

Microwave Radiometer Data Reprocessing

Ulrich Löhnert
(University of Cologne)

with contributions from Nico Cimini, Harald Czekala, Thomas Rose, Bernhard Pospichal,..

*Ground-based profiling networks for improving weather forecasts
(COST ES1303 Training School)*

EMS Workshop, 3 September 2017, Dublin



What do we need for network operation?

→ homogenized data streams from different instrument types

- from the past until present
- common data formats, file contents and meta data
- common quality control / flagging
- common retrieval applications
- data quicklooks of level1 and level2 products

Goal: be able to
reprocess & on-line process
raw data from MWRs of
different manufacturers to
standardized meteorological
products



Manufacturers' efforts

Currently: Record and view data on-line on instrument PC, data archiving on instrument PC

Missing: Script-based, automated re-processing and historic quicklook generation on external server

Future (RPG): Central data server for monitoring of a multiple instrument network → level1 quality control: calibration, drifts, stability, receiver sanity, spectral consistency based on essential meta-data recording for fully-traceable uncertainty characterization...



Current solution: mwr_pro

- mwr_pro stands for “MicroWave Radiometer PROcessing”
- Based on code developed at UC throughout the last decade
- Processing routines are IDL-based within a “bash-script environment” (linux)
- The final products:
 - are quality controlled
 - stored as netcdf
 - plotted in quicklooks
 - rely on common retrievals
 - are documented concerning genesis (i.e. reproducible)
 - are available through the web (continuously)
- developed specifically for HATPROs at JOYCE
- applied successfully to data from KIT, MeteoSwiss, Leipzig, Cabauw,...



mwr_pro data flow

Need IDL routine tailored to instrument output format

- ✓ RPG
- Radiometrics (planned)

- ✓ Need site-specific multi-variate regression retrievals
- Alternatively: Neural network or 1D-Var (planned)

Read raw data from MWR

Quality control

Apply multi-variate regression retrievals

Store level 1 & 2 data to netcdf (daily basis)

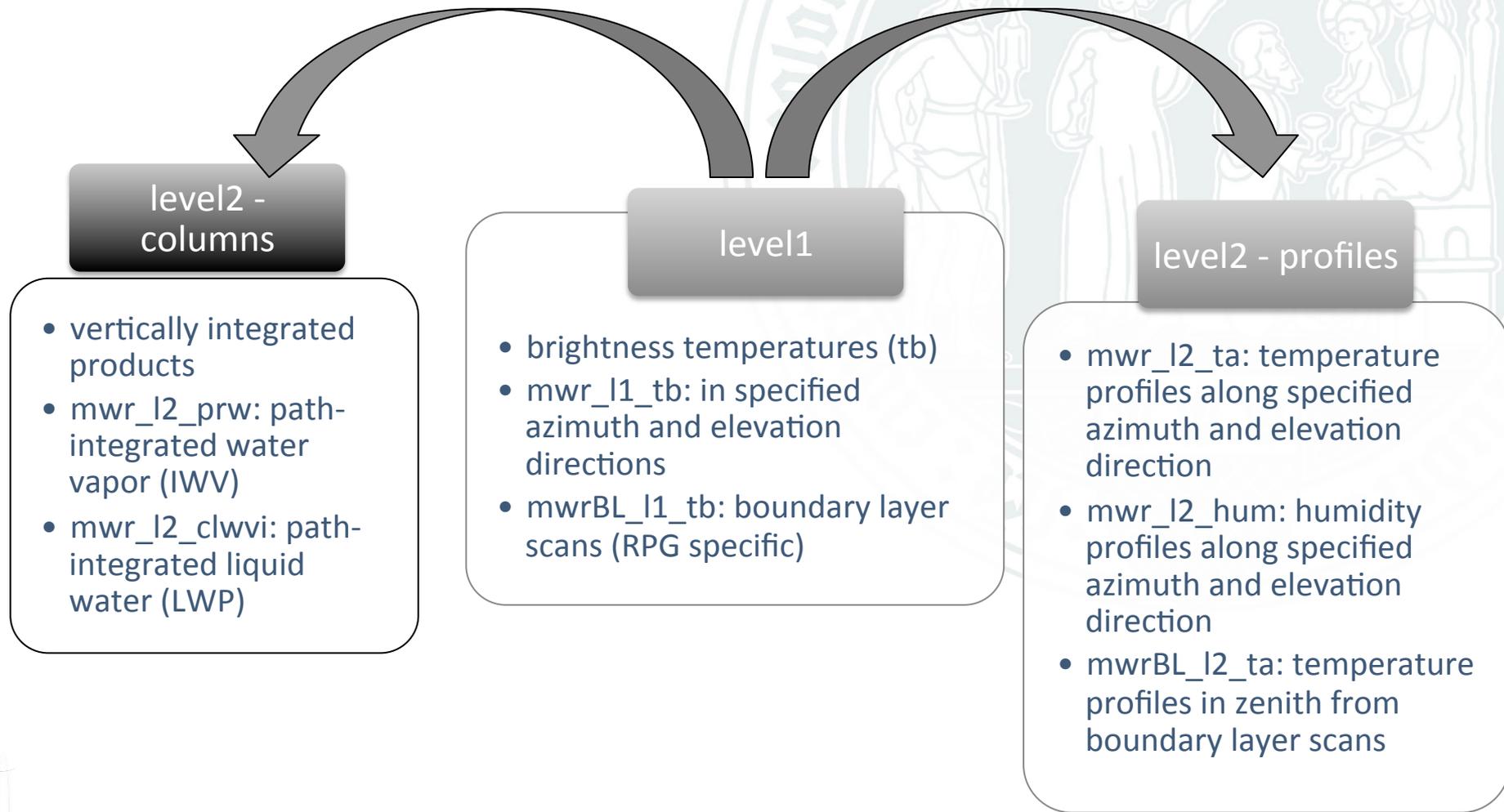
Visualize data on-line (also via web)

