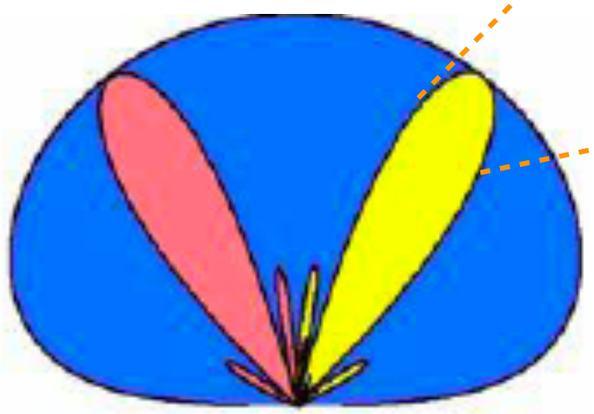
NenuFAR Beamforming time-frequency data

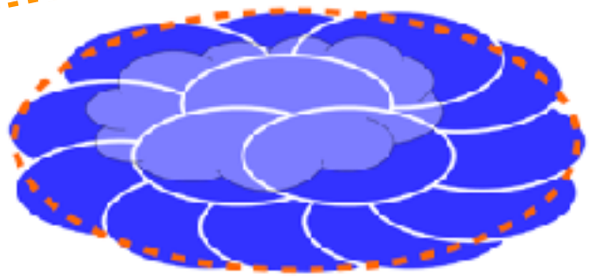
Philippe Zarka

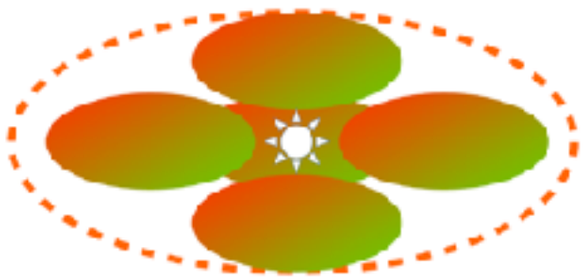
LESIA, Observatoire de Paris, CNRS, PSL, SU, UP
& USN, Observatoire de Paris, CNRS, PSL UO

Analog + Digital beamforming

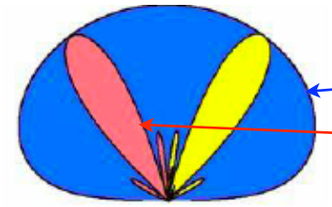
- 768 beamlets (f_c,δf=200 kHz,θ,ϕ)







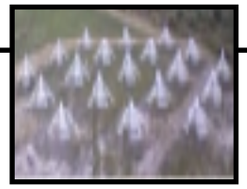
Receivers and signal path



antenna beam

MA beam (analog-phased)

Compact core (400 m) : 96 MA x 2 polar.



Distant MA (≤3 km) : 6 MA x 2 polar.

analog signals

analog signals

analog signals

- Digitization
- Channelization

LOFAR FR606 backend

- Digitization
- Channelization
- Beamforming

LOFAR Super Station

LaNewBa

- Digitization
- Channelization
- Beamforming

NenuFAR Transient Buffer

Broadcast on VLAN

UnDySPuTeD : dynamic spectra

- Channelization
- Integration

UnDySPuTeD : pulsar

- Coherent dedispersion
- Folding / Integration

SETI Machine

NenuFAR Beamformer

NICKEL Correlator

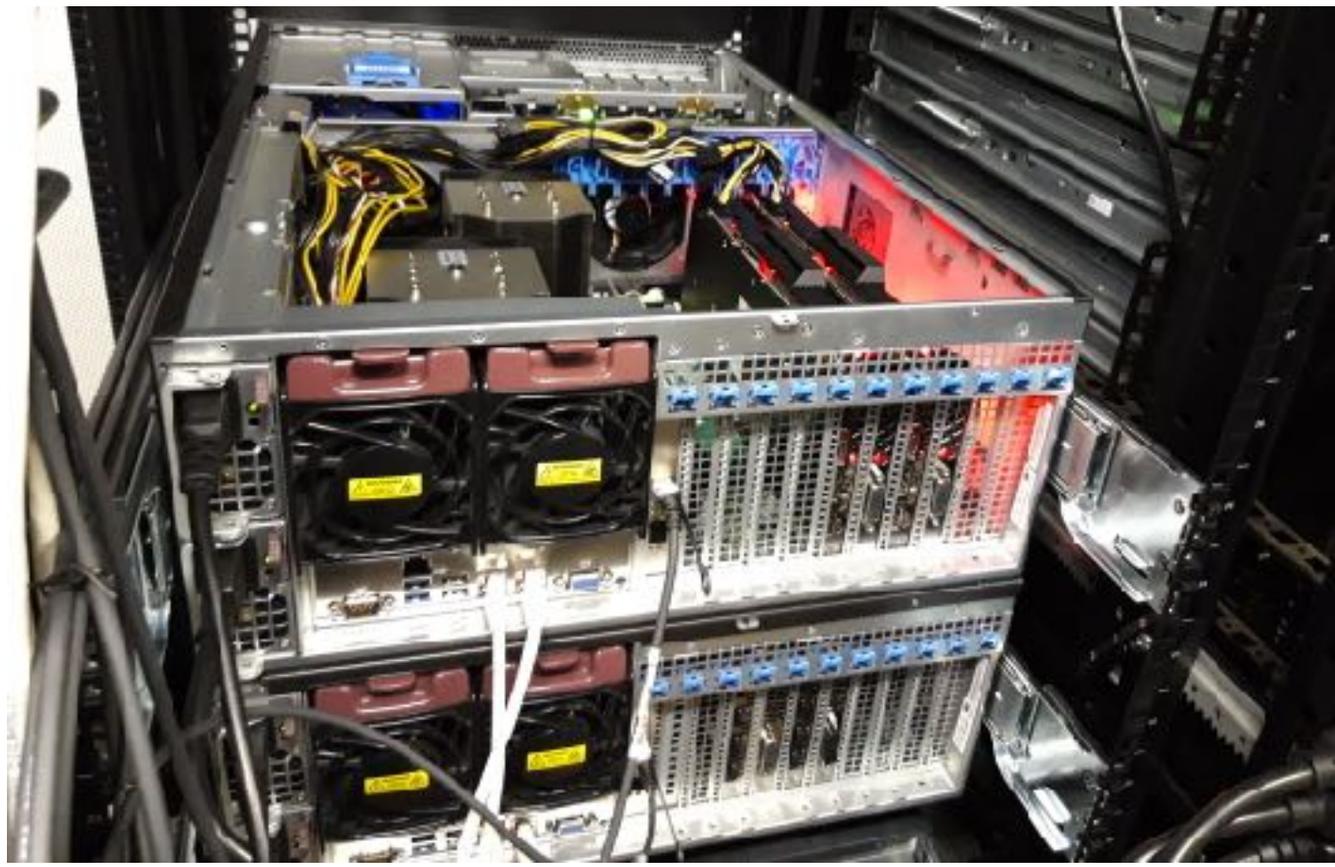
- Channelization
- Correlation
- Calibration + Imaging

NenuFAR Imager

UnDySPuTeD : the Unified Dynamic Spectrum, Pulsar and Time Domain receiver

UnDySPuTeD

- 2x Servers :
  - 2x Intel Xeon E5-2620v4 8cores
  - 32GB DDR4
  - 2x GPU Nvidia GTX 1080/2080/3080



UnDySPuTeD dynamic spectra

- Dynamic Spectra
  - 1 à 2 streams each at 4.8 Gb/s, B=195 kHz, dt = 5.12 µs
  - win-FFT + $<|x_i.x_j|^2>$ to produce 4 Stokes, continuous, real-time
    - B_{min} = 762 Hz, dt_{min} = 1 ms
    - B_{max} = 195 kHz, dt_{max} = 1 s
  - Flux à traiter : 4x 150-300 Mo/s (int8-int16) → 1.2 Go/s (float32)
  - Calcul FFT : ~ 2 GFLOPS

LPC2E

UnDySPuTeD documentation

/cep/lofar/nenufar/pro/undysputed/docs/**Undysputed_Manual+OnlineHelp_IC.pdf**

User Manual for undysputed-tf

Ismaël Cognard, April 8, 2020

The code `tf/undysputed-tf` is used to process Nenufar LaNewBa wave forms data and get spectra with different time and frequency resolutions.

Options, Limitations, Outputs, ‘How to’, Lost packets, Hardware, Code

Outputs description section

In addition to the ASCII .log and .apodisation files, the `'tf'` code (written in C) outputs its results (`.spectra` or `.raw`) as binary files. The types are `ULONG` for unsigned 64bits integer, `INT` for signed 32bits integer, `CHAR` for 8-bits integer while `FLOAT` is for a 32bits float.

Format of `spectra` file

The spectra `*.spectra` file is made of 'block' (made of several spectra) and is following the described format.

Format of `raw` file

A kind of header :

INT nbchan : nber of subsequent beamlets/channels
or nber of NenuFAR/LOFAR 195kHz beamlets
INT nbsample : nber of samples (that is fftlen*nfft)
INT bytespersample : 4/8 for 8/16bits data
... and then for each `nbchan` beamlets :

UnDySPuTeD documentation

Parameters help

Parameters :

Some parameters which are not included or going to be included in the GUI you have to specify specially for pulsars observations and time-frequency acquisition.

Pulsars parameters ...

