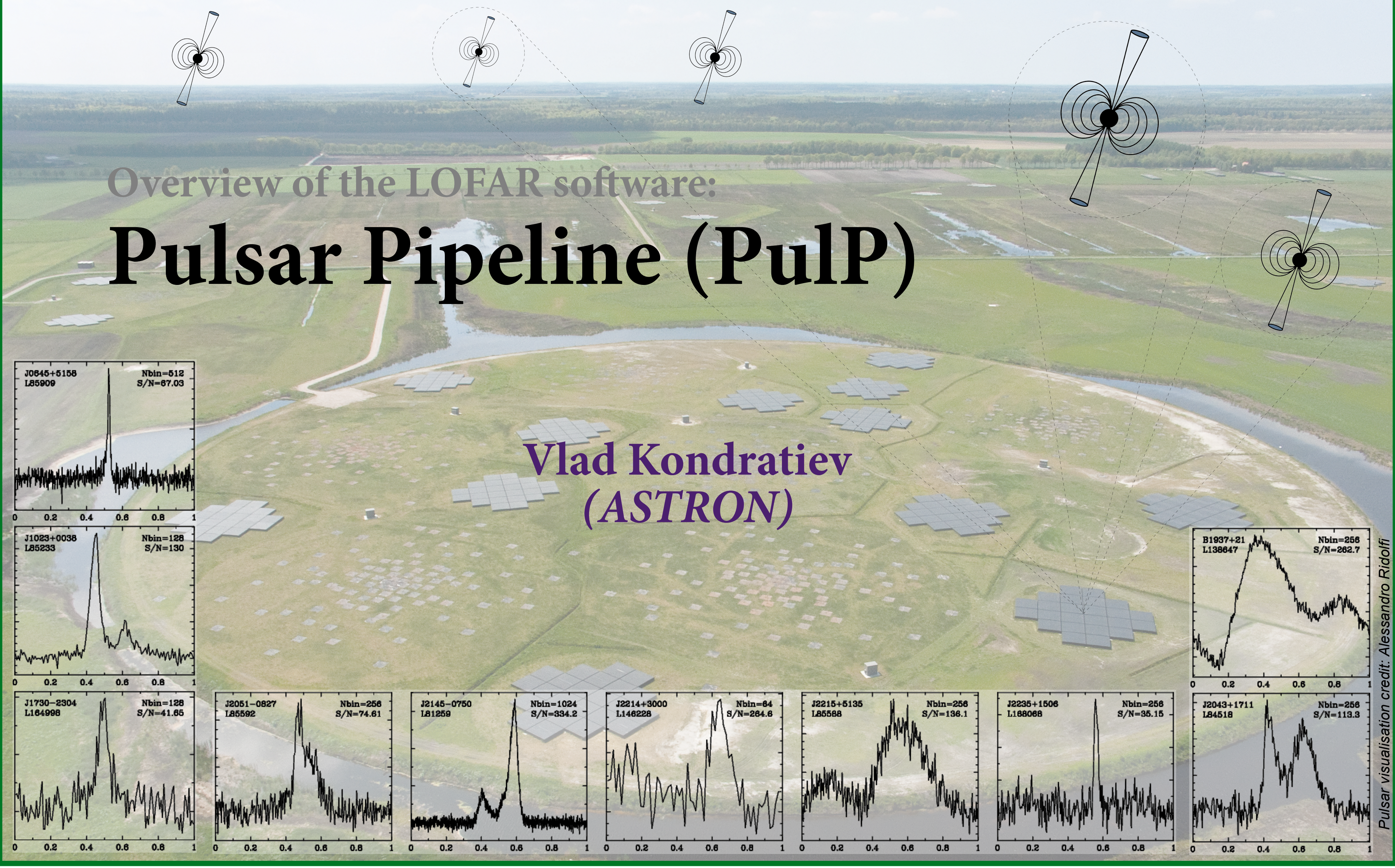
Overview of the LOFAR software:

Pulsar Pipeline (PulP)

Vlad Kondratiev
(ASTRON)



7th LOFAR Data Processing School — April 15, 2024

PulP overview

PulP is LOFAR **Pul**sar **P**ipeline for *known pulsars*. The essential goal of the PulP is to get the average profile of the pulsar(s) and provide a user with freq/time/phase/pol data cubes for further analysis. It is ***not*** the *search* pipeline, i.e. you can not do periodicity and single-pulse searches for a large range of dispersion measure trials. However, PulP can provide both PSRFITS/filterbank data and raw data converted to 8-bit for further searches.

PulP overview

PulP is LOFAR **Pul**sar **P**ipeline for *known pulsars*. The essential goal of the PulP is to get the average profile of the pulsar(s) and provide a user with freq/time/phase/pol data cubes for further analysis. It is ***not*** the *search* pipeline, i.e. you can not do periodicity and single-pulse searches for a large range of dispersion measure trials. However, PulP can provide both PSRFITS/filterbank data and raw data converted to 8-bit for further searches.

- Bookkeeping, service functions
  - → Logging
  - → Cluster configuration/settings
  - → User options
  - → Where input data are?
  - → Observing setup (HDF5 metadata / *parset*)
  - → Coordination of processing data for different TABs/frequency parts
  - → Feedback files for LTA ingest
- The actual data processing
- Diagnostic summaries and pipeline output data products

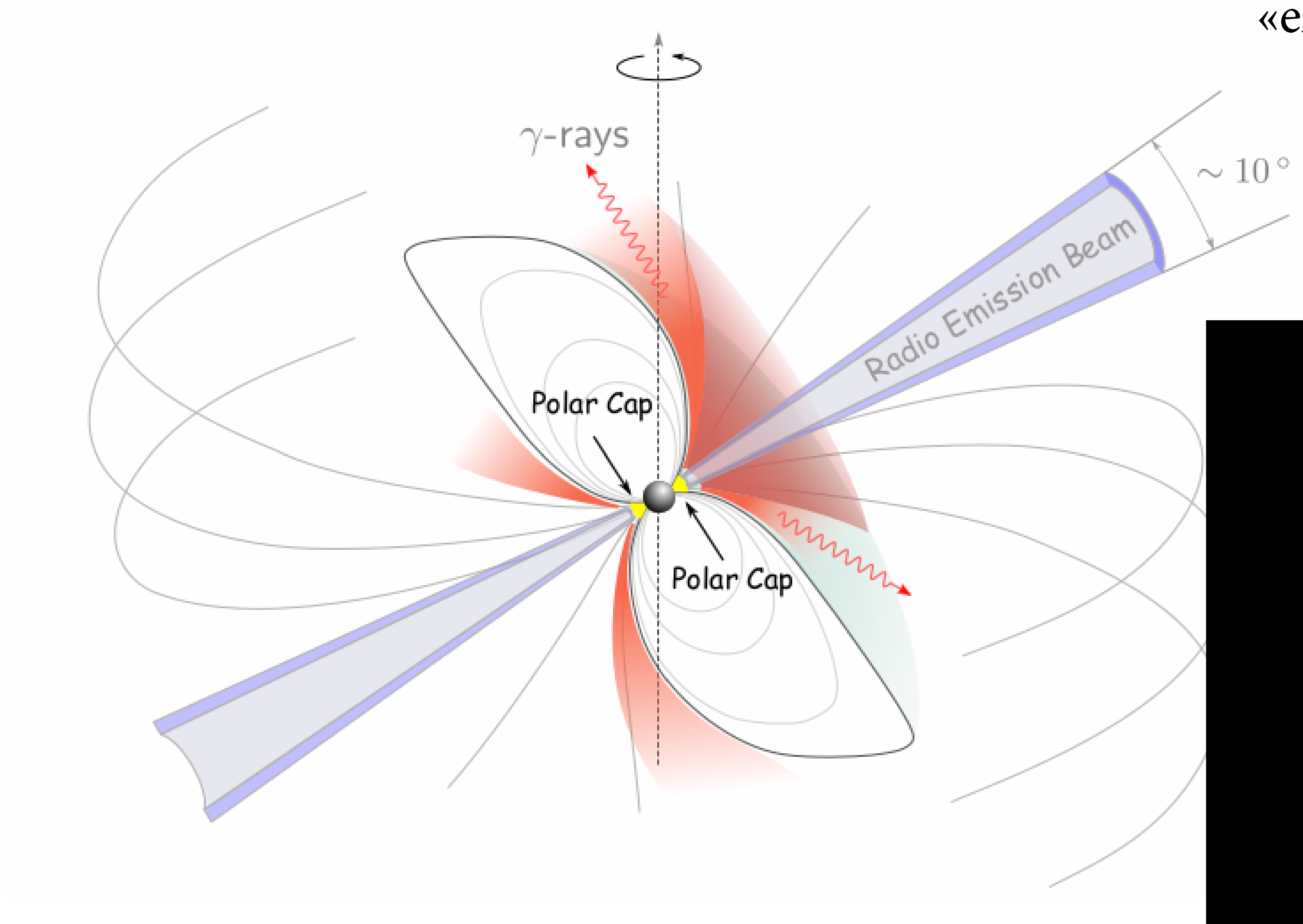
PulP overview

PulP is LOFAR **Pul**sar **P**ipeline for *known pulsars*. The essential goal of the PulP is to get the average profile of the pulsar(s) and provide a user with freq/time/phase/pol data cubes for further analysis. It is ***not*** the *search* pipeline, i.e. you can not do periodicity and single-pulse searches for a large range of dispersion measure trials. However, PulP can provide both PSRFITS/filterbank data and raw data converted to 8-bit for further searches.

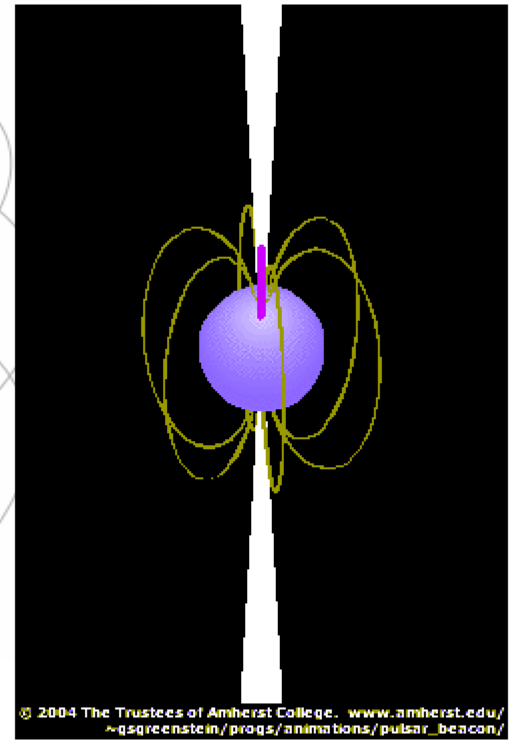
- Bookkeeping, service functions
  - → Logging
  - → Cluster configuration/settings
  - → User options
  - → Where input data are?
  - → Observing setup (HDF5 metadata / *parset*)
  - → Coordination of processing data for different TABs/frequency parts
  - → Feedback files for LTA ingest
- **The actual data processing**
- **Diagnostic summaries and pipeline output data products**

What is pulsar?