- ✓ manual filter (“by eye”)
- ✓ user-defined thresholds (level1 and level2)
- ✓ “housekeeping” data
- ✓ rain flags from precipitation sensor
- spectral consistency checks (planned)

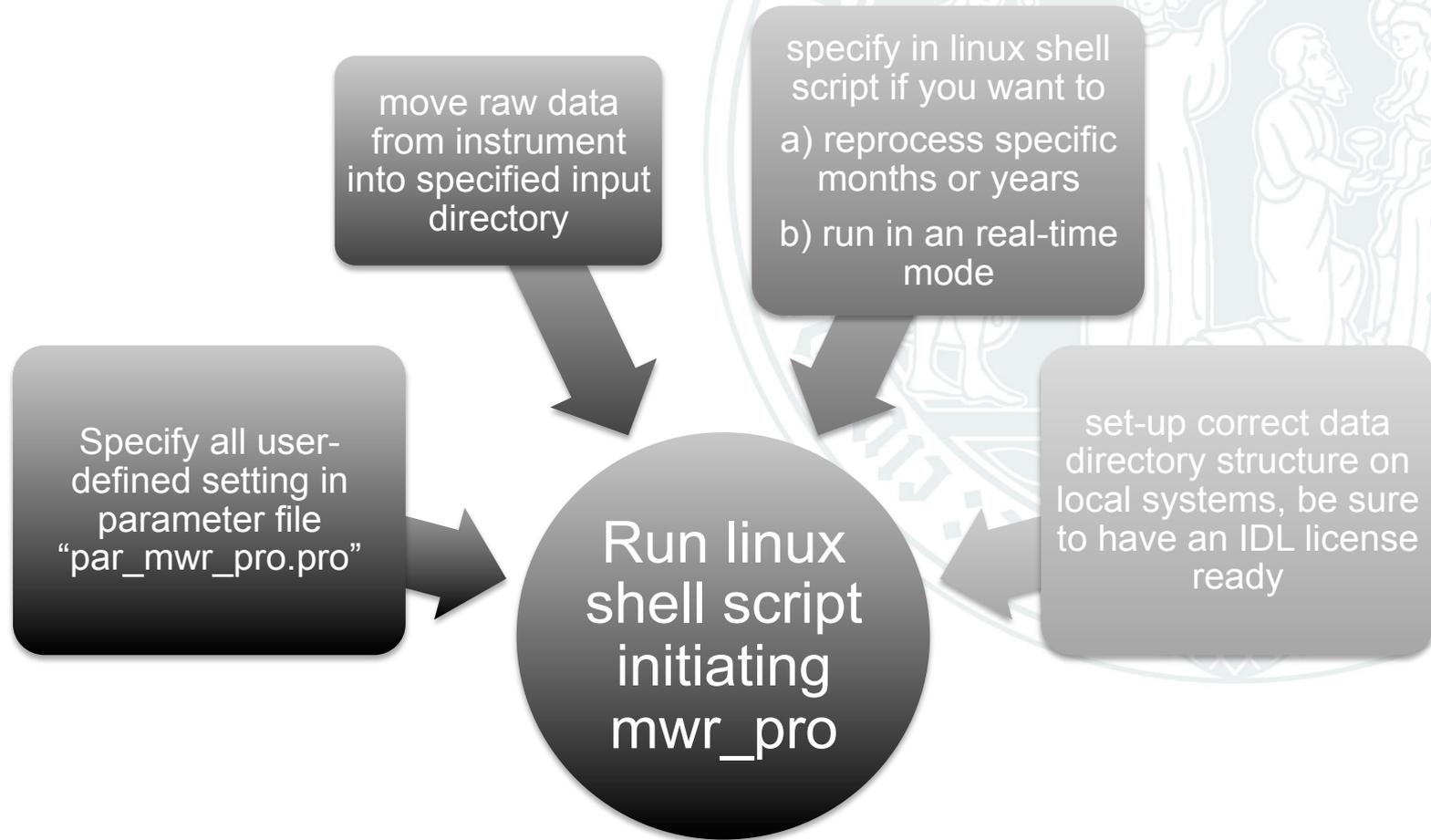
- ✓ According to netcdf CF1.6 convention additionally fulfilling the requirements of [SAMD](#)



mwr_pro data structure



How to basically run mwr_pro?