WaveForm parameters :

Parameters line:

```
tf: rawrt
```

Dynamic Spectrum parameters :

Parameters line (more details in 'user guide' of Documentation):

```
tf: df=xx dt=xx
```

with the following choices:

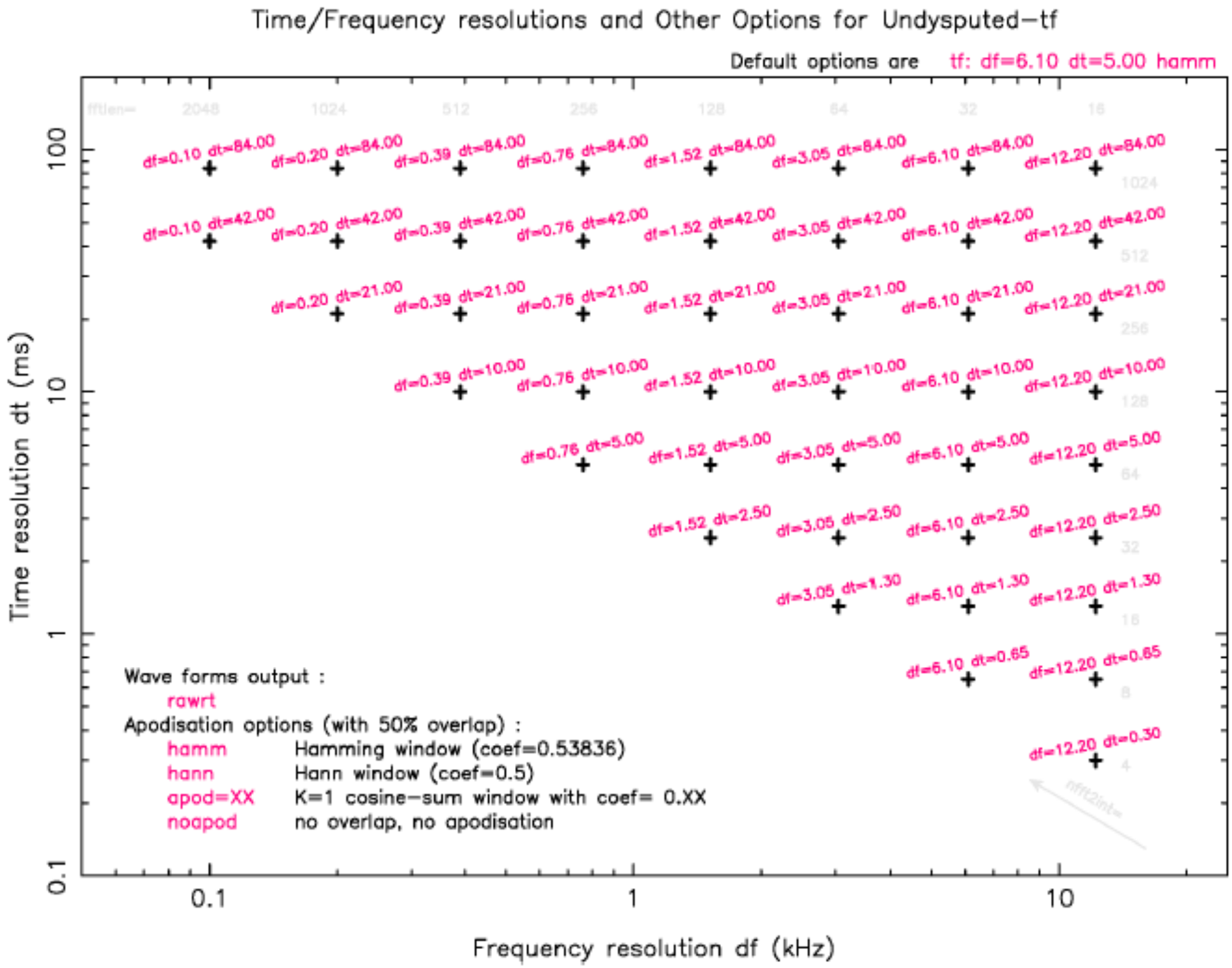
- if df=0.10 :
  dt= 42.0 or 84.0
- if df=0.20 :
  dt=21.0 or 42.0 or 84.0
- if df=0.39 :
  dt=10.0 or 21.0 or 42.0 or 84.0
- if df=0.76 :
  dt=5.0 or 10.0 or 21.0 or 42.0 or dt=84.0
- if df=1.52 :
  dt=2.5 or dt=5.0 or 10.0 or 21.0 or 42.0 or dt=84.0
- if df=3.05 :
  dt=1.30 or dt=2.5 or dt=5.0 or 10.0 or 21.0 or 42.0 or dt=84.0
- if df=6.10 :
  dt=0.65 or dt=1.30 or dt=2.5 or dt=5.0 or 10.0 or 21.0 or 42.0 or dt=84.0
- if df=12.20 :
  dt=0.30 or dt=0.65 or dt=1.30 or dt=2.5 or dt=5.0 or 10.0 or 21.0 or 42.0 or dt=84.0

Example:

Present parameters for Exoplanets:
• Day: Freq 30.1-67.4 MHz
• Night: Freq 21.1-58.4 MHz
tf: df=3.05 dt=21.0 hamm

Present parameters for Stars:
• Freq 12-86.9 MHz
tf: df=3.05 dt=42.0 hamm

UnDySPuTeD-tf parameters



→ min(b.τ) = 4

UnDySPuTeD-tf parameters

- Interactive parameter determination using /cep/lofar/nenufar/pro/undysputed/**resolutions_nu_tf.pro**

```
IDL command Examples:
    resolutions_nu_tf,/hlp

RESOLUTIONS_NU_TF, dt=dt (msec), df=df (kHz), /table, /help

resolutions_nu_tf,/help

Undysputed-tf resolutions, from I. Cognard short user manual lists exact and VCR resolutions

CALL
RESOLUTIONS_NU_TF, dt=dt, df=df, /table

INPUTS
dt [msec], df [kHz]

KEYWORDS
/table to display the tables with all possibilities
/help     => extensive help
/hlp      => summary help

OUTPUTS
dt=..., df=... : checks compatibility of given (dt, df)
dt=... : provides all possible df for a given dt (range of possible dt = 0.3 - 84 msec)
df=... : provides all possible dt for a given df (range of possible df = 0.1 - 12 kHz)
```

UnDySPuTeD-tf parameters

`resolutions_nu_tf,dt=10`

```
================================================================================
dt (msec)  10.486
df (kHz)   12.207   6.104   3.052   1.526   0.763   0.381
--------------------------------------------------------------------------------
VCR: dt(msec) = 10.00 => df(kHz) = 12.20  6.10  3.05  1.52  0.76  0.39
================================================================================
```

`resolutions_nu_tf,df=5`

```
================================================================================
df (kHz)    6.104
dt (msec)    0.65    1.31    2.62    5.24   10.48   20.97   41.94   83.88
--------------------------------------------------------------------------------
VCR: df(kHz) = 6.10 => dt(msec) = 0.65  1.30  2.50  5.00  10.00  21.00  42.00  84.00
================================================================================
```

`resolutions_nu_tf,dt=10,df=5`

```
================================================================================
  6.104 kHz x 10.486 msec are compatible  =>  VCR:    tf: df=6.10 dt=10.00 hamm
================================================================================
```

`resolutions_nu_tf,dt=10,df=1`

```
================================================================================
  0.763 kHz x 10.486 msec are compatible  =>  VCR:    tf: df=0.76 dt=10.00 hamm
================================================================================
```

`resolutions_nu_tf,dt=10,df=0.1`

```
df and dt not compatible :    0.0953674 kHz   x      10.4858 msec
```

`resolutions_nu_tf,/table`

Data pre-processing

- Reduction code **read_nu_spec.pro** (IDL) + python analog *(cf. A. Loh)*
- Last version (2.4 of 2021/10/07) on /cep/lofar/nenufar/pro/undysputed/ **read_nu_spec.pro**
- Up-to-date documentation (pdf) on /cep/lofar/nenufar/pro/undysputed/docs/ **read_nu_spec.pdf**
- Fully operational, tested, parameterized, flexible (keywords), stable
- Processes one Undysputed-tf file n.spectra at a time
- Performs on-the-fly
  - data selection /t,f,beam
  - RFI mitigation /f, sb, t + sb & channel flagging
  - correction or whitening of bandpass & gain(t) [reading altazA files]
  - intra-channel de-dispersion
  - integration / t,f
  - computation of Stokes parameters (uncalibrated)
  - cosmetic post-processing of output data
  - L1 fits + L0 log txt writing

read_nu_spec code

/cep/lofar/nenufar/pro/undysputed/docs/**read_nu_spec.pdf**

Pre-processing NenuFAR-UnDySPuTeD time-frequency data

P. Zarka, v2.4, 26/10/2021.

The present document details how to read / pre-process / reduce NenuFAR-UnDySPuTeD time-frequency data.

1. UnDySPuTeD and LANewBa data

UnDySPuTeD[1] (hereafter Undysputed) is the receiver of the NenuFAR beamformer mode. It is fed by the main backend LANewBa[2], that is itself fed by the 96 Mini-Arrays (MA) x 2 polarizations (NE and NW) analog signals from the core.

2. UnDySPuTeD data files structure

...

3. Data reduction principles

...

read_nu_spec code

4. Data reduction software

The main command to run read_nu_spec.pro within IDL is :

```
READ_NU_SPEC, file, data,time,freq,beam,ndata, nt,dt,nf,df,ns, jd0,h0, corrf,corrt,fref,
      tmin=tmin,tmax=tmax, julian=julian, fmin=fmin,fmax=fmax, exactfreq=exactfreq,
      beams=beams, nchannels=nchannels, ntimes=ntimes, nstokes=nstokes,
      nobandpass=nobandpass, ex_chan=ex_chan, ex_beamlets=ex_beamlets,
      fclean=fclean, bclean=bclean, tclean=tclean, fflat=fflat, tflat=tflat, dm=dm,
      fcompress=fcompress, tcompress=tcompress, fill=fill,
      round_times=round_times, round_freq=round_freq, block_inc=block_inc,
      writefits=writefits, writetxt=writetxt, info=info, help=help
```

We describe below parameters and keywords, and then provide call examples. The description is an expanded version of the program header.