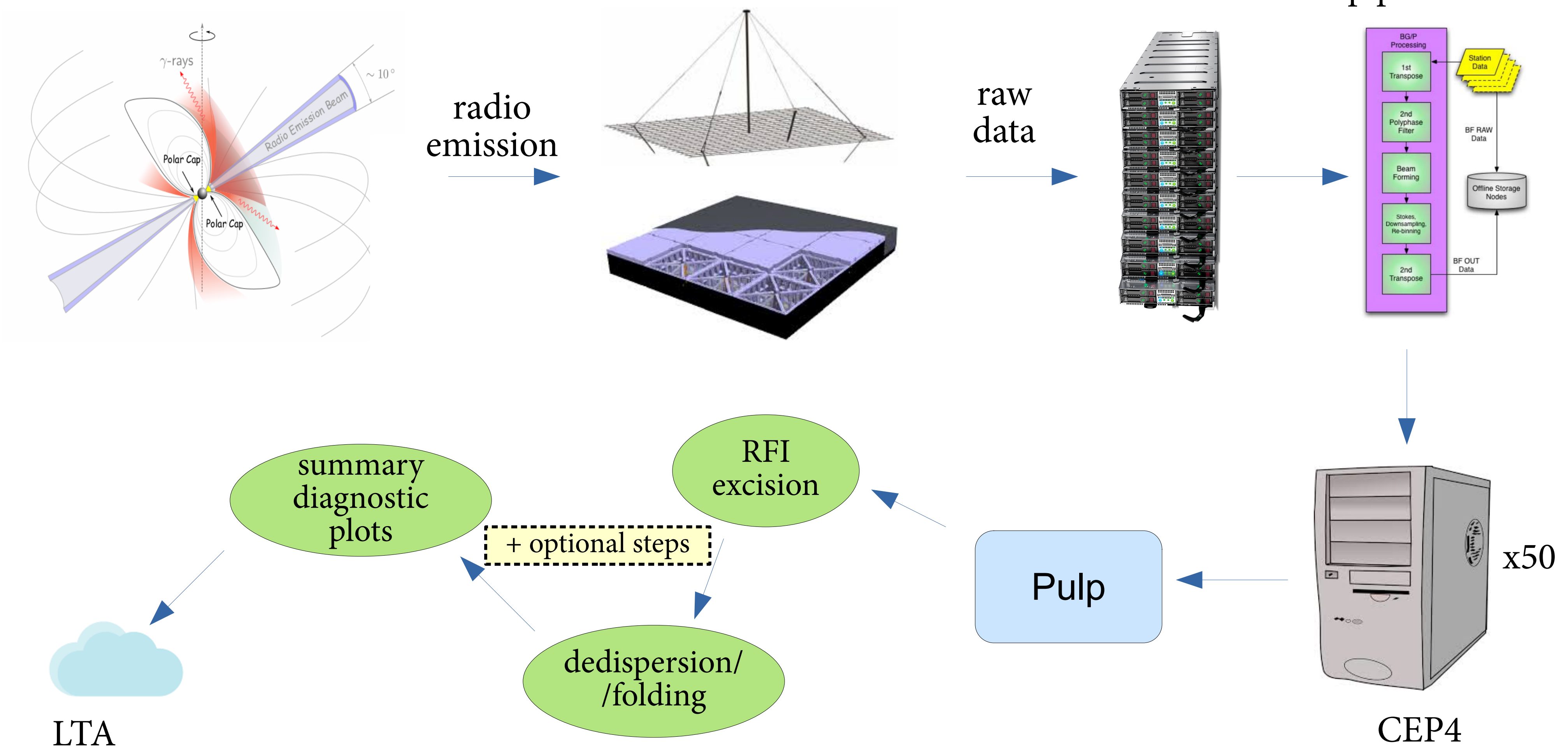
rapidly rotating
highly-magnetised neutron star,
«electric lighthouse»



Credit: Andrey Timokhin

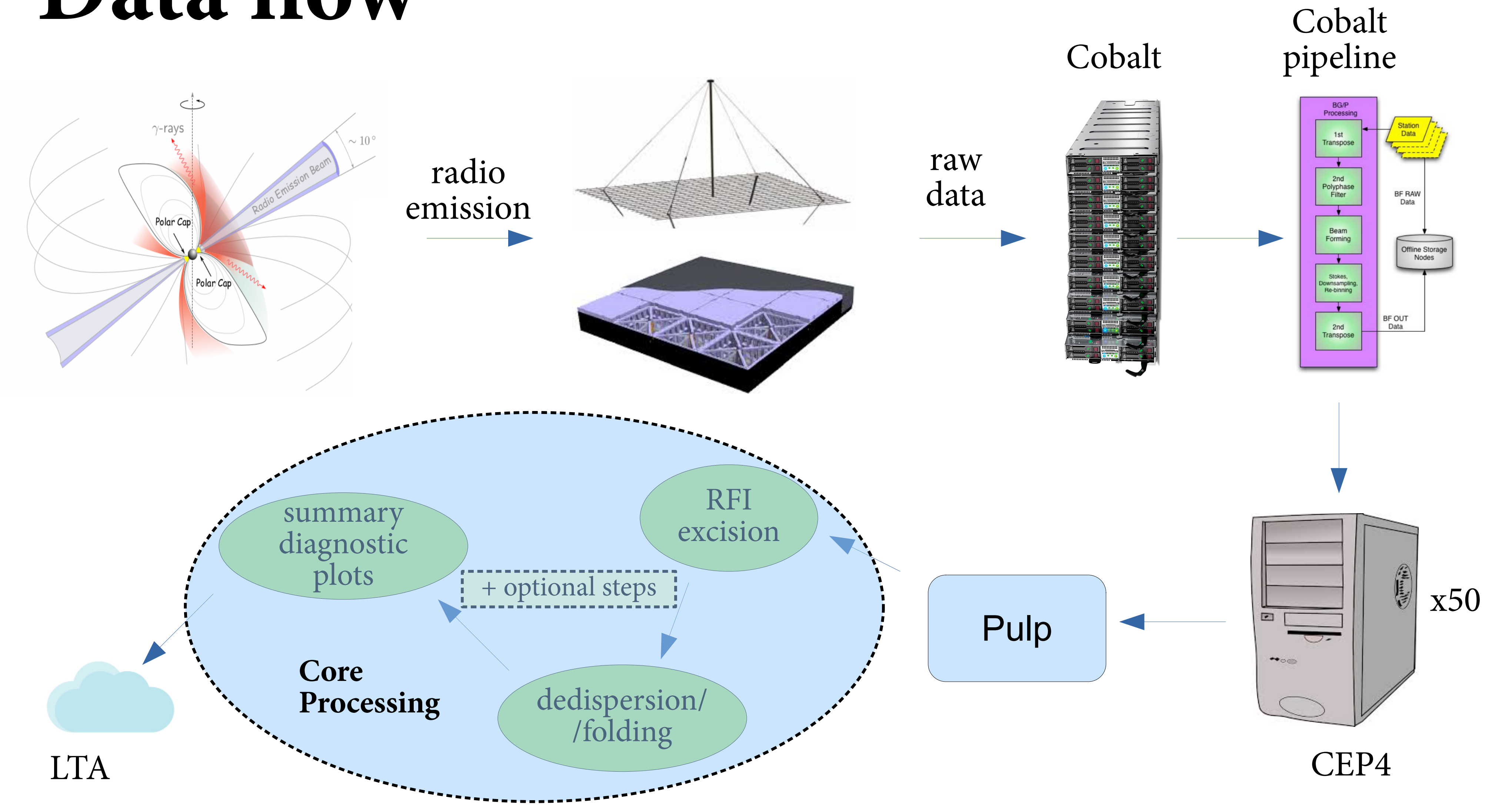


Data flow



γ-rays
~ 10°
Radio Emission Beam
Polar Cap
Polar Cap
radio
emission
raw
data
Cobalt
Cobalt
pipeline
BG/P
Processing
1st
Transpose
2nd
Polyphase
Filter
Beam
Forming
Stokes,
Downsampling,
Re-binning
2nd
Transpose
Station
Data
BF RAW
Data
Offline Storage
Nodes
BF OUT
Data
x50
CEP4
Pulp
RFI
excision
dedispersion/
/folding
+ optional steps
summary
diagnostic
plots
LTA

Data flow



γ-rays
~ 10°
Radio Emission Beam
Polar Cap
Polar Cap
radio
emission
raw
data
Cobalt
Cobalt
pipeline
BG/P
Processing
1st
Transpose
2nd
Polyphase
Filter
Beam
Forming
Stokes,
Downsampling,
Re-binning
2nd
Transpose
Station
Data
BF RAW
Data
Offline Storage
Nodes
BF OUT
Data
x50
CEP4
Pulp
RFI
excision
summary
diagnostic
plots
+ optional steps
dedispersion/
/folding
Core
Processing
LTA