Reading raw data

RPG - raw data in binary format

- get_spec.pro (TB: *.WVL & *.OLC files)
- get_brt.pro (TB: *.BRT files)
- get_blb.pro (TB boundary layer scan: *.BLB files)
- get_hkd.pro (“housekeeping” data: *.HKD files)
- get_met.pro (met. environment data: *.MET files)
- get_irt.pro (infrared thermometer: *.IRT files)
- get_spec.pro (spectral consistency check; planned)

→ all data combined in get_rpg.pro

RESCOM - raw data in ASCII format

- get_dec.pro (TB: *.dec files)
- all data combined in get_res.pro

RADIOMETRICS:

- routines available, implementation planned

Kipp&Zonen:

- ...

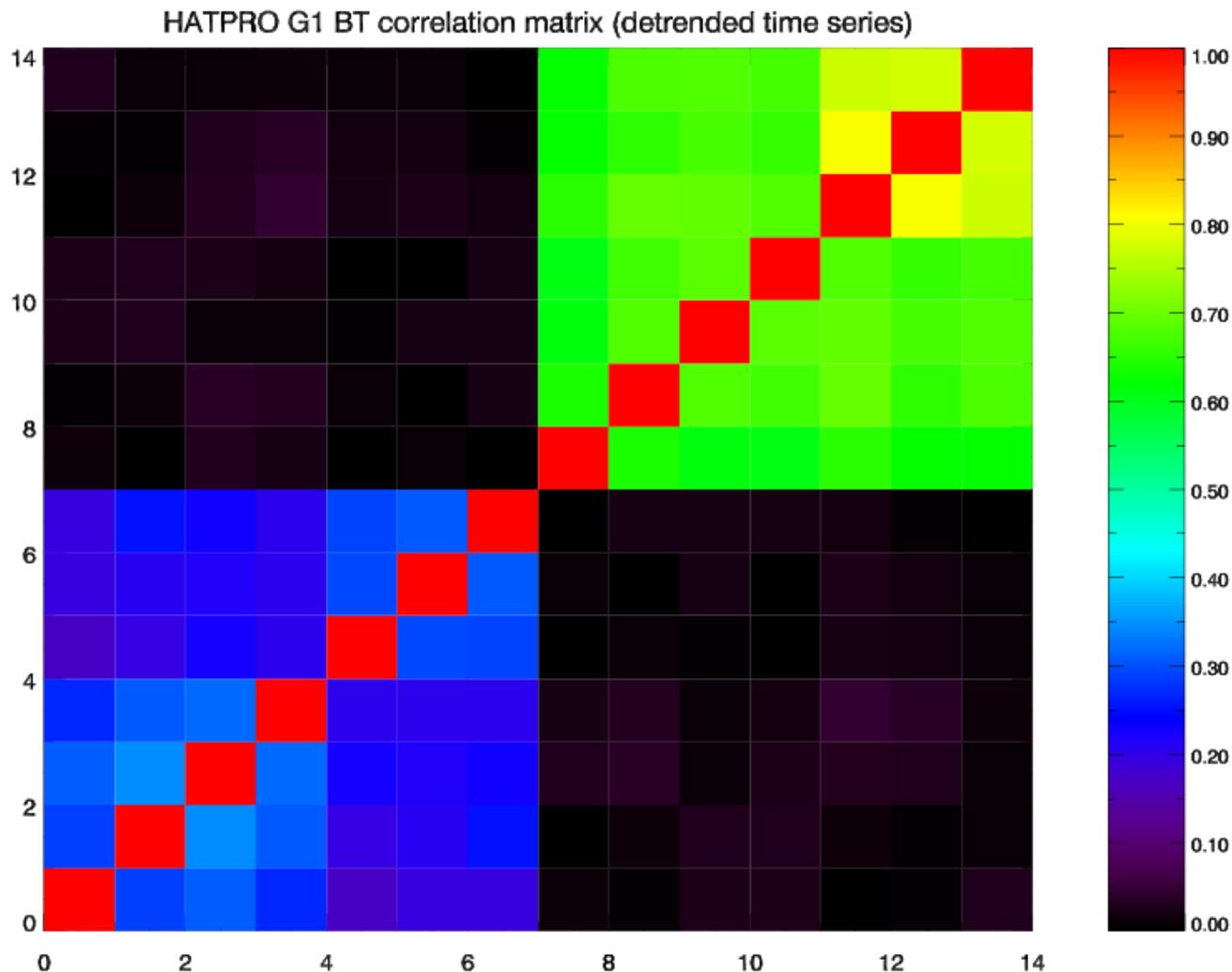


User-defined specifications in `par_mwr_pro.pro`

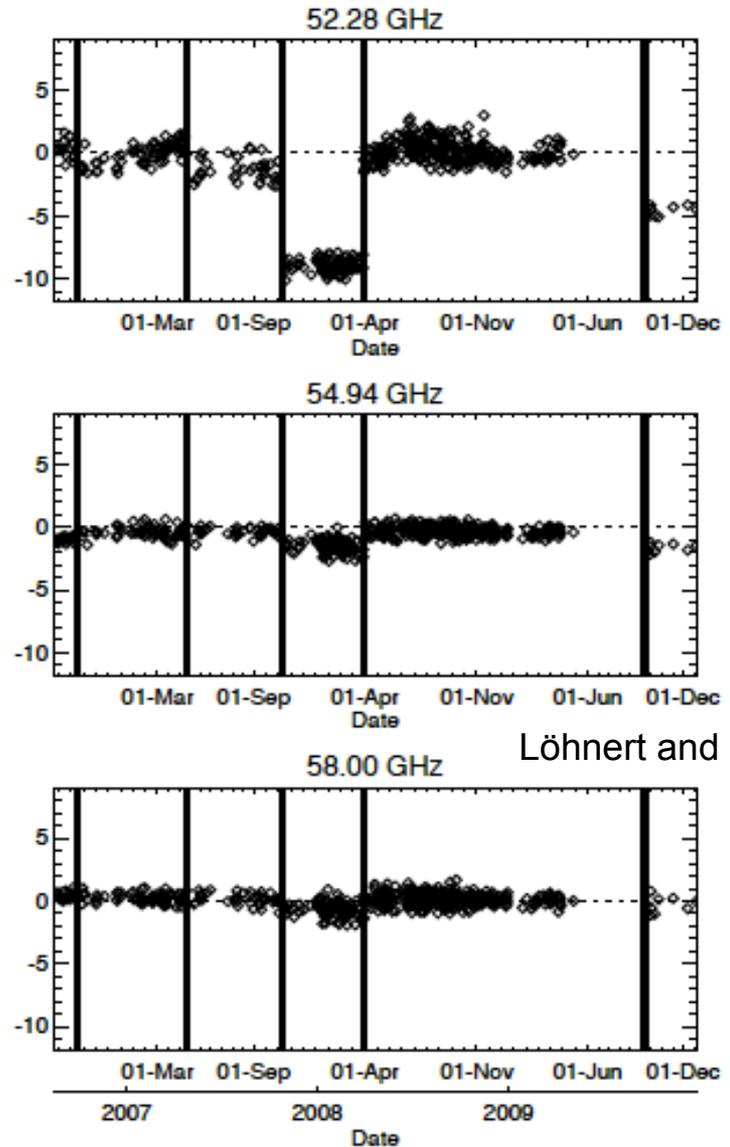
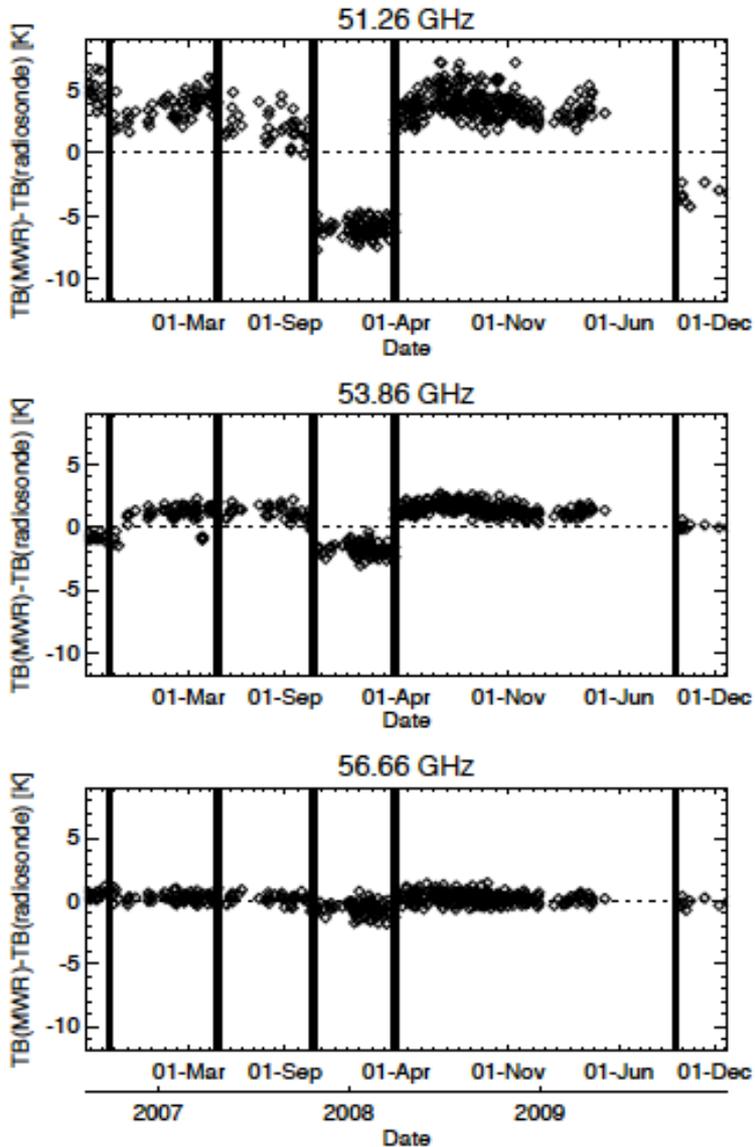
- level1 and level2 thresholds
- TB uncertainty and absolute error specification
- bias correction for TBs using, e.g. NWP model output
- solar angle flagging
- retrievals to be carried out
- LWP offset correction - “zeroing”
- station/instrument description (netcdf header)
- plotting parameters for quicklooks:
 - axis ranges
 - azimuth-time contours
 - combined quicklooks with ceilometer, cloud radar, ..



