Research Centre for Geosciences (GFZ), Potsdam, Germany

Daniel Spengler  orcid.org/0000-0003-2939-8764

Supervisor/Project leader, Helmholtz-Centre Potsdam, German Research Centre for Geosciences (GFZ), Potsdam, Germany

REFERENCES

- Heistermann M, Jacobi S, Pfaff T.** Technical Note: An open source library for processing weather radar data (wradlib). *Hydrol. Earth Syst. Sci.* 2013; 17: 863–871. DOI: <https://doi.org/10.5194/hess-17-863-2013>
- Ryzhkov AV, Zrnich DS.** *Radar Polarimetry for Weather Observations.* Springer Atmospheric Sciences; 2019. DOI: <https://doi.org/10.1007/978-3-030-05093-1>
- Mühlbauer K, Heistermann M, Pfaff T,** et al. *wradlib: An open source library for weather radar data processing;* 2022. DOI: <https://doi.org/10.5281/zenodo.6442964>
- Hitschfeld W, Bordan J.** Errors inherent in the radar measurement of rainfall at attenuating wavelengths. *Journal of the Atmospheric Sciences.* 1953; 11(1): 58–67. DOI: [https://doi.org/10.1175/1520-0469\(1954\)011<0058:EIITRM>2.0.CO;2](https://doi.org/10.1175/1520-0469(1954)011<0058:EIITRM>2.0.CO;2)
- Harrison, Driscoll, Kitchen M.** Improving precipitation estimates from weather radar using quality control and correction techniques. *Meteorological Applications.* 2000; 7(6): 135–144. DOI: <https://doi.org/10.1017/S1350482700001468>
- Jacobi S, Heistermann M.** Benchmarking attenuation correction procedures for six years of single-polarized C-band weather radar observations in South-West Germany. *Geomatics, Natural Hazards and Risk.* 2016; 7(6): 1785–1799. DOI: <https://doi.org/10.1080/19475705.2016.1155080>
- Bringi VN, Chandrasekar V, Balakrishnan N,** et al. An Examination of Propagation Effects in Rainfall on Radar Measurements at Microwave Frequencies. *Journal of Atmospheric and Oceanic Technology.* 1990; 7(6): 829–840. DOI: [https://doi.org/10.1175/1520-0426\(1990\)007<0829:AEOPEI>2.0.CO;2](https://doi.org/10.1175/1520-0426(1990)007<0829:AEOPEI>2.0.CO;2)
- Testud J, Le Bouar E, Obligis E,** et al. 2000 The Rain Profiling Algorithm Applied to Polarimetric Weather Radar. *Journal of Atmospheric and Oceanic Technology.* 17(3): 332–356. DOI: [https://doi.org/10.1175/1520-0426\(2000\)017<0332:TRPAAT>2.0.CO;2](https://doi.org/10.1175/1520-0426(2000)017<0332:TRPAAT>2.0.CO;2)
- Ryzhkov A, Diederich M, Zhang P,** et al. Potential Utilization of Specific Attenuation for Rainfall Estimation, Mitigation of Partial Beam Blockage, and Radar Networking. *Journal of Atmospheric and Oceanic Technology.* 2014; 31(3): 599–619. DOI: <https://doi.org/10.1175/JTECH-D-13-00038.1>
- Grover M, Mühlbauer K, Goudenhoofd E.** *openradar/xradar: xradar v0.3.0 (Version 0.3.0);* 2023. DOI: <https://doi.org/10.5281/zenodo.7091737>
- Helmus JJ, Collis SM.** The Python ARM Radar Toolkit (Py-ART), a Library for Working with Weather Radar Data in the Python Programming Language. *Journal of Open Research Software.* 2016; 4(1): e25. DOI: <https://doi.org/10.5334/jors.119>
- Beamish AL, Anbuhl L, Behlin R,** et al. FERN.Lab: Bridging the gap between remote sensing academic research and society. *Remote Sensing Applications: Society and Environment.* 2021; 24. DOI: <https://doi.org/10.1016/j.rsase.2021.100641>
- Lampridis K.** 2022 *Cookiecutter Python Package.* <https://python-package-generator.readthedocs.io/en/master/>.
- Sphinx.** *Python Documentation Generator;* 2015. <https://www.sphinx-doc.org/en/master/>.
- Pylint Pylint.** 2.16.0-dev documentation. <https://pylint.pycqa.org/en/latest/>.
- FERN.Lab.** *FERN.Lab: Cookiecutter-pypackage;* 2022. <https://github.com/FernLab/cookiecutter-pypackage>.
- Park HS, Ryzhkov A, Zrnich DS** et al. The Hydrometeor Classification Algorithm for the Polarimetric WSR-88D: Description and Application to an MCS. *Weather and Forecasting.* 2009; 24(3): 730–748. DOI: <https://doi.org/10.1175/2008WAF2222205.1>
- Crisologo I, Vulpiani G, Abon CC,** et al. Polarimetric rainfall retrieval from a C-Band weather radar in a tropical environment (The Philippines). *Asia-Pacific J Atmos Sci.* 2014; 50: 595–607. DOI: <https://doi.org/10.1007/s13143-014-0049-y>
- Vulpiani G, Montopoli M, Passeri LD,** et al. On the Use of Dual-Polarized C-Band Radar for Operational Rainfall Retrieval in Mountainous Areas. *Journal of Applied Meteorology and Climatology.* 2012; 51(2): 405–425. DOI: <https://doi.org/10.1175/JAMC-D-10-05024.1>
- Diederich M, Ryzhkov A, Simmer C,** et al. Use of Specific Attenuation for Rainfall Measurement at X-Band Radar Wavelengths, Part I: Radar Calibration and Partial Beam Blockage Estimation. *Journal of Hydrometeorological.* 2015; 16(2): 487–502. DOI: <https://doi.org/10.1175/JHM-D-14-0067.1>
- Marshall JS, Palmer WMK.** The distribution of raindrops with size. *Journal of Atmospheric Sciences.* 1948; 5(4): 165–166. DOI: <https://doi.org/10.1175/JHM-D-14-0067.1>

TO CITE THIS ARTICLE:

Künzel A, Mühlbauer K, Neelmeijer J, Spengler D 2023 WRaINfo: An Open Source Library for Weather Radar INformation for FURUNO Weather Radars Based on Wradlib. *Journal of Open Research Software*, 11: 9. DOI: <https://doi.org/10.5334/jors.453>

Submitted: 16 January 2023 **Accepted:** 25 September 2023 **Published:** 12 October 2023

COPYRIGHT:

© 2023 The Author(s). This is an open-access article distributed under the terms of the Creative Commons Attribution 4.0 International License (CC-BY 4.0), which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited. See <http://creativecommons.org/licenses/by/4.0/>.

Journal of Open Research Software is a peer-reviewed open access journal published by Ubiquity Press.