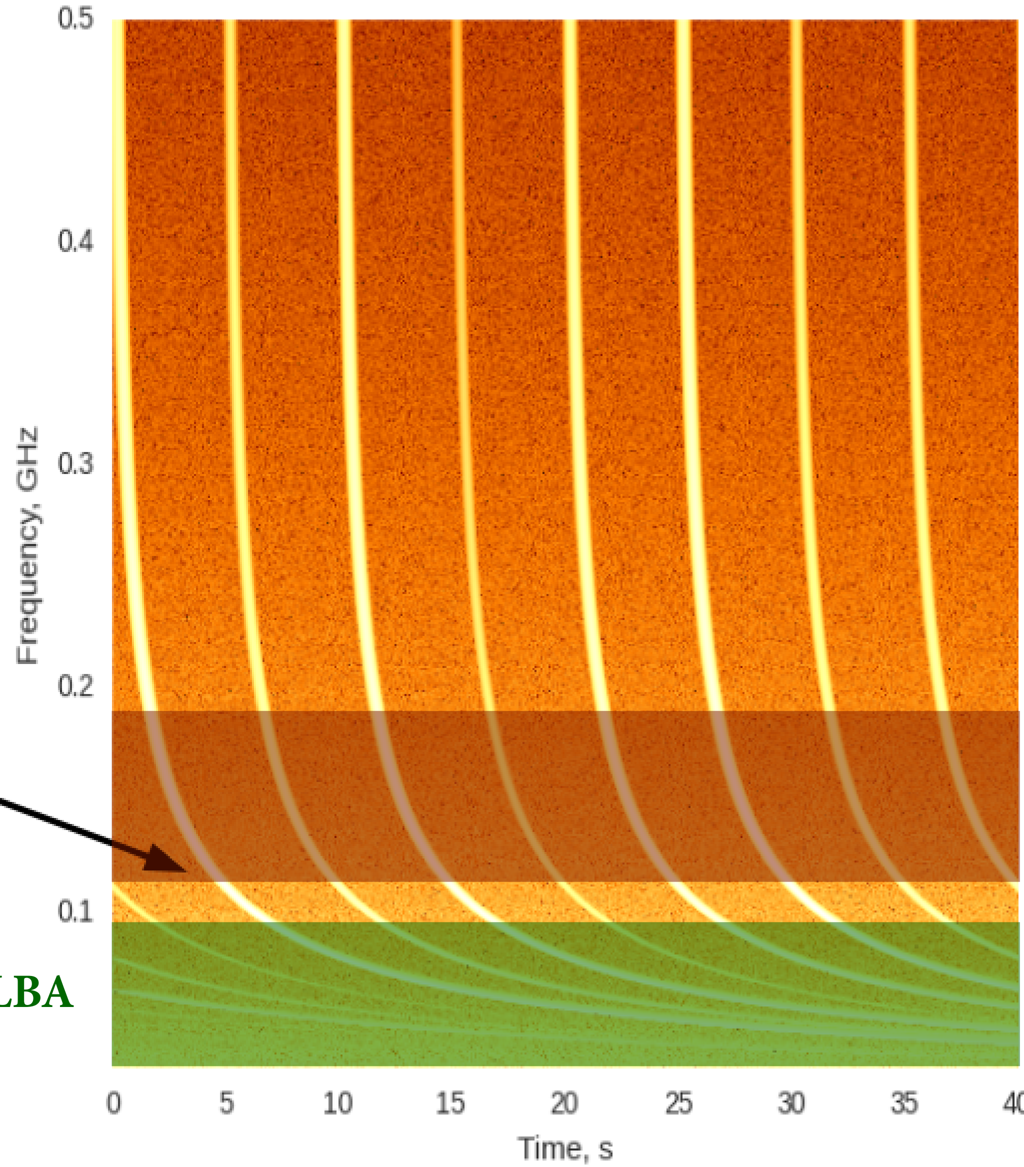
Dispersion

Simulated ultra-broadband pulse recording

DM = 15 pc cm^{-3}
P = 5 s

Dispersive delay

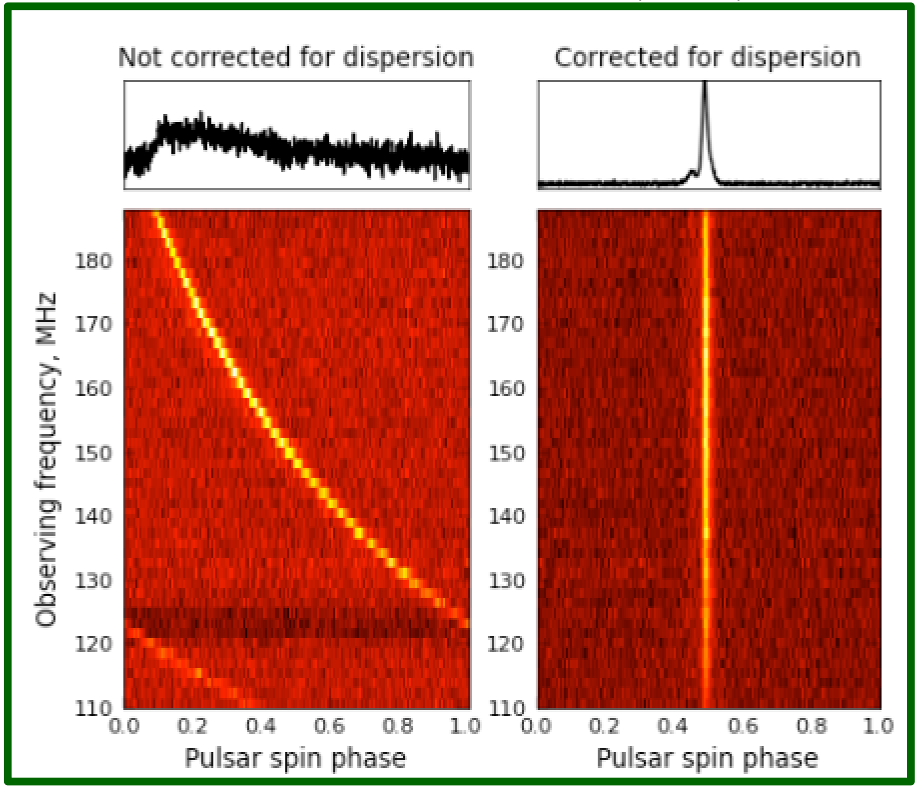
$$\delta t \sim DM / \nu^2$$



Credit: Anya Bilous

Dispersion

PSR B2021+51 **DM is off by only 3 pc/cc!**

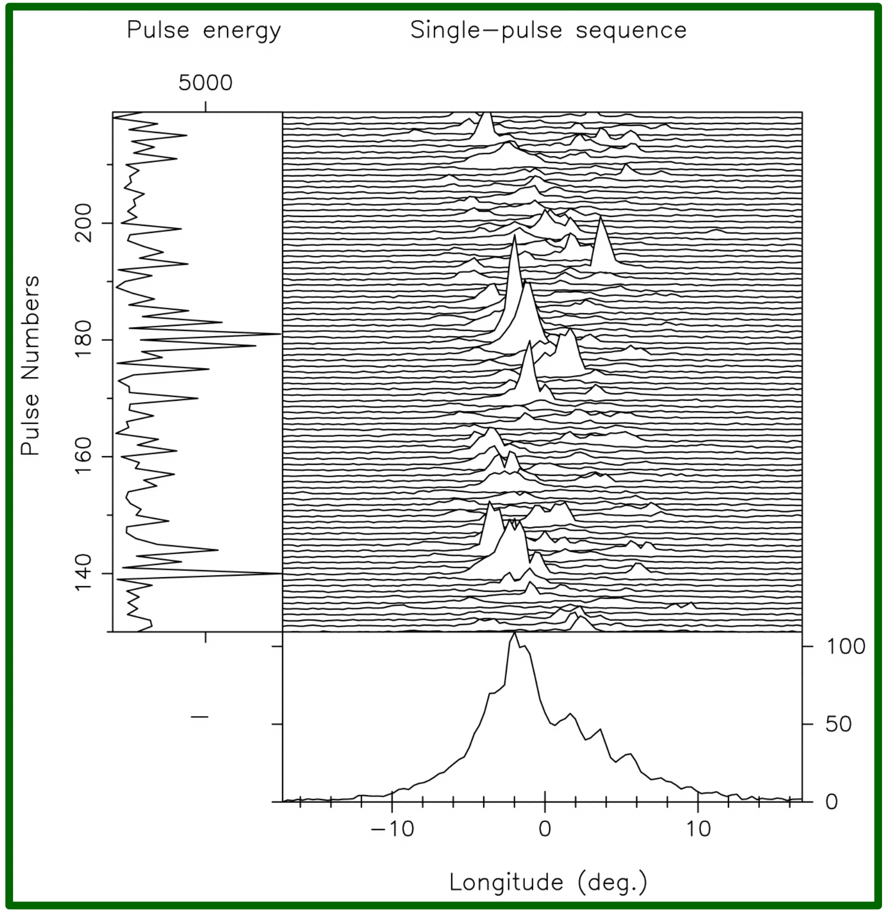


Credit: Anya Bilous

- DM [pc cm^{-3}] measures the integrated column density of free electrons along the line of sight
- Can be corrected using (in)coherent dedispersion

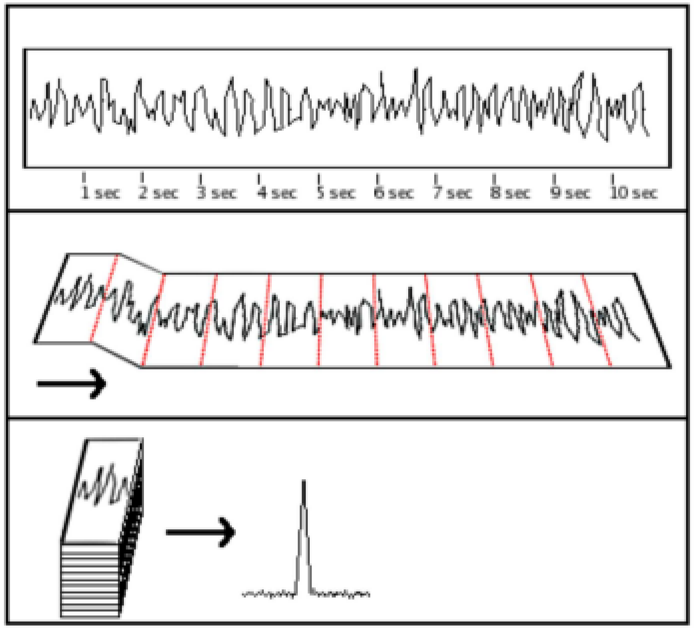
Folding

PSR B0943+10



Deshpande & Rankin (1999)

in a nutshell



Pulsar software (needed by PulP)