5. Usage

5.1 Use cases

ES02 : Exoplanets and Stars

→ *RFI mitigation, temporal integration from 21 msec to 250 msec, spectral integration to 4 channels per subband (~48 kHz), correction of gain jumps, Stokes IVL or IQUV*

```
READ_NU_SPEC, path+file, data,time,freq,beam,ndata, nt,dt,nf,df,ns, jd0,h0, corrf,corrt,
      nchannels=4, ntimes=12, nstokes=3 [or 4], ex_chan=[0], ex_beamlets=[list…],
      fclean=[4.,101], /bclean, /tclean, tflat=[7,36.5,70.], /fill,
      writefits=2, writetxt=2, block_inc=1000
```

list of excluded beamlets is determined from the integrated spectrum of raw data.

Beamformed-TF pipeline

- Suite of IDL procedures based on **read_nu_spec.pro**
  - proc_esXX.pro
  - batch_proc_esXX.pro
  - check_esXX_fits.pro

- RFI mitigation, correction of gain jumps , Stokes IVL or IQUV, temporal and spectral integrations

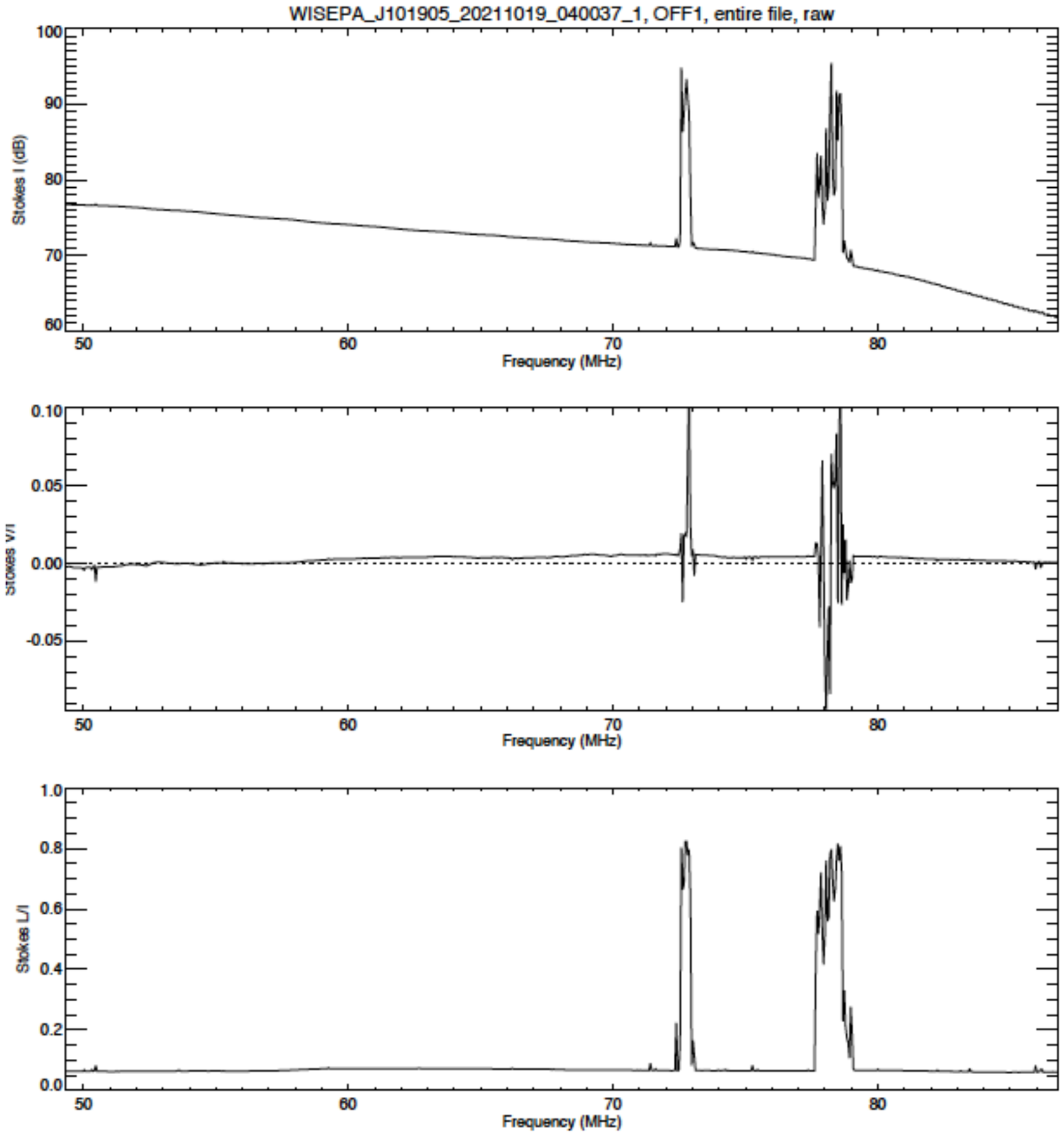
- Produces L1 **fits** data, **txt** raw header, **pdf quick-looks** on L0 data directory

- Runs at ~2±1 x observing time speed per thread

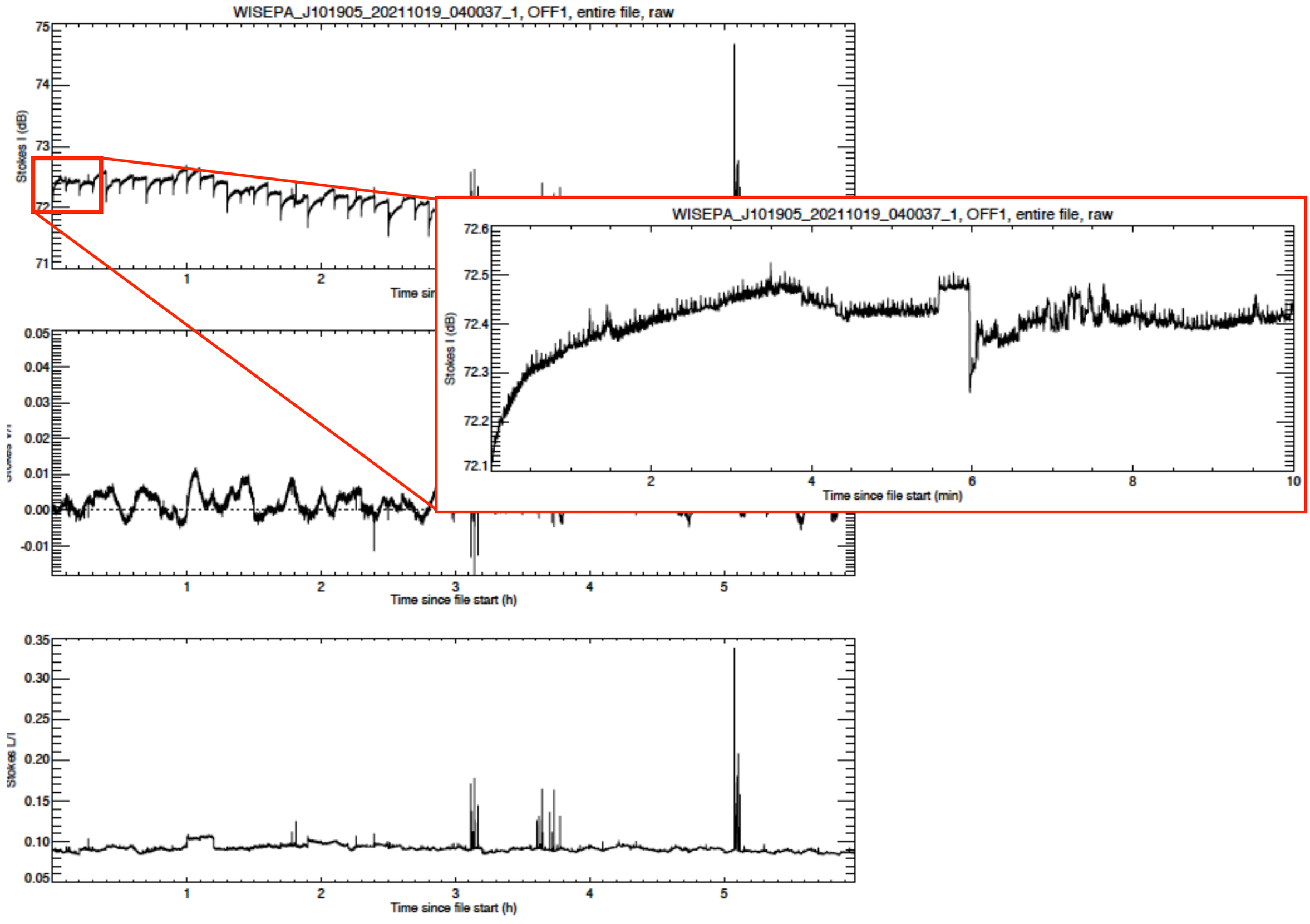
- Used in routine by ES02 (Exoplanet/Stars), ES05 (FRB), ES06 (Planetary Lightning), ES10 (RRLs) pipelines ; main difference = read_nu_spec parameters