Correlated uncertainties in MWR TB observations



Often necessary: offset correction



Löhnert and Maier 2012



Quality control flagging

- User-defined thresholds for TB, IWV, LWP and T-profile
- Rain flags supplied by some manufacturers through precipitation sensor
- Housekeeping flags (e.g. RPG)
 - get_hkd.pro returns one flag (rec_sanity = 0 or 1) which tells us if MWR is functioning well internally → rec_sanity=0
 - rec_sanity is analysed separately for K and V band
 - If one of the following internal RPG bits is set, rec_sanity is set to 1:
 - bit1-7: K-Band channel check --> 1: OK, 0: malfunction
 - bit9-15: V-Band channel check --> 1: OK, 0: malfunction
 - bits25&26: --> 2: no thermal stability in K-Band receiver
 - bits27&28: --> 2: no thermal stability in V-Band receiver
 - bit30: --> 1: hot load sensors disagree more than 0.3K



Manual quality control flagging

```
#This file contains manually set quality control flags for MWR data.
#Faulty data times can be manually set and will be flagged in quicklooks and
final netcdf products.
#Possible reasons: disturbances on radome, radio-frequency interference, mis-
calibration, ...
#Note: TBs and products will still be available during the specified times.
#1st column contains date of faulty data following format specification
#2cnd column contains number of faulty intervals on one day
#3rd and 4th column: start time and end time in decimal(!) hours - note, e.g.
19:30=19.50
#5th column: set to 1 if Band1 channels are subject to error
#6th column: set to 1 if Band2 (if existing) channels are subject to error
#7th column: set to 1 if Band3 (if existing) channels are subject to error
#The second to last line must always contain the string "date of last change"
#The last row must contain the actual date of last change.
#You must adhere to formats given in the example below!
#BEGIN OF EXAMPLE
yymmdd nn hh.hh hh.hh 1 2 3
110117 1 19.00 21.00 1 0 0
110118 2 10.00 11.00 1 0 0
          12.50 14.50 1 0 0
date of last change
```



Quality control flagging – how mwr_pro bits are set

```
***QUALITY CONTROL: FLAG extracted TBs and retrieval products
--> levell and level2 flags are generated

flag_1b = REPLICATE(0, N_ELEMENTS(tb.time))
flag_1c = REPLICATE(0, N_ELEMENTS(tb.time))
flag_2a = REPLICATE(0, N_ELEMENTS(tb.time))
flag_2b = REPLICATE(0, N_ELEMENTS(tb.time))
flag_2c = REPLICATE(0, N_ELEMENTS(l0c.time))

;- FLAGS are set as bits
;Bit1: MANUAL FILTER band1 (user edited filter_*.dat file)
;Bit2: MANUAL FILTER band2 (user edited filter_*.dat file)
;Bit3: MANUAL FILTER band3 (user edited filter_*.dat file)
;Bit4: RAIN FLAG (RPG specific)
;Bit5: SANITY RECEIVER band1
;Bit6: SANITY RECEIVER band2
;Bit7: SANITY RECEIVER band3
;Bit8: TB THRESHOLD band1 (set in par_mwr_pro.pro)
;Bit9: TB THRESHOLD band2 (set in par_mwr_pro.pro)
;Bit10: TB THRESHOLD band3 (set in par_mwr_pro.pro)
;Bit11: retrieved LWP/IWV threshold (set in par_mwr_pro.pro)
;Bit12: retrieved TEMPERATURE threshold (set in par_mwr_pro.pro)
```