https://github.com/vkond/pulp

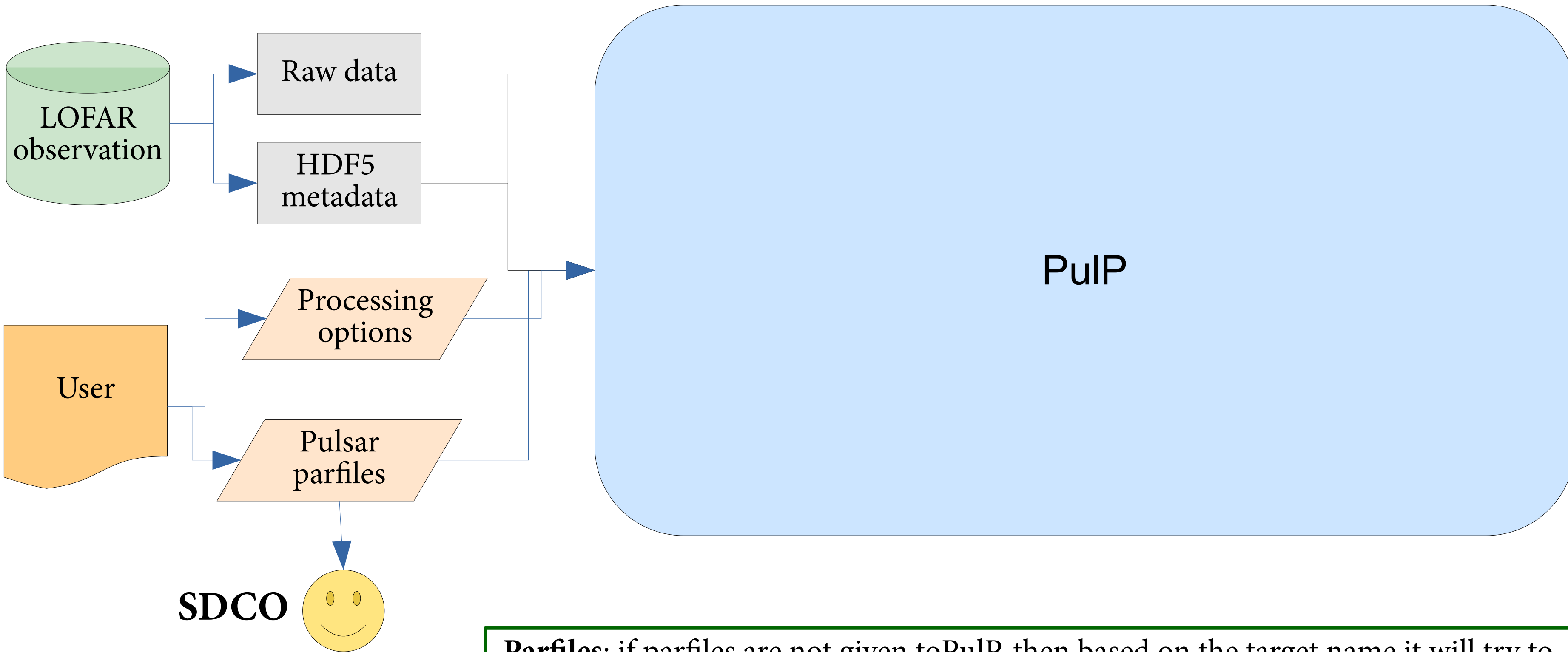
↓

https://git.astron.nl/ro/pulp

- FFTW
- PGPLOT, + python bindings
- TEMPO
- TEMPO2
- psrcat
- Sigproc
- **PRESTO** (by Scott Ransom, https://www.cv.nrao.edu/~sransom/presto/)
- psrdada
- **PSRCHIVE** (by Willem van Straten, http://psrchive.sourceforge.net/)
- DAL
- **DSPSR** (by Willem van Straten, https://dspsr.sourceforge.net)
- COAST_GUARD (by Patrick Lazarus, for RFI excision)
- LOFAR-BF-pulsar-scripts (by VK, https://github.com/vkond/LOFAR-BF-pulsar-scripts)

- in the future (needed for pulsar flux calibration):
  - → casacore
  - → python-casacore
  - → mscorpol

PulP flowchart (1)



Parfiles: if parfiles are not given toPulP, then based on the target name it will try to find the corresponding pulsar in the ATNF catalog. If no pulsar is found in the catalog, PulP will look for the brightest pulsar in a given SAP and fold it.

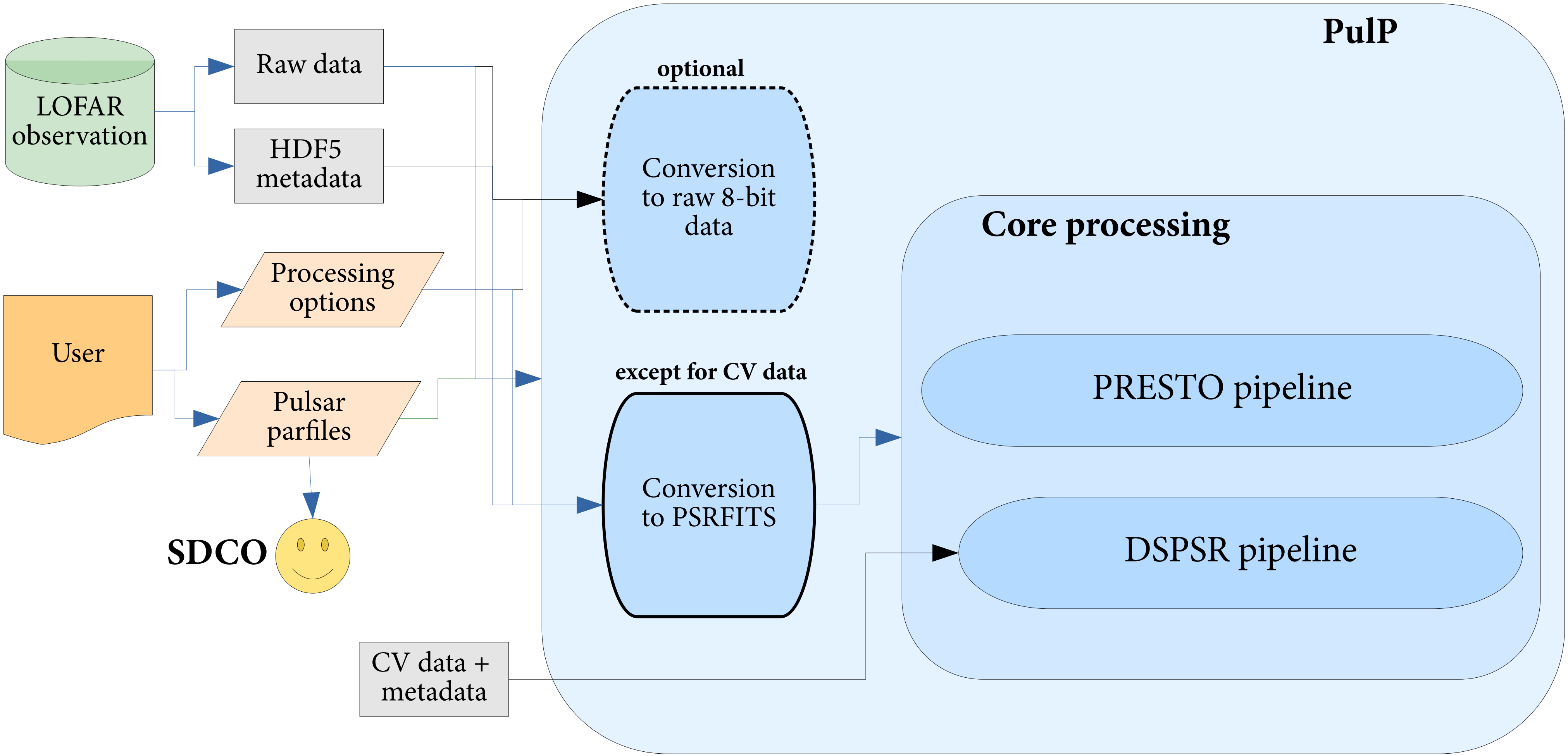
Pulsar ephemeris (parfiles)

```
PSRJ           J0034-0534
RAJ             00:34:21.8320019         1   0.000688440711207545 94
DECJ           -05:34:36.81231           1   0.02161483083208828989
F0             532.71342977772821597     1   0.00000002836577022925
F1             -9.33246370730386516 3e-16 1   4.926059192015837647 6e-17
PEPOCH         49550.037311801294202
POSEPOCH       49550.037311801294202
DMEPOCH        49550
DM             13.764894275846959064     1   0.00004430172861405973
DM1            0
PMRA           8.0823671616462304792         0.13169130726699274092
PMDEC          -9.5157740417196312044        0.30750778304651565920
BINARY         ELL1
PB             1.5892817926966151351     1   0.00000000792094121129
A1             1.4377774324431148653     1   0.00000359435416977026
TASC           49550.704855759820283     1   0.00003326831506895045
EPS1           6.4089927497823510916e-05 1   0.00000873904414835617
EPS2           -3.0385531645805153 1e-05 1   0.00001349161624088706
START          55959.632675467299123     1
FINISH         56448.301487586140865     1
TZRMJD         56190.012138005647902
TZRFRQ         137.15199999999998681
TZRSITE        t
TRES           31.090
EPHVER         5
CLK            TT(TAI)
MODE 1
EPHEM          DE421
NITS           1
NTOA           218
CHI2R          132.9672 207
```

can be as simple as this:

```
PSR       J1706+35
RAJ       17:07:03.61
DECJ      +35:55:54.5
P0        0.159764851
P1        0.0
PEPOCH    58244.04308936660
DM        19.240
EPHEM     DE405
CLK       UNCORR
```

PulP flowchart (2)



Input raw data

- HDF5 format
- Header information (metadata) is stored in *_bf.h5* file
- The raw data itself is stored in *_bf.raw* file
- This *.raw* file is linked from within *.h5* file and can be accessed directly via opened *.h5* file