- Can be adpated to ES07, ES11 ⟶ quick-looks + event selection

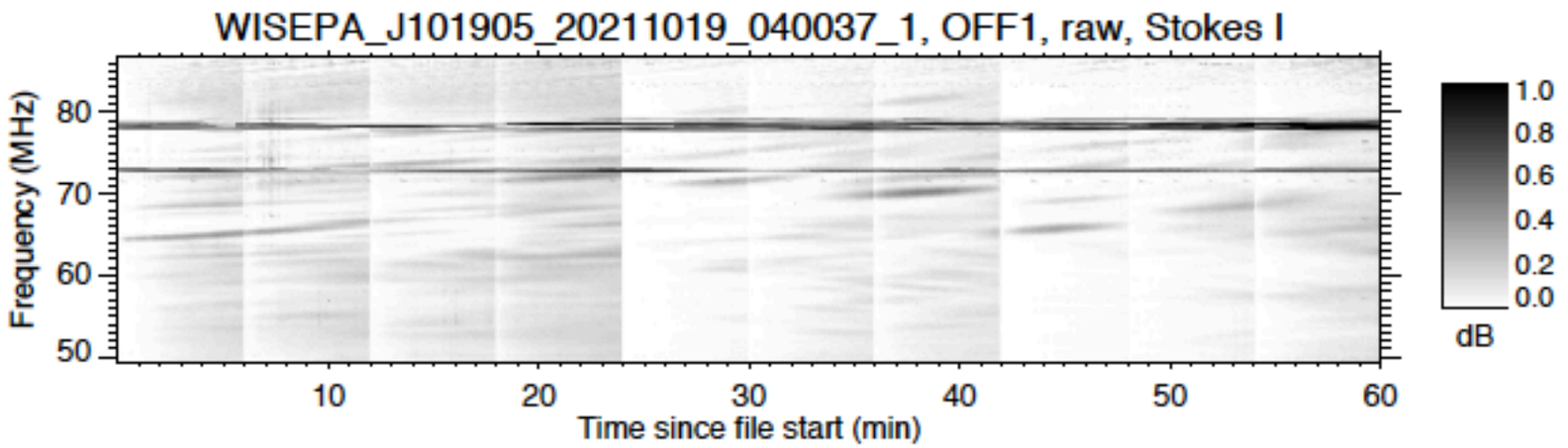
Quick-look example (WISEPA_J101905 observation)

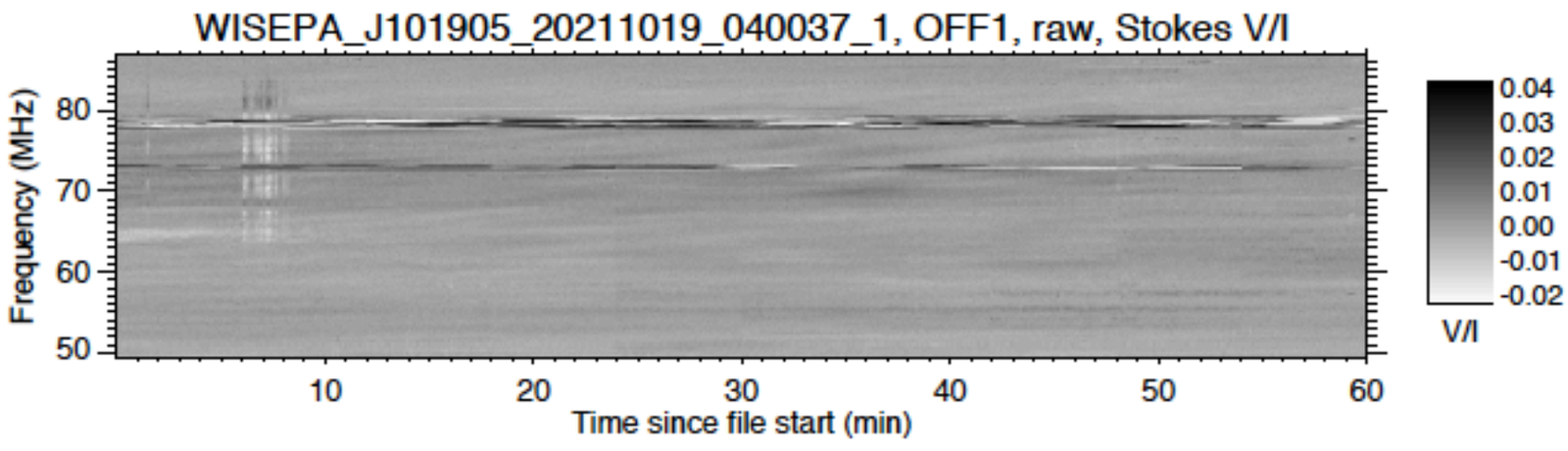


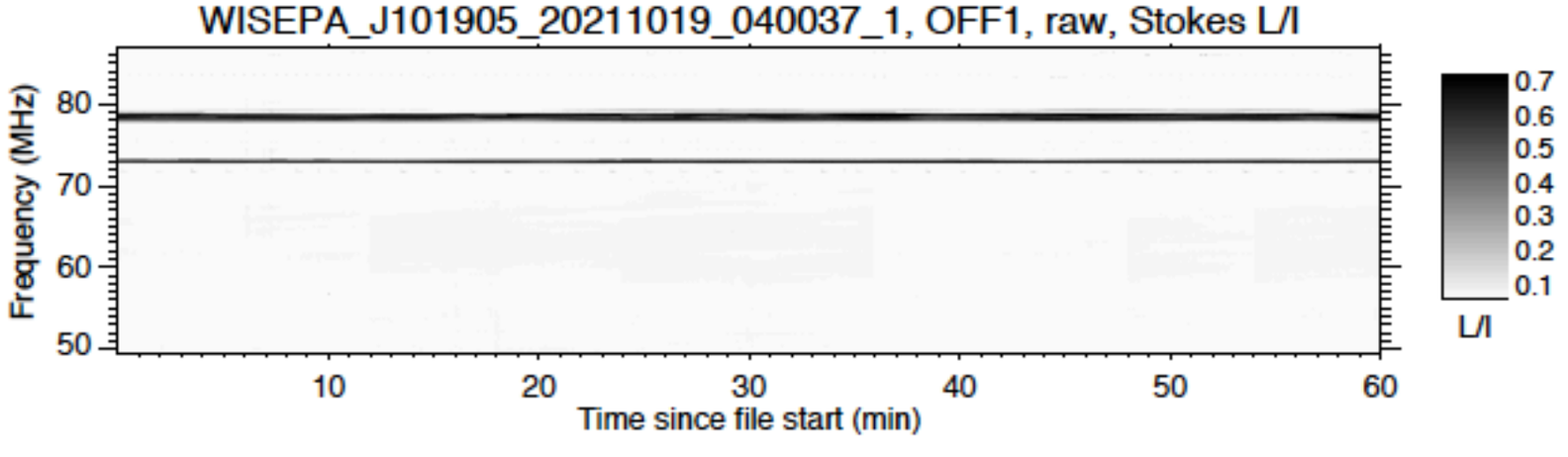
Quick-look example (WISEPA_J101905 observation)



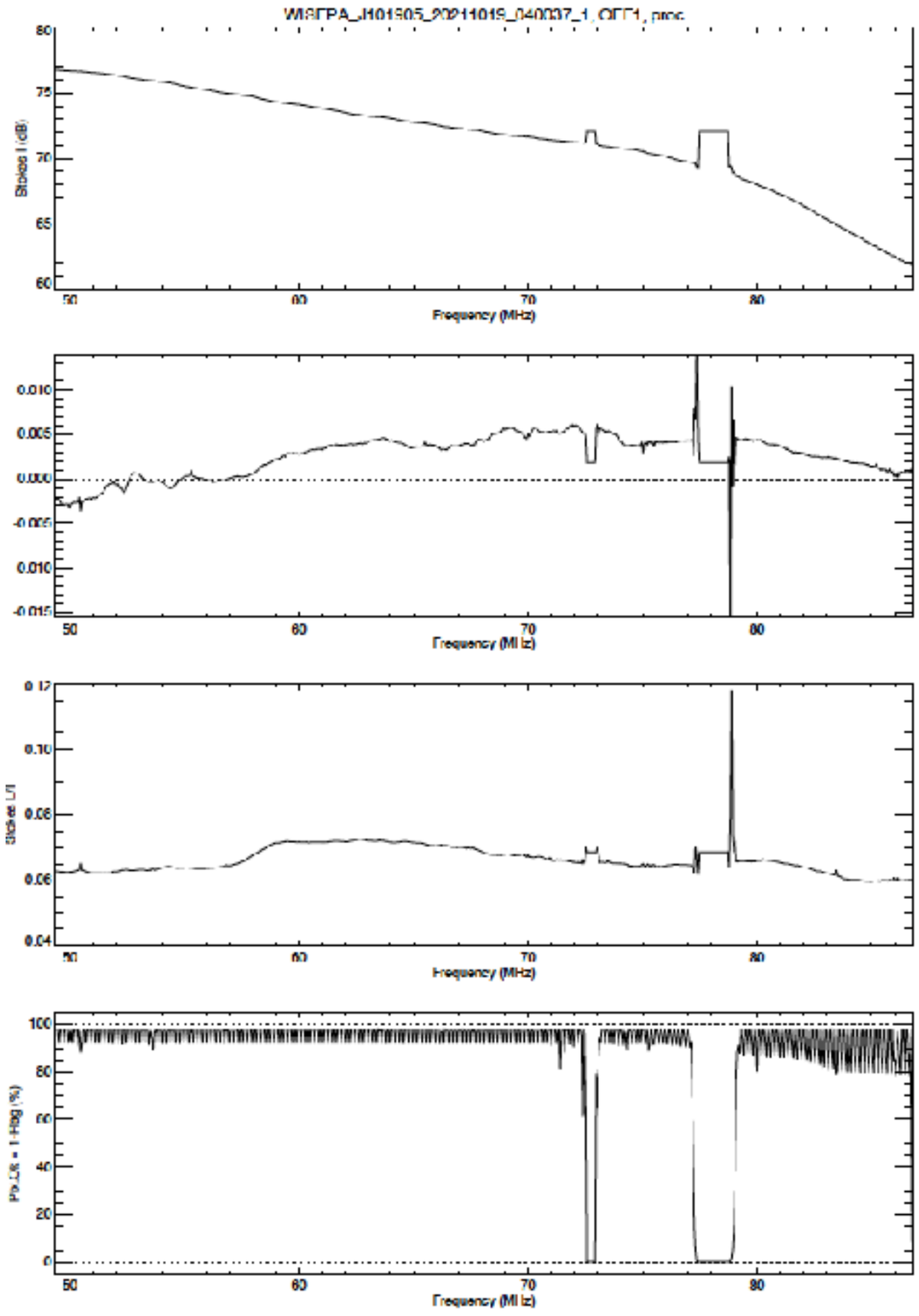
Quick-look example (WISEPA_J101905 observation)