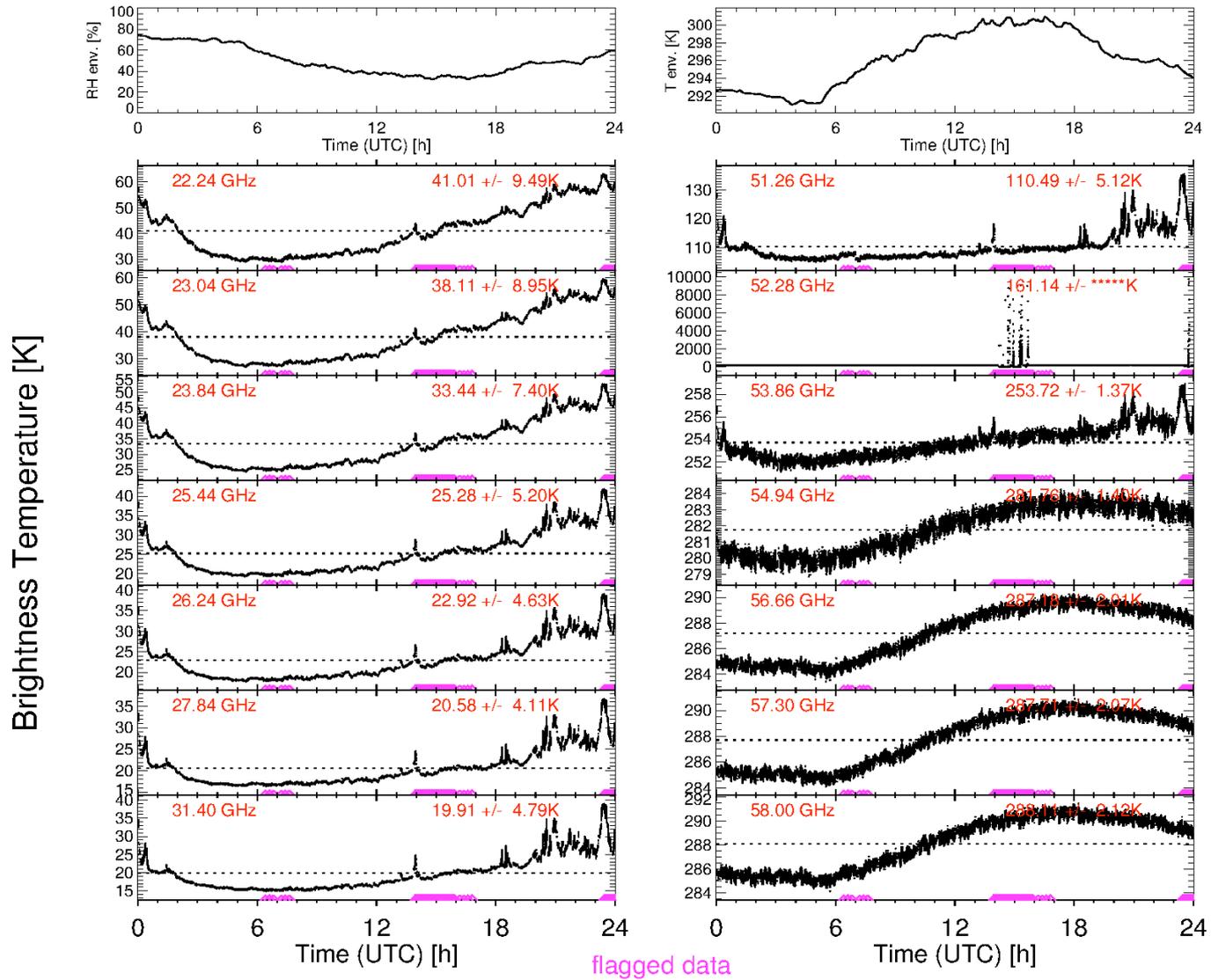


mwr_pro version 4

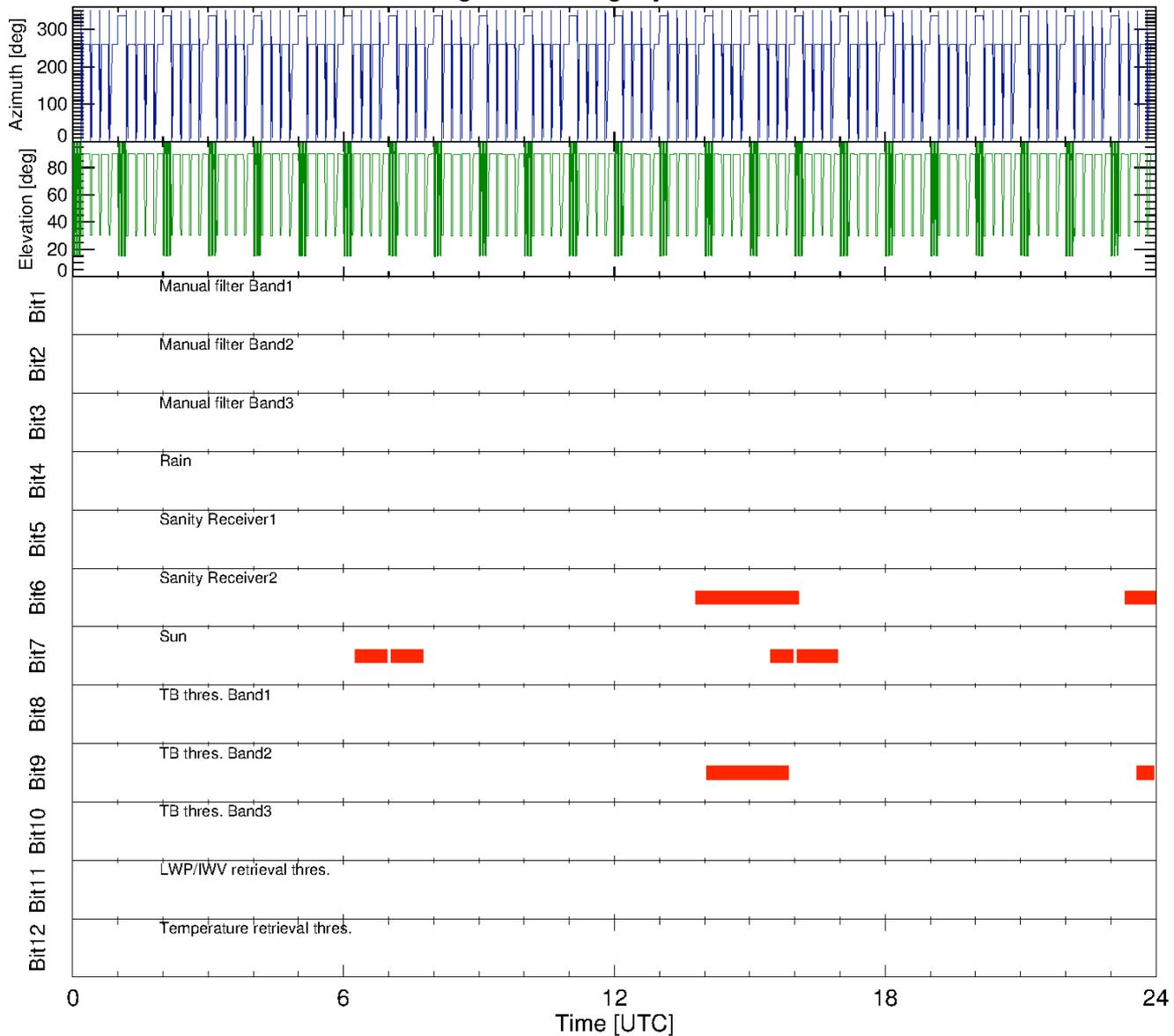
- downloadable at ftp://gop.meteo.uni-koeln.de/pub/loehnert/mwr_data_flow/
- with basic instructions (including example data)
- Also available: example output files generated by mwr_pro
- TOPROF recommendations for network operation and calibration
- Cologne link to quicklook browser: <http://gop.meteo.uni-koeln.de/~hatpro/dataBrowser/dataBrowser4.html>
- Questions? Mail to me: ulrich.loehnert@uni-koeln.de