- Filename structure:
  → Lnnnnnn_SAPxxx_Byyy_Sz_Pmmm_bf.h5

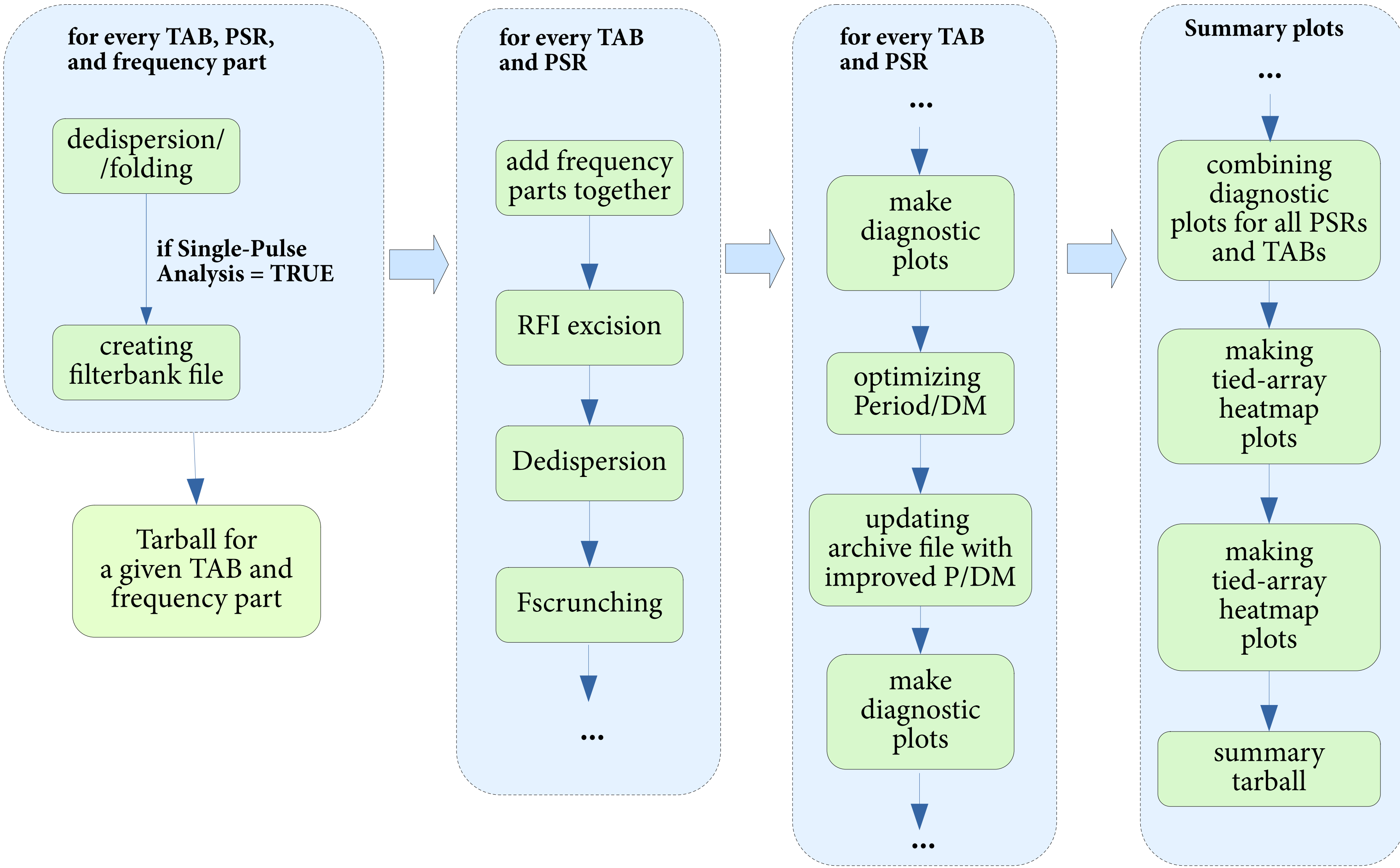
    - Lnnnnnn – LOFAR observation ID (ObsID)
    - xxx – Sub-array pointing (SAP) number
    - yyy – Tied-array beam (TAB) number
    - z – Stokes parameter, can only take values 0,1,2,3
        - Stokes I observation – have only S0 files
        - Stokes IQUV observation: S0 – I, S1 – Q, S2 – U, S3 – V
        - Complex-voltage data: S0 – Xreal, S1 – Ximag, S2 – Yreal, S3 – Yimag
    - mmm – Frequency part, i.e. when every file has only fraction of subbands

Data conversion

- Conversion to raw 8-bit data (optional)
  - → *digitize.py*
  - → written by Marten van Kerkwijk
  - → available at: *https://github.com/mhvk/scintellometry/blob/master/scintellometry/lofar/digitize.py*
  - → *digitize.py -s 5 -o <output dir> <input .h5>*

- Conversion from raw 32-bit data to PSRFITS (for non-CV data)
  - → custom-made program *2bf2fits*
  - → written by Tom Hassall, Patrick Weltevrede, with contribution from Vlad Kondratiev
  - → currently available at LOFAR Users Software Repository
  - → will make it available at Github as well
  - → does not save scales/offsets in PSRFITS
  - → needs major revisiting…
  - → Command example (very detailed input):
    - *2bf2fits -CS -H -append -nbits 8 -A 100 -sigma 3 -nsubs 400 -sap 0 -tab 0 -stokes 0 -o L667444_SAP0_BEAM0 -nsamples 24 -nchans 16 -ra 2.15980858832 -dec 1.30000703891 -psr B0809+74 -clock 200 -band HBA_110_190 -startdate 2018-09-12 -starttime 20:17:00.000000000 -samptime 0.0104858 -duration 299.977 -subs 54..453 -obsid L667444 -observer Pizzo /data/projects/PipelineTests/L667444/cs/L667444_SAP000_B000_S0_P000_bf.raw*

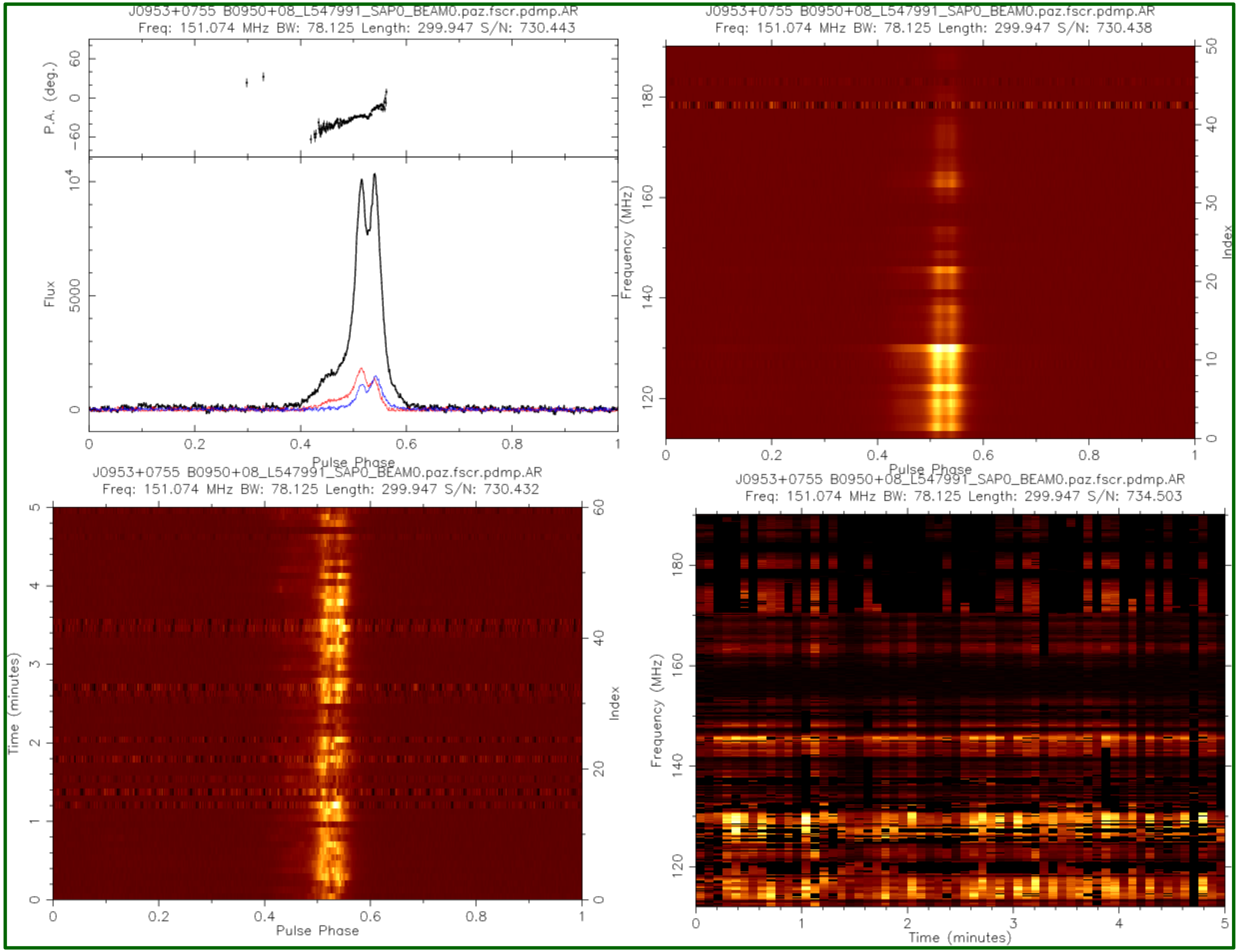
DSPSR Pipeline



PulP output data

- Raw data in 8-bit format (optional)
- Raw .h5 metadata files
- Pulsar data cubes (both from PRESTO and DSPSR pipelines)
- PRESTO pipeline:
  → rfifind mask
  → PSRFITS filterbank data
- DSPSR pipeline:
  → filterbank file(s) when SP analysis was done (optional)
- Single-pulse data (optional)
  → .singlepulse
  → Single-pulse plots
- Diagnostic plots
  → Plot with multiple profiles (multiple TABs, etc.) – *combined.png*
  → DSPSR diagnostic plots – *status.png*
  → Localization maps – *TAheatmap_*.png*

DSPSR Pipeline – plots (1)