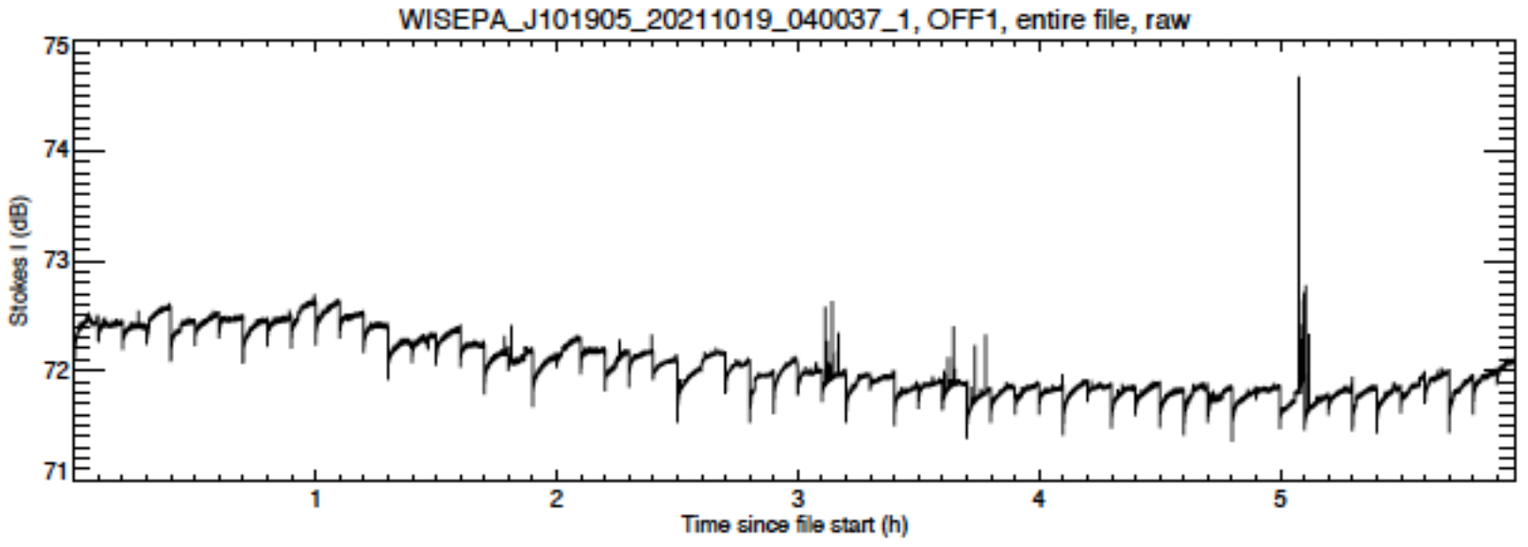




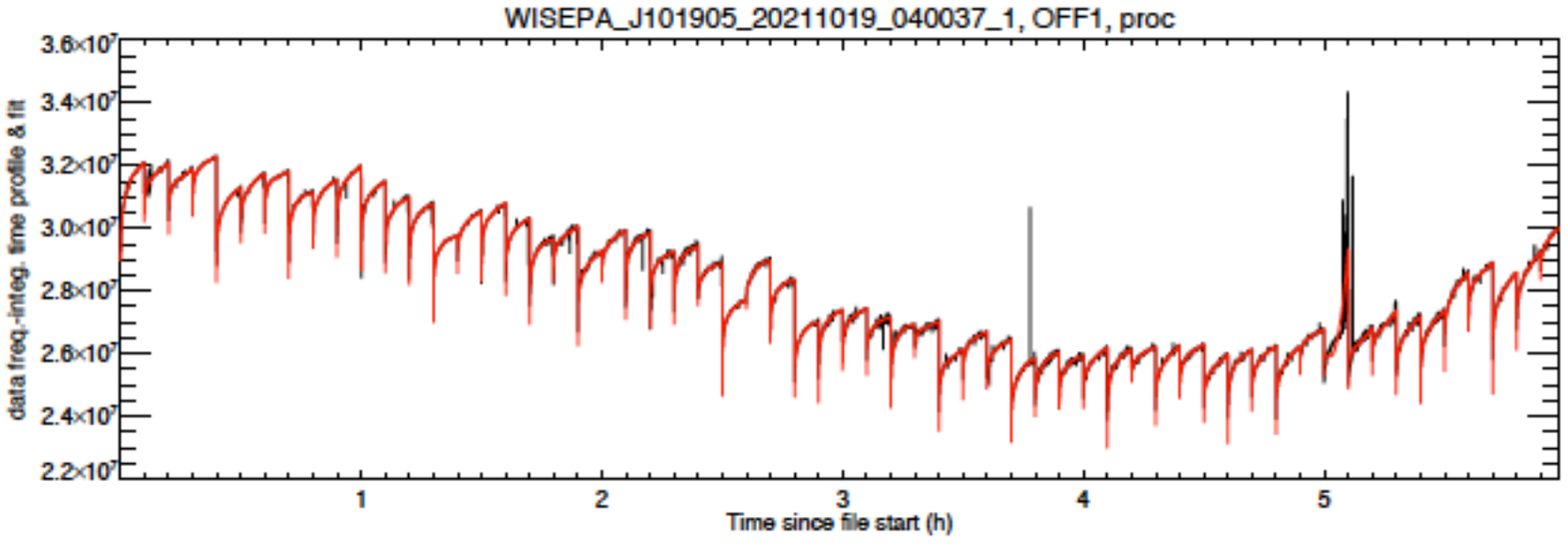
Quick-look example (WISEPA_J101905 observation)



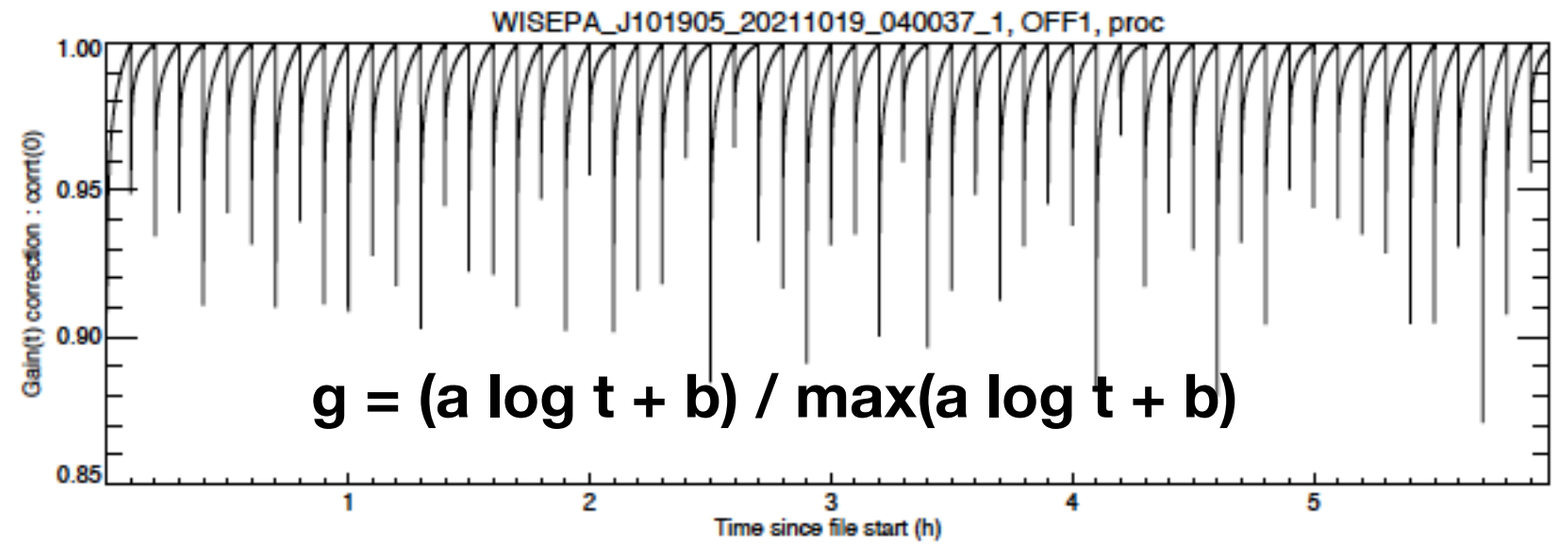
Quick-look example (WISEPA_J101905 observation)

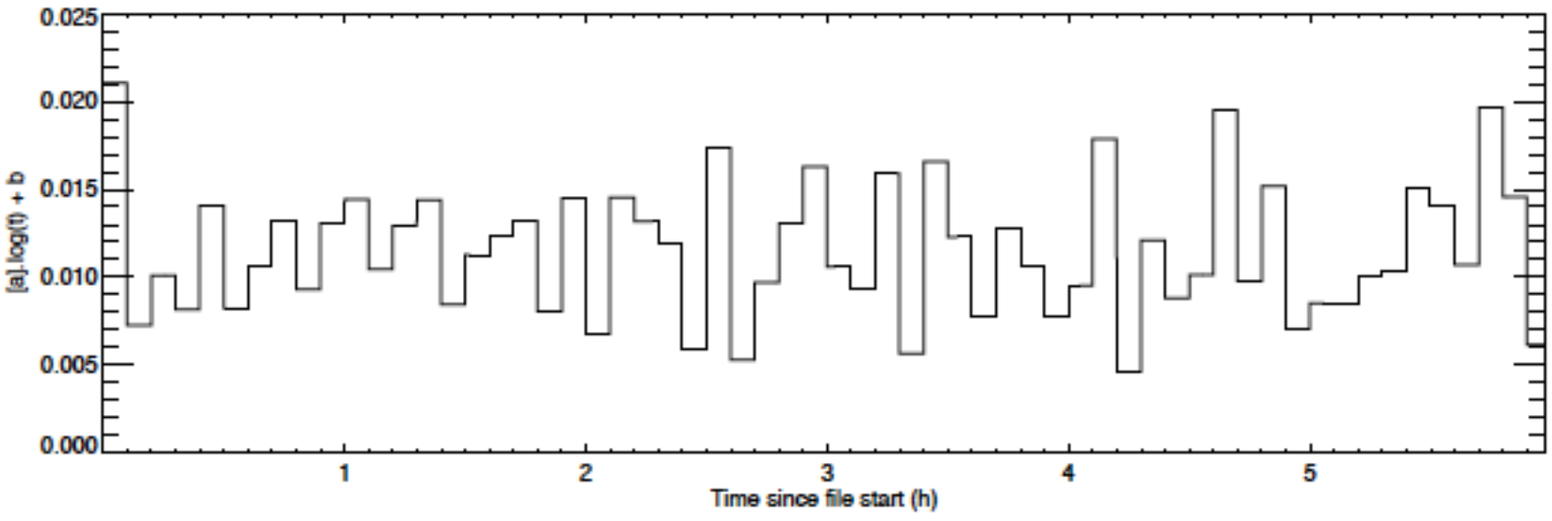


$$I = (a \log t + b)(ct^2+dt+e)$$

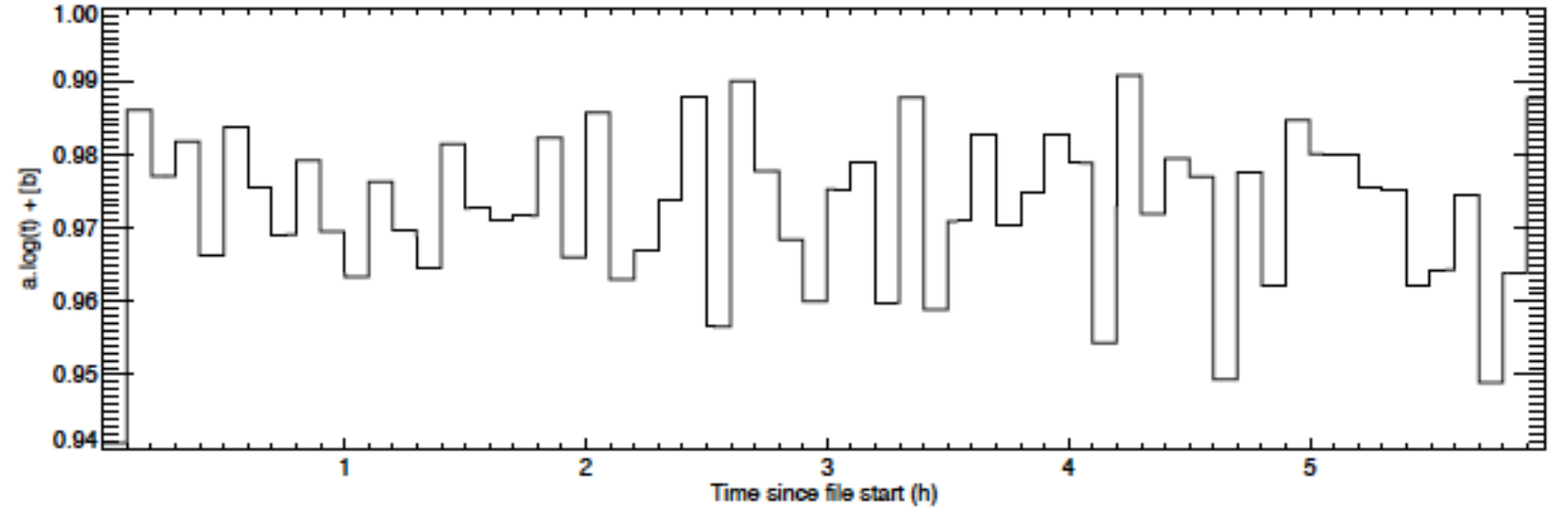


Quick-look example (WISEPA_J101905 observation)



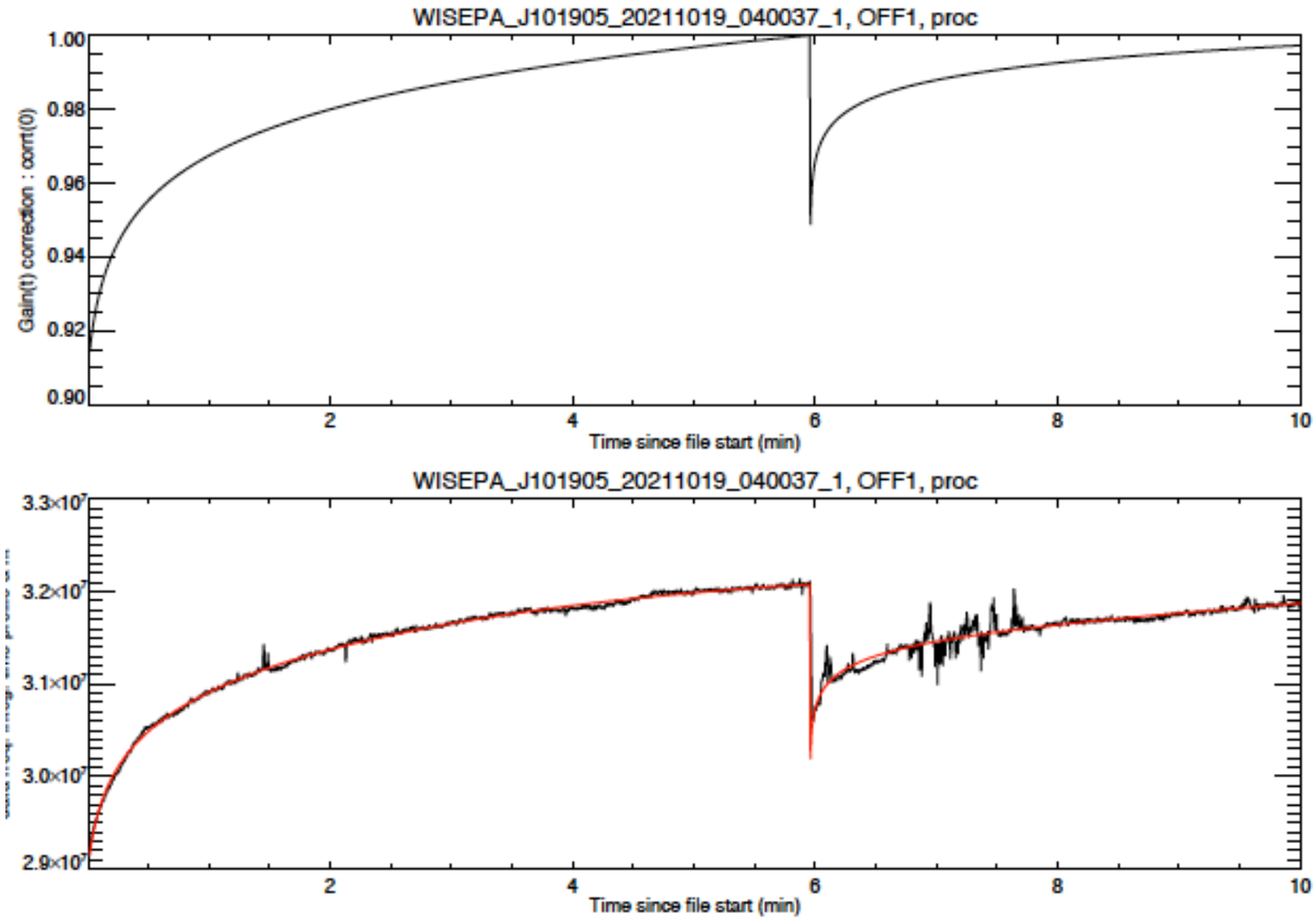


a

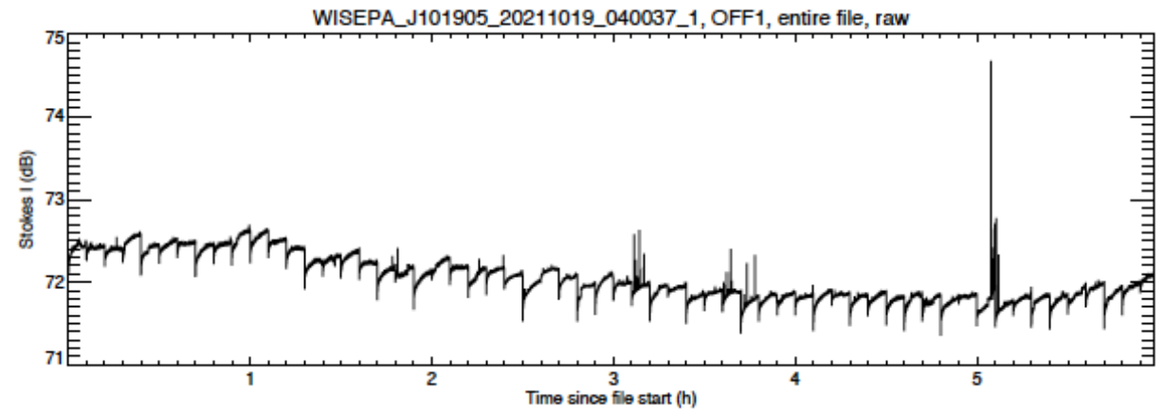


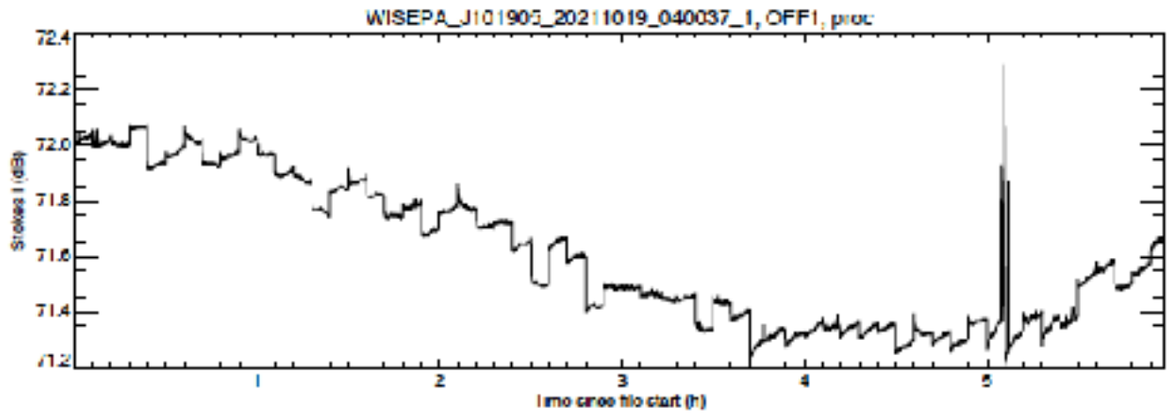
b

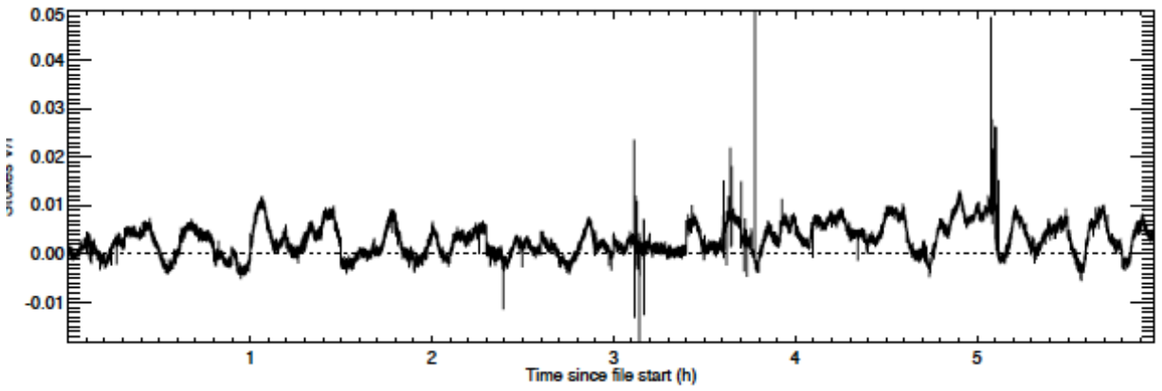
Quick-look example (WISEPA_J101905 observation)

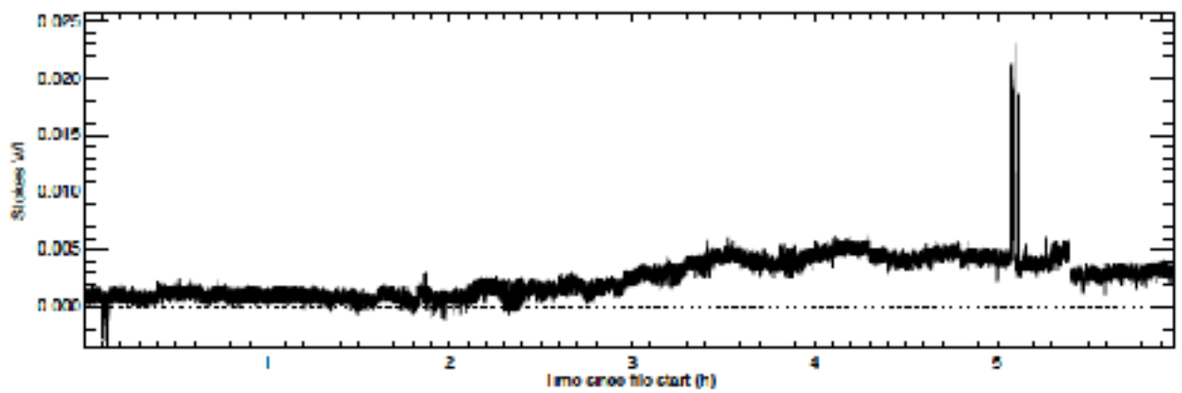


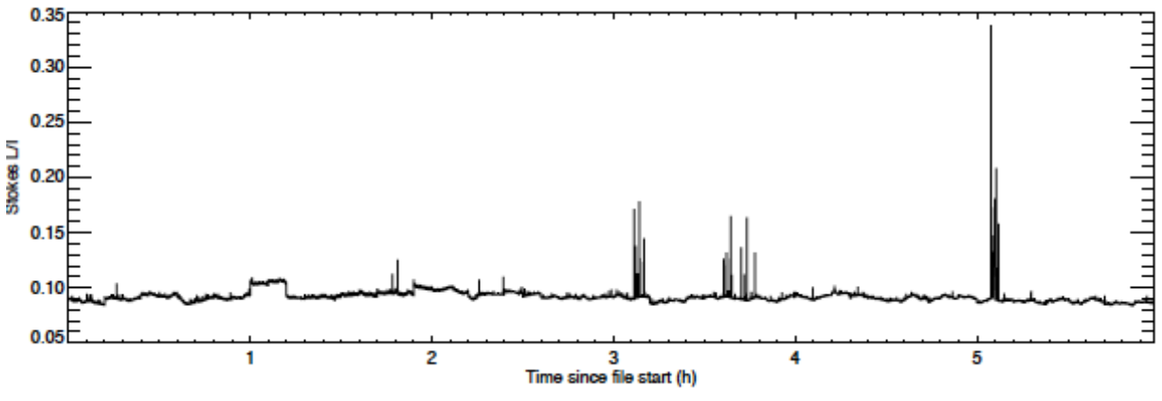
Quick-look example (WISEPA_J101905 observation)