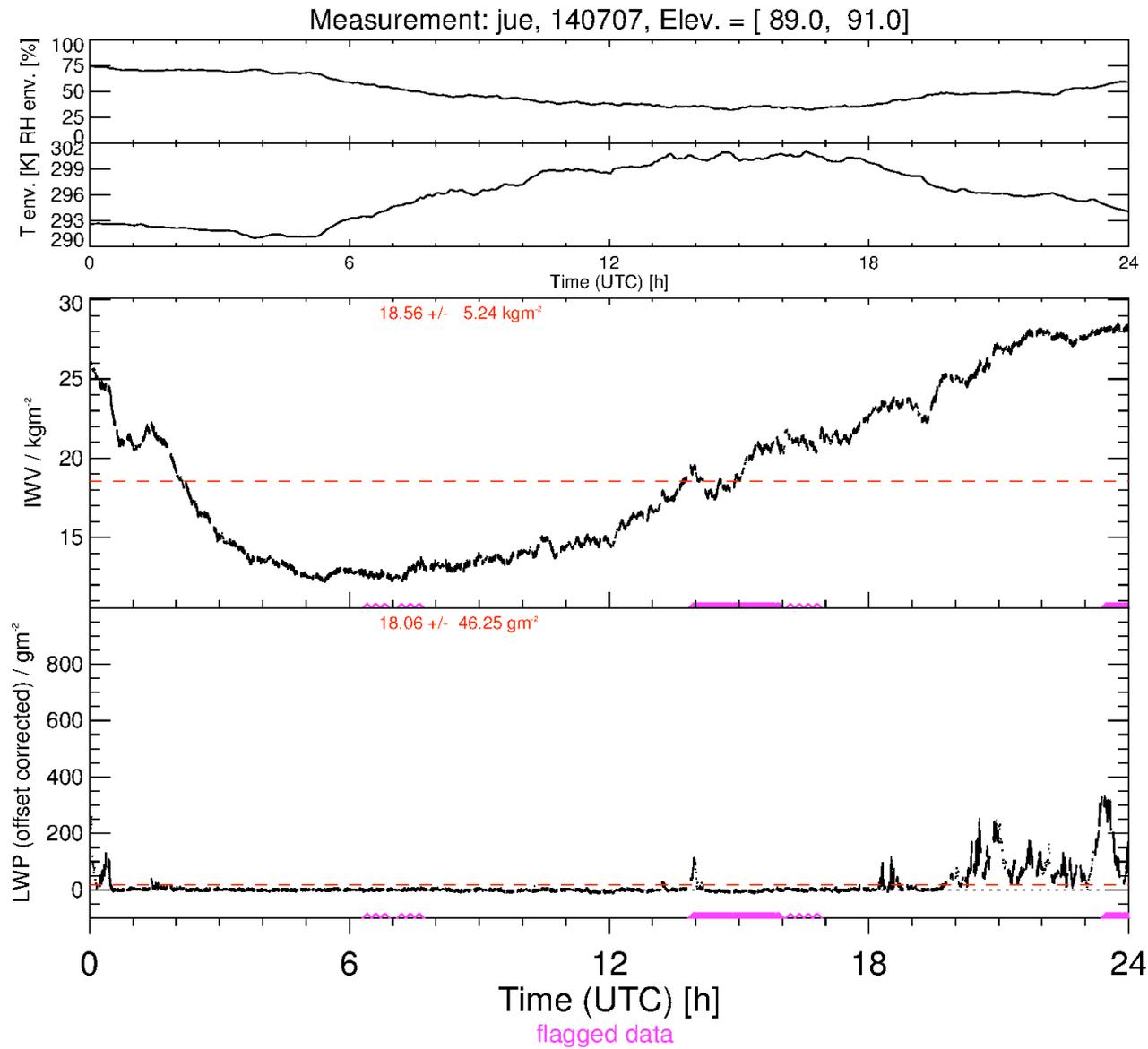


Measurement: jue, 140707, Elev. = [89.0, 91.0]

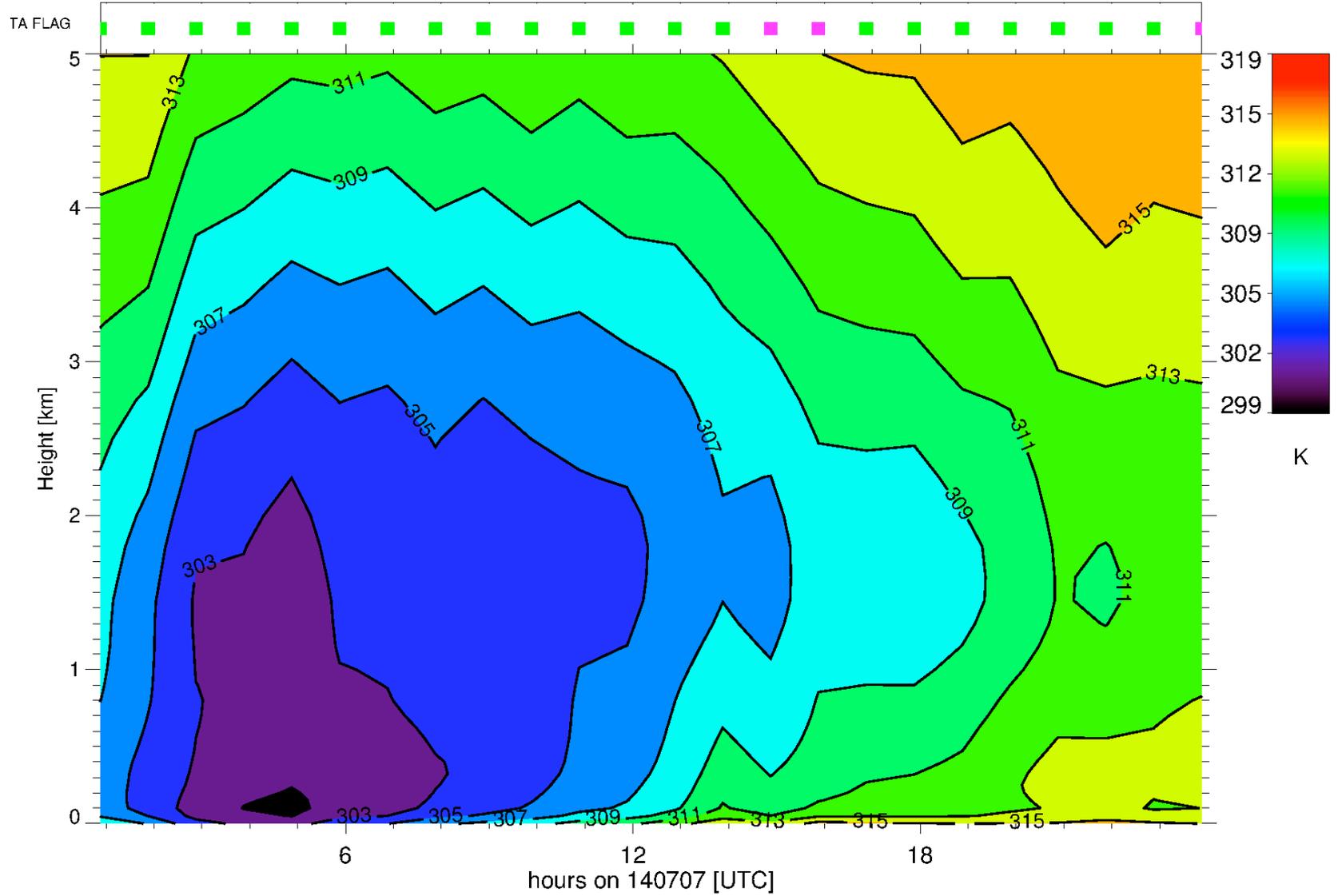


Angles and flags: jue, 140707





Eq. pot. temperature (level2c), jue



Rel. humidity and LWP (level2c), jue