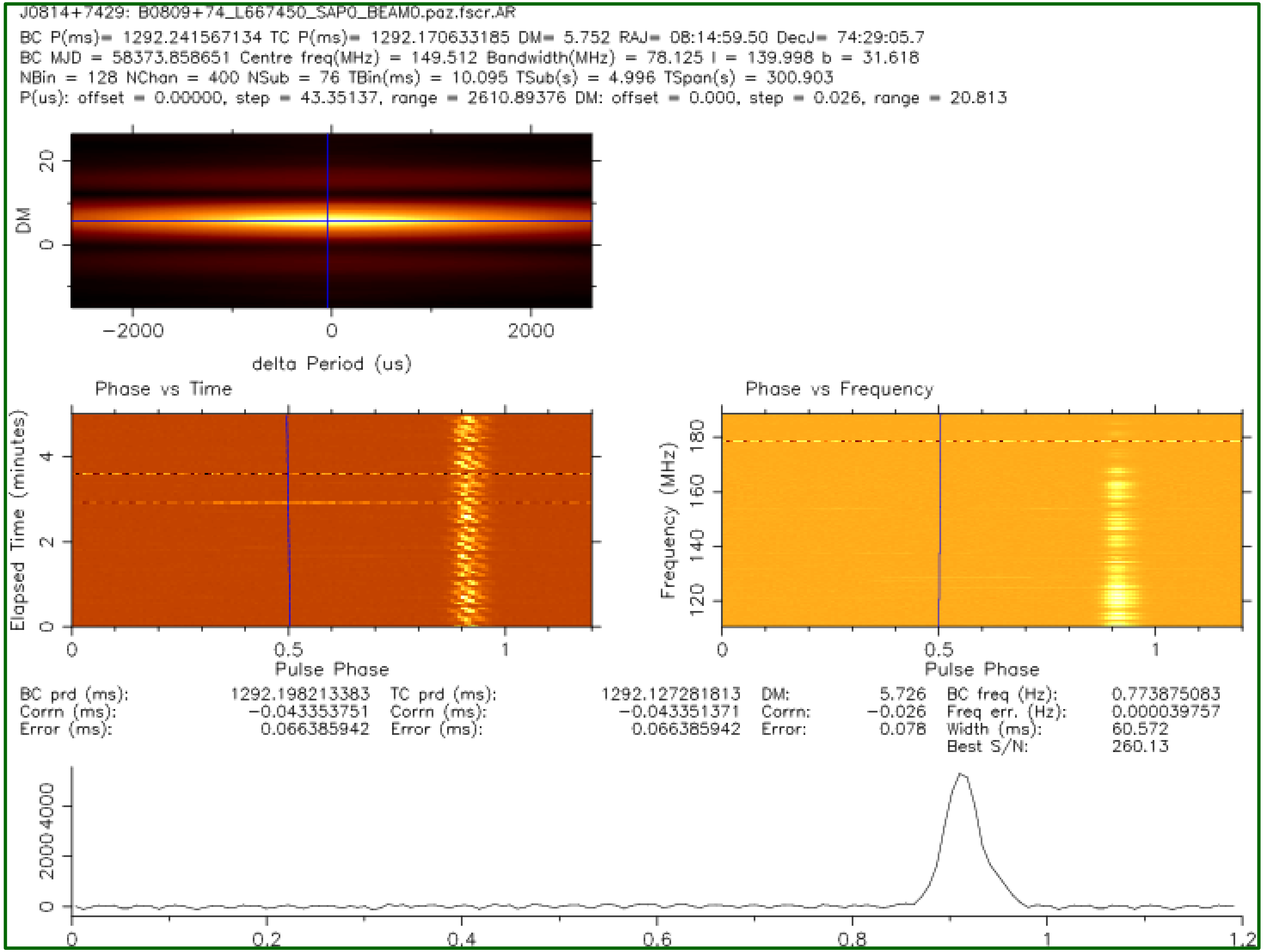
status.png

*_diag.png

*_diag_pdmp.png

DSPSR Pipeline – plots (2)

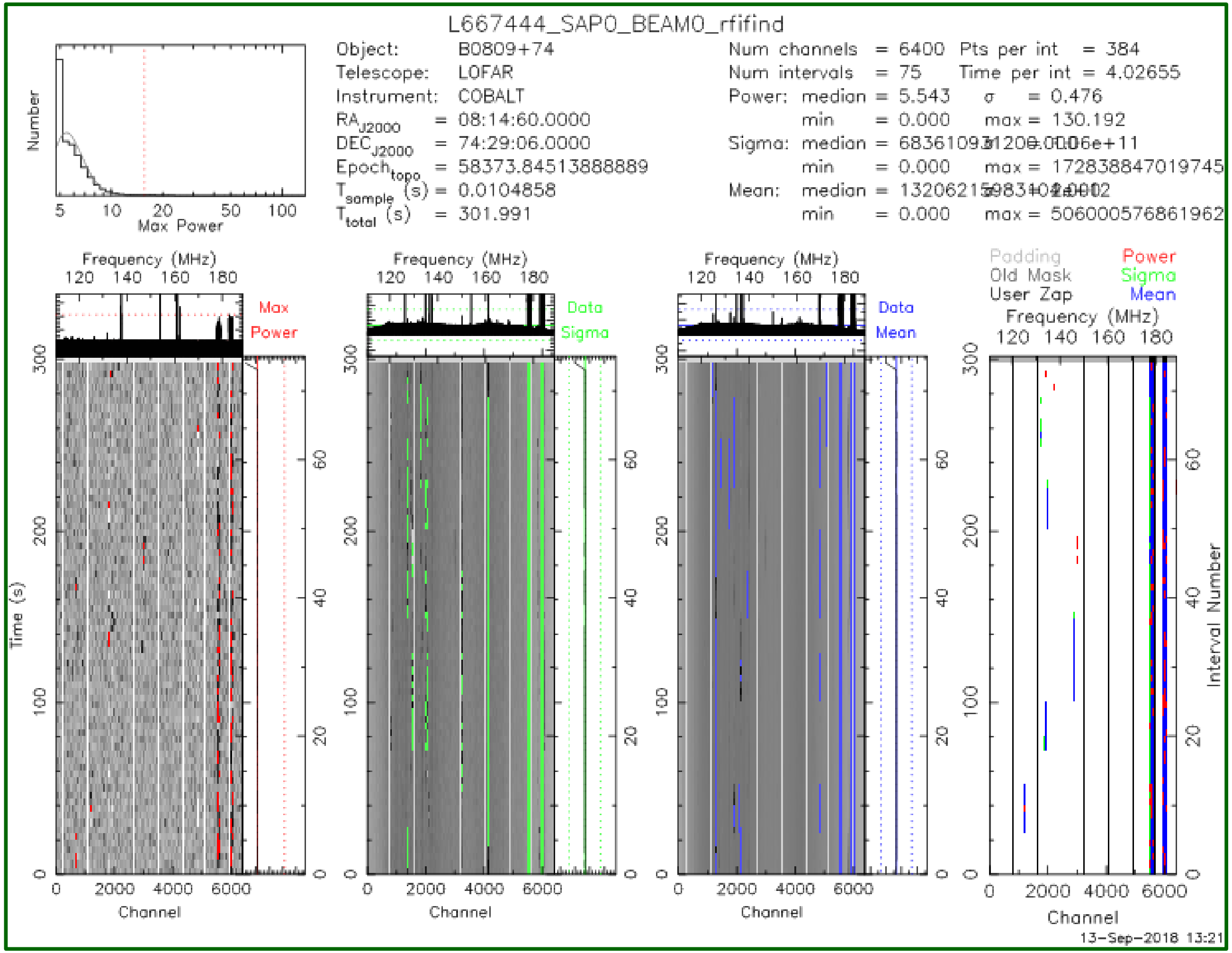
pdmp



*_pdmp.ps

PRESTO Pipeline – plots (3)

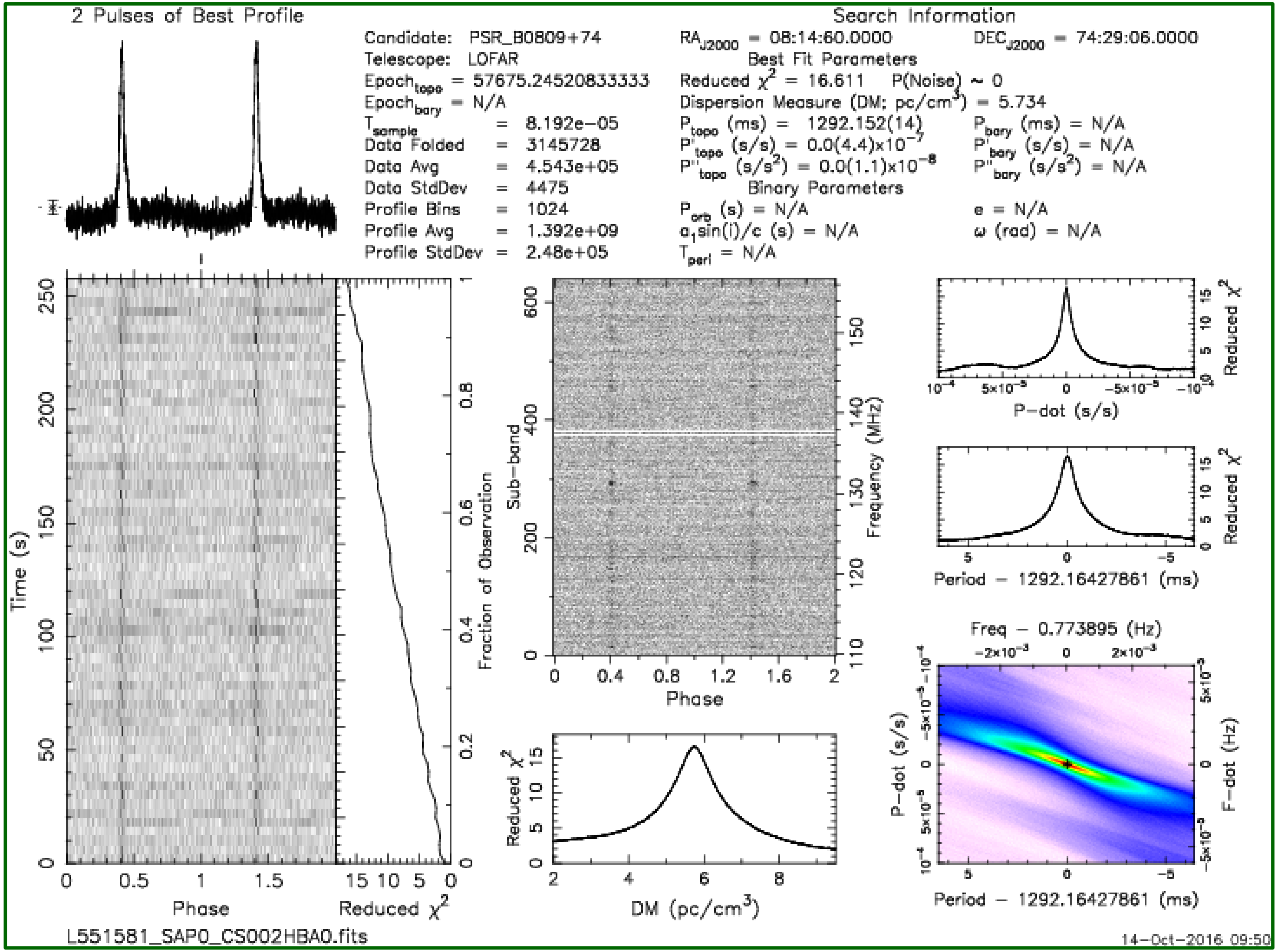
rfifind



*_rfifind.ps

PRESTO Pipeline – plots (4)

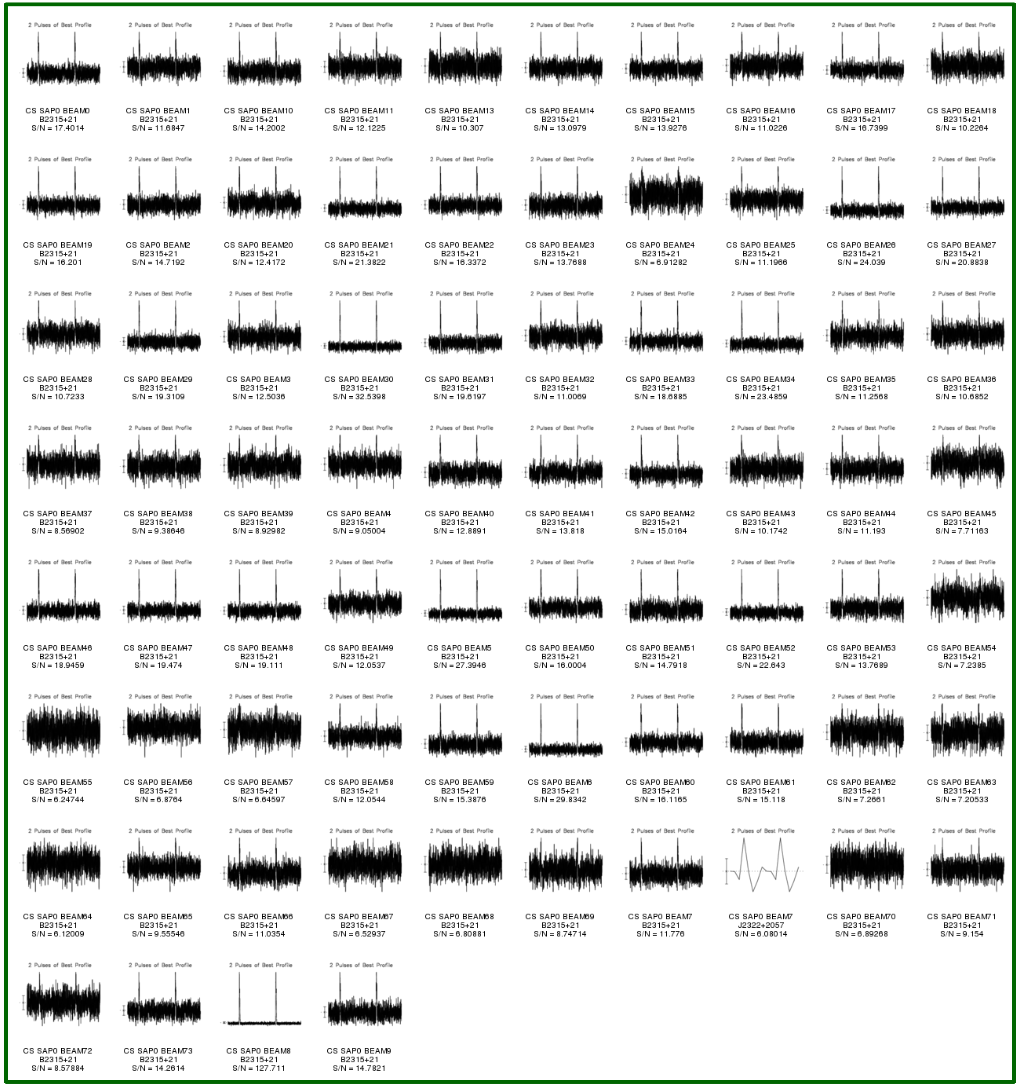
prepfold



*_pfd.png

PRESTO Pipeline – plots (5)

multiple TABs
prepfold

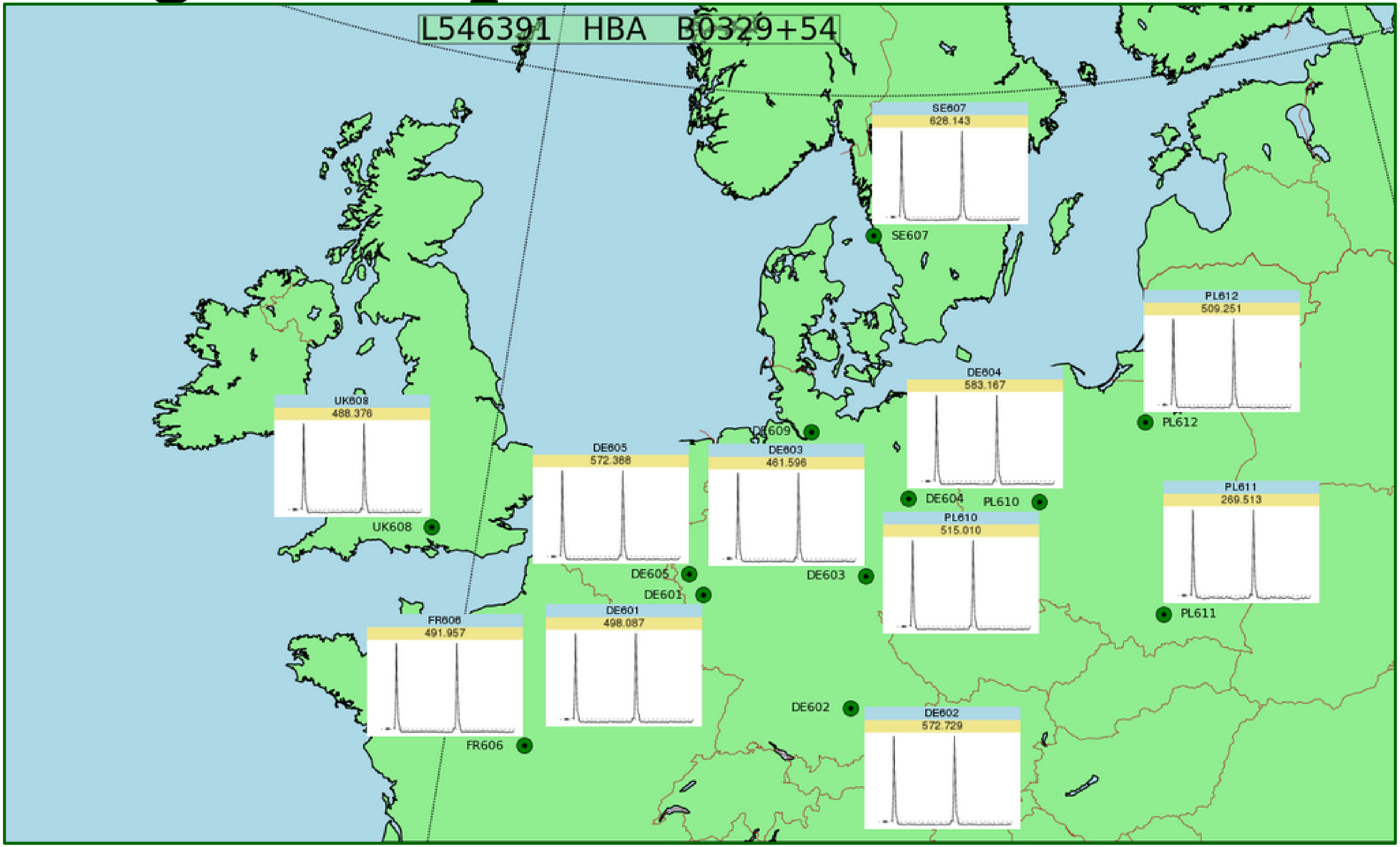


combined.png

Diagnostic plots (6)

for FE observations

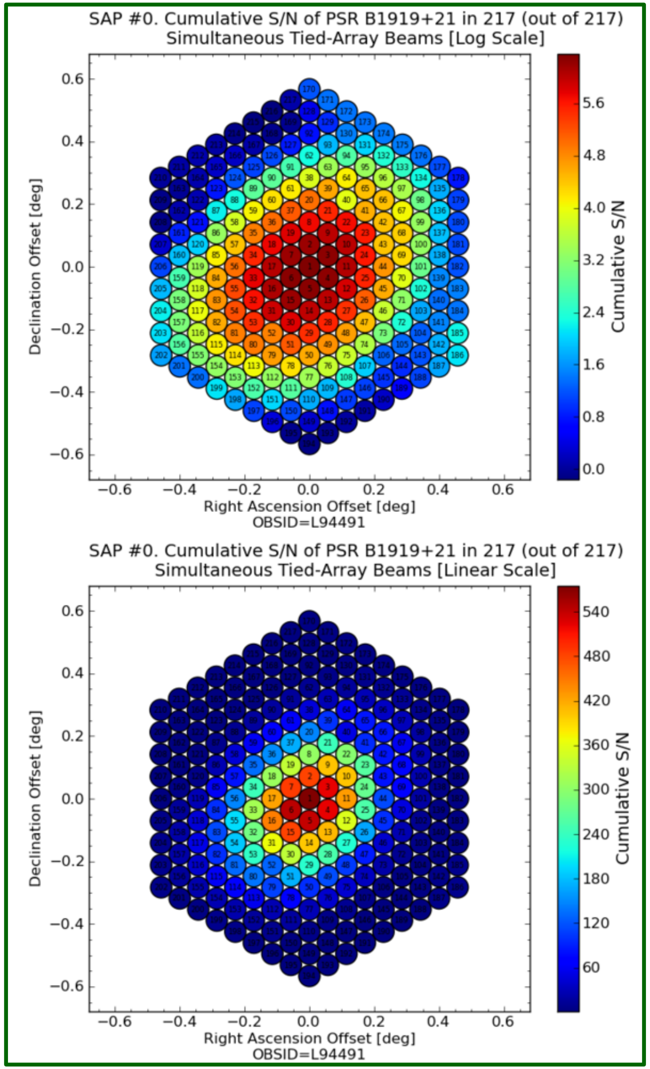
status.png



Diagnostic plots (7)

multiple TABs

TAheatmap_*.png




time for questions...
Pulsar visualisation credit: Alessandro Ridolfi

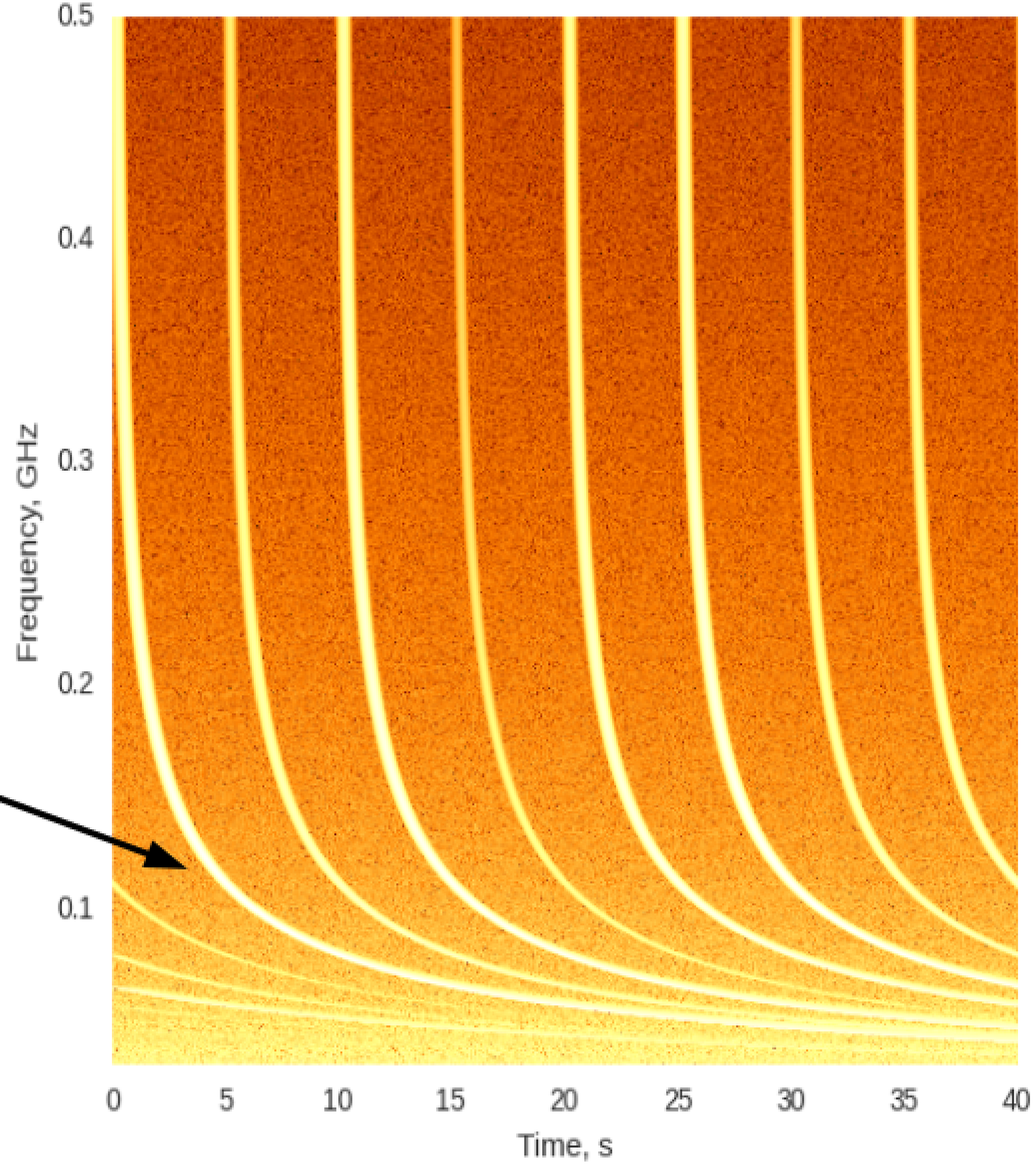
Dispersion

Simulated ultra-broadband pulse recording

DM = 15 pc cm^{-3}
P = 5 s

Dispersive delay

$$\delta t \sim DM / \nu^2$$



Credit: Anya Bilous

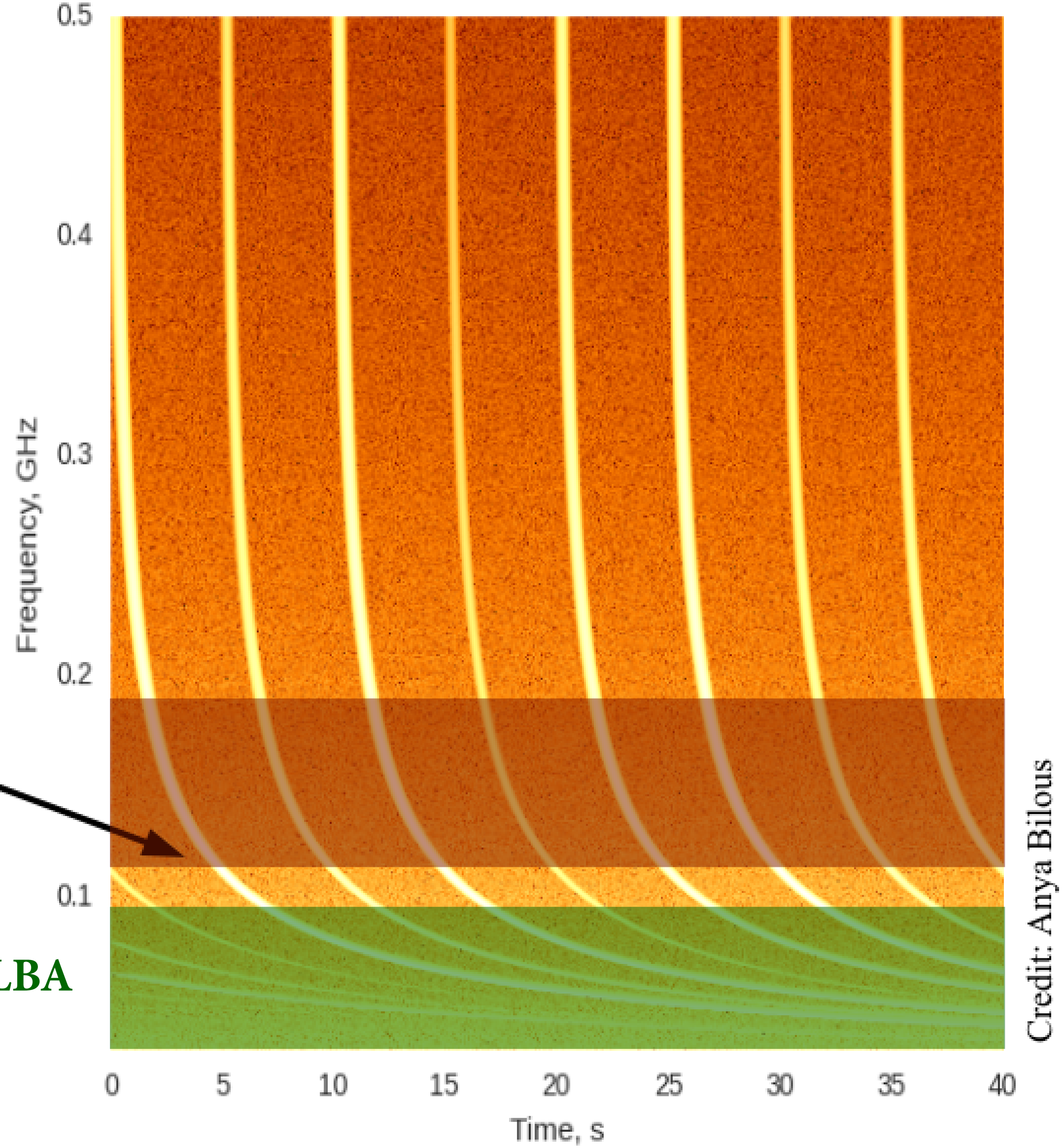
Dispersion

Simulated ultra-broadband pulse recording

$DM = 15\ \mathrm{pc\ cm^{-3}}$
$P = 5\ \mathrm{s}$

Dispersive delay

$$\delta t \sim DM / \nu^2$$

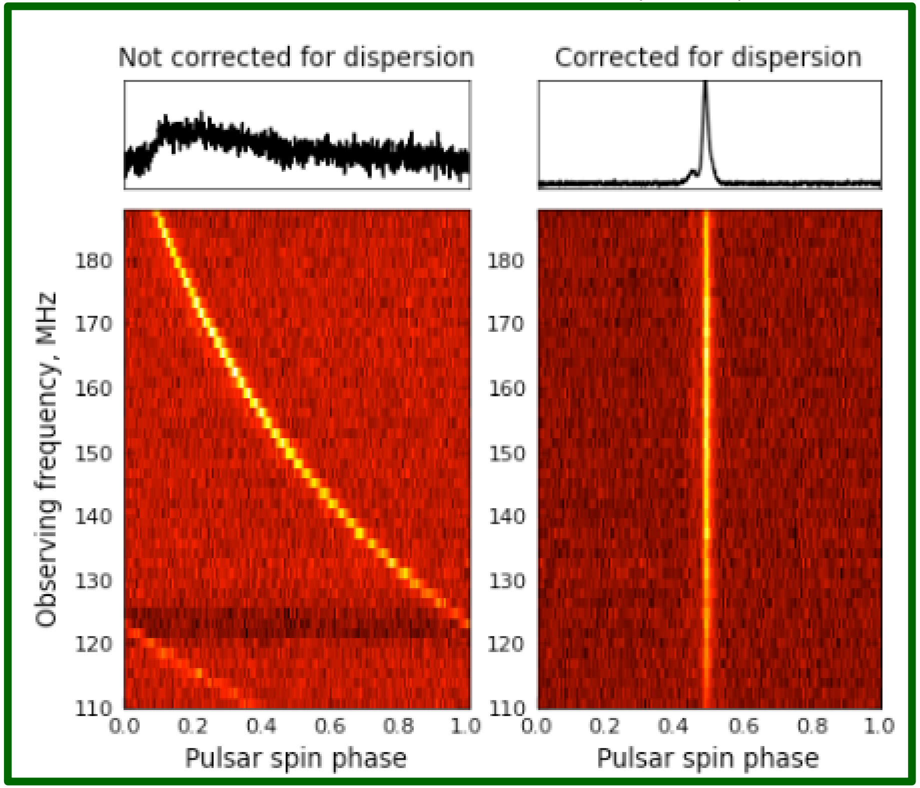


LOFAR HBA

LOFAR LBA