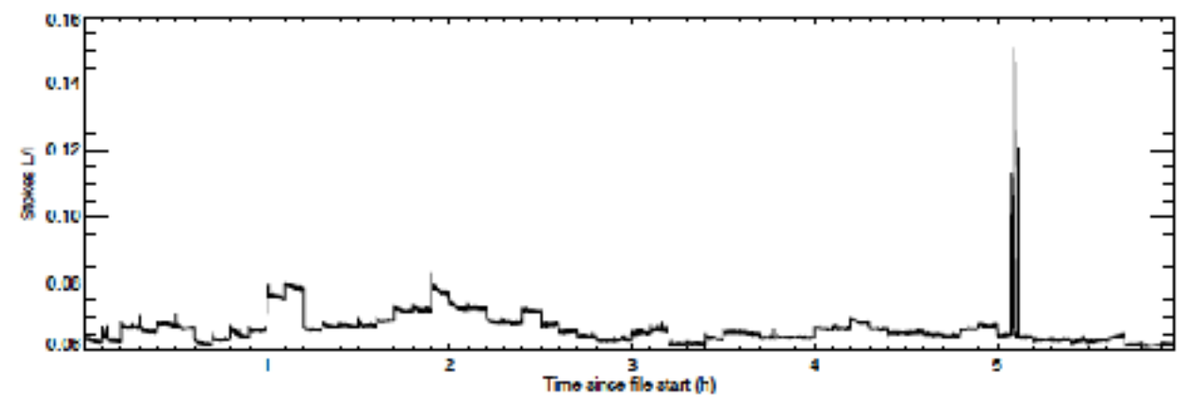


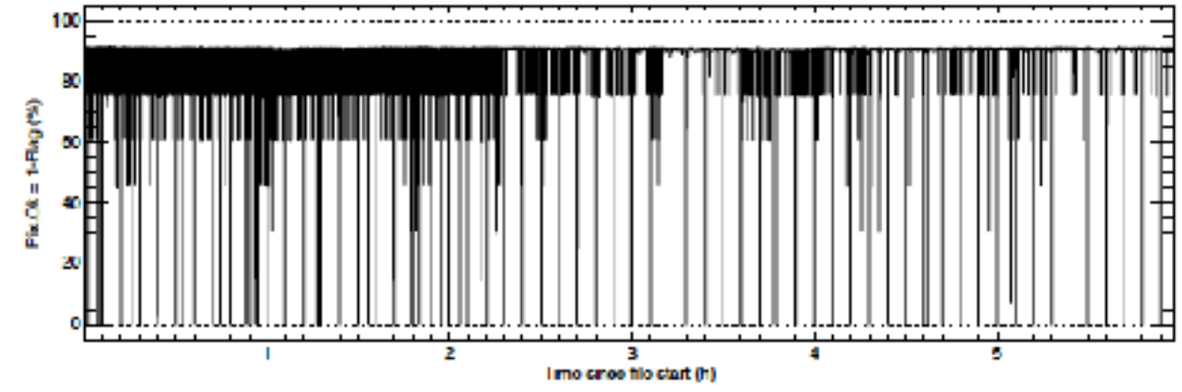




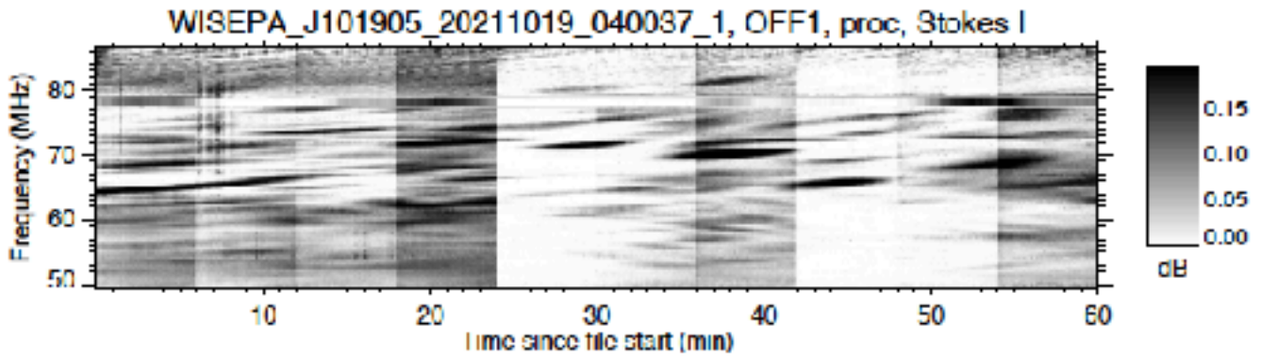


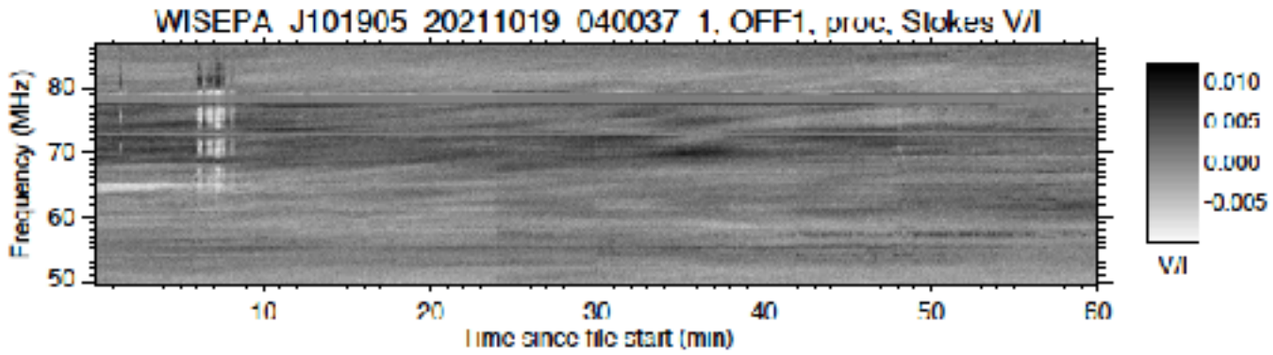


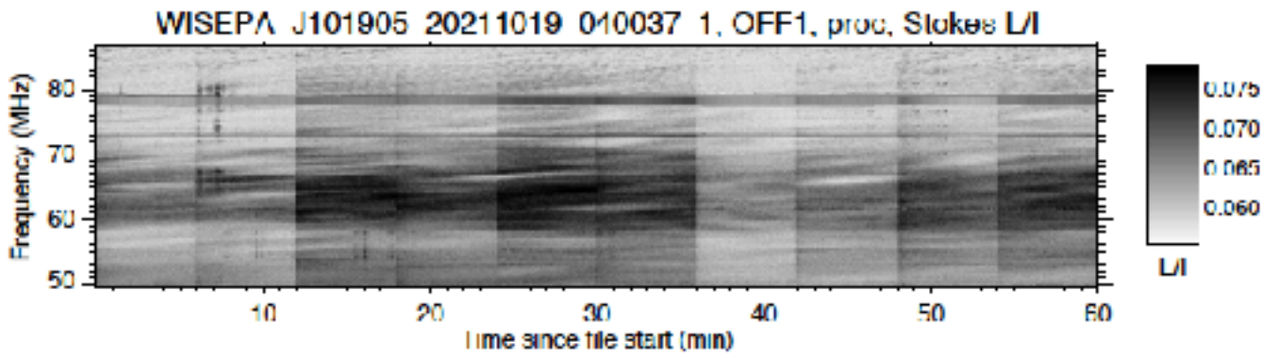


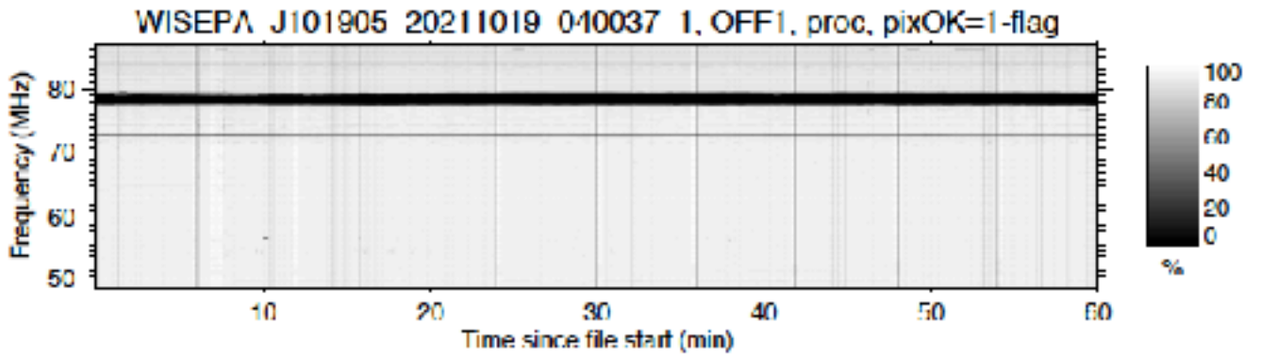


Quick-look example (WISEPA_J101905 observation)









Reading L1 data

- Documentation (pdf) on /cep/lofar/nenufar/pro/undysputed/docs/ **read_nu_spec.pdf**
- Code /cep/lofar/nenufar/pro/undysputed/**read_nu_fits.pro**

5.2 Contents & Re-reading of L1 FITS files

The command to run READ_NU_FITS.pro within IDL is :

```
READ_NU_FITS, file, command,param,variab, nt,dt,nf,df,ns,jd0,h0,fref,
      data,time,freq,beam,ndata,corrt,corrf, nodata=nodata, nstokes=nstokes, quiet=quiet
```

With keywords:
nstokes =0 [default] → read all nstokes =1 → read I
nstokes =2 → read IV nstokes =3 → read IVL
/nodata → reads header / housekeeping information only
/quiet → does not print any message while reading.

Estimating data volumes

- In the VCR when programming the observations
- Using /cep/lofar/nenufar/pro/general/**data_rate_vol.pro**

```
IDL command Examples:

    data_rate_vol,/hlp

DATA_RATE_VOL, Nma,Nch(/sb),dt(sec),Df(MHz),Nb(bytes/samp),Tobs(sec), D_tot(byte/s),V_tot(bytes), /BF, /IM, /WF, /SUM
```

data_rate_vol,/help

```
Computes and displays NenuFAR data rates and volumes

INPUTS
Nma  = number of Mini-Arrays involved
Nch  = number of channels / 195.3125 kHz subband
dt   = integration time of spectra / of visibility sets (sec)
Df   = total bandwidth (MHz)
Nb   = N bytes of raw data samples (1 = 8 bits, 2 = 16 bits)
Tobs = total observation time (sec)

KEYWORDS
/BF  => Beamformer mode   (uses Nch, dt, Df)
/IM  => Imager mode       (uses Nma, Nch, dt, Df)
/WF  => Waveform mode     (uses Df, Nb)
/TB  => Transient Buffer mode (uses Nma, Nb)
/SUM => 1 line SUMMARY

OUTPUTS
D_tot = rate (bytes/sec)
V_tot = volume (bytes)
```

Estimating data volumes

```
data_rate_vol,80,64,0.021,37.5,16,3600, /BF
```

```
Rate=      47.6  KB/s/SB
Rate=       8.9  MB/s
Rate=       4.5  MB/s/node
Volume=    31.4  GB   in   3600. sec
```

```
data_rate_vol,80,64,0.021,37.5,16,3600, /BF, /SUM
```

```
BF :  64 ch/SB   0.021 sec  37.5 MHz  3600. sec : Rate =  47.6 KB/s/SB    8.9 MB/s    4.5 MB/s/node : Volume =  31.4 GB
```

```
data_rate_vol, 0,64,0.021,37.5, 0,3600, /BF, /SUM
```

```
BF :  64 ch/SB   0.021 sec  37.5 MHz  3600. sec : Rate =  47.6 KB/s/SB    8.9 MB/s    4.5 MB/s/node : Volume =  31.4 GB
```

```
data_rate_vol, 0,2048,0.084,75, 0,3600, /BF
```

```
Rate=     381.0  KB/s/SB
Rate=     142.9  MB/s
Rate=      71.4  MB/s/node
Volume=   502.2  GB   in   3600. sec
```

→ 50 to 500 GB / hour for a typical tf observation

Other tools

- extract_nu_spec.pro

```
EXTRACT_NU_SPEC, file, slice_ext, blockmin, blockmax,
 tmin=tmin,tmax=tmax, path=path, info=info, help=help
```

- read_nu_raw.pro

```
READ_NU_RAW, file,fftlen, data,time,freq
```

Hands-on…

→ check

/databf2/nenufar/workshop/beamformed/bf-handson.pro

Nançay Decameter ARray

Daily Jupiter observations