Relations between IDL Processing framework and open source radar data processing

Jordi Figueras i Ventura

2016.06.24

Last modified: 2017.02.27

List of IDL functionalities (based and radar volume data) in the malsplus processing framework

|  |  |  |  |
| --- | --- | --- | --- |
| **IDL Processing Framework** | **Py-ART** | **Wradlib (only functionalities not covered by Py-ART)** | **Other** |
| **Supported data formats:** |  |  |  |
| rainbow | Yes | Yes |  |
| Metranet (ELDES) | Own development |  |  |
| MXPol | Yes |  |  |
| **Sanity:** |  |  |  |
| DT filter +despeckle to create a clutter mask | Yes |  |  |
| Melting layer estimation | No |  |  |
| **Preprocessing:** |  |  |  |
| Clutter suppression based on clutter mask | Yes |  |  |
| Visibility correction | Yes |  |  |
| Zh, Zdr Bias correction | Yes |  |  |
| *PhiDP processing:* |  |  |  |
| Offset correction | Yes |  |  |
| Offset correction, SNR censoring and clutter suppression | Yes |  |  |
| Kalman | No |  |  |
| Median | Yes |  |  |
| Median 2 windows | Yes |  |  |
| Multistep (Vulpiani approach) | No | Yes |  |
| Hubbert1995 (Similar to Hubbert & Bringi 1995) | Not as such |  | CSU radar tools |
| *Kdp processing:* |  |  |  |
| Derivation | Not as such |  |  |
| Derivation 2 windows | Not as such |  |  |
| Linear fit | Not as such | Yes |  |
| Linear fit 2 windows | Not as such |  |  |
| Linear fit 3 windows | Not as such |  |  |
| Multistep (Vulpiani approach) | Not as such | Yes |  |
| Least square | Yes |  | CSU radar tools |
| Least square 2 windows | Yes |  |  |
| Kalman | No |  |  |
| *Attenuation correction:* |  |  |  |
| Based on self-consistency | No |  |  |
| Z-Phi | Yes |  |  |
| Linear dependence with PhiDP | Yes | Yes (from Kdp) |  |
| **Hydrometeor classification:** |  |  |  |
| Fuzzy logic based on Dolan et al 2009 | No |  | CSU radar tools |
| Fuzzy logic developed by fvj | No |  |  |
| Unsupervised by Jacopo Grazioli | No |  |  |
| Semi-supervised by Nikola Besic (with ice crystals) | Yes |  |  |
| Semi-supervised by Nikola Besic (without ice crystals) | No |  |  |
| **Precipitation rate estimation:** |  |  |  |
| Simple Z-R relation | Yes | Yes |  |
| Kdp-R relation | Yes | Yes |  |
| Ah-R relation | Yes | No |  |
| variable Z-R according to position respect to melting layer | Not as such | No |  |
| Variable Z-R according to position respect to melting layer  Ah-R, Z-R combination below the melting layer | Not as such | No |  |
| **Vertical Wind shear estimation** | Yes |  |  |
| **Backscatter cross-section from reflectivity values** | Not as such |  |  |
| **Data quality monitoring:** |  |  |  |
| XvsC Position: Search for suitable range bins for radar inter-comparison | Yes |  |  |
| meanZ: Compute the hourly-averaged reflectivity, phidp, and flag matrix of data quality | Yes |  |  |
| XvsC: comparison of the averaged reflectivity of two radars at collocated points | Yes |  |  |
| Intercomp: comparison of the instantaneous values of the polarimetric variables of two radars at collocated points | Yes |  |  |
| Histogram: 1D and 2D histograms of polarimetric variables according to various criteria | Yes |  |  |
| Sun monitoring: radar receiver monitoring using sun hits obtained during normal radar operations | Yes |  |  |
| Monitoring: monitoring of the Zdr in moderate rain, RhoHV in rain, the system PhiDP offset and the reflectivity bias according to the self-consistency algorithm | Yes |  |  |
| Sensor comparison: Point to point comparison of radar data time series with other sensors (disdrometer or rain gauges) | Yes |  |  |
| Quantiles of data | Yes |  |  |
| **Quasi vertical profiles** | Not as such |  |  |
| **Radar data on a plane trajectory** | No |  |  |

List of IDL functionalities (based and radar spectral data) in the malsplus processing framework

|  |  |  |  |
| --- | --- | --- | --- |
| **IDL Processing Framework** | **Py-ART** | **Wradlib** | **Other** |
| **Supported data formats:** |  |  |  |
| In-house DX50 format |  |  |  |
| MXPol |  |  |  |
| **Spectra averaging** |  |  |  |
| **0-Doppler suppression** |  |  |  |
| **Noise estimation** |  |  |  |
| **Noise clipping** |  |  |  |
| **Spectra smoothing with moving window** |  |  |  |
| **Filtering based on sRhohv threshold** |  |  |  |
| **Polarimetric variables computation** (with that all vol radar data functionalities apply) |  |  |  |
| **PSD retrieval from sZh and information on the melting layer** |  |  |  |
| **PSD moments estimation** |  |  |  |

List of IDL functionalities (based on radar IQ data) in the malsplus processing framework

|  |  |  |  |
| --- | --- | --- | --- |
| **IDL Processing Framework** | **Py-ART** | **Wradlib** | **Other** |
| **Supported data formats:** |  |  |  |
| In-house rad4alp format |  |  |  |
| **Time domain windowing** |  |  |  |
| **FFT** (with that all FFT data functionalities apply) |  |  |  |

|  |
| --- |
| **IDL functionalities outside the processing framework** |
| Scattering calculations of different hydrometeors |
| COSMO data conversion into polar radar format |
| Sun measurements from sun scan |

Functionalities covered by open source tools not available in IDL framework

|  |  |
| --- | --- |
| **Functionality** | **Library** |
| **Phidp-kdp processing:** |  |
| Linear programming phase estimation | Py-ART |
| Kdp retrieval by Maesaka | Py-ART |
| **Attenuation correction:** |  |
| Hitschfeld & Bordan method | Wradlib |
| Kraemer method | Wradlib |
| **Doppler de-aliasing:** |  |
| Region-based dealias | Py-ART |
| Washington 4DD code for Doppler de-aliasing | Py-ART |
| **DSD retrieval** | CSU radar tools |
| **Ice and water mass retrieval** | CSU radar tools |
| **Rainfall estimation:** |  |
| Blended algorithm combination of different polarimetric variables | CSU radar tools |
| **Echo classification:** |  |
| Convective-stratiform separation | Py-art |
| Filter Gabella | Wradlib |
| **Vertical Profile of Reflectivity (VPR)** (tools for computing it) | Wradlib |
| **Precipitation adjustment using rain gauges** | Wradlib |
| **Turbulence detection** | PyTDA |
| **Two dimensional wind retrieval** | SingleDop |
| **Estimation of beam blockage from polarimetric data** | PyBlock |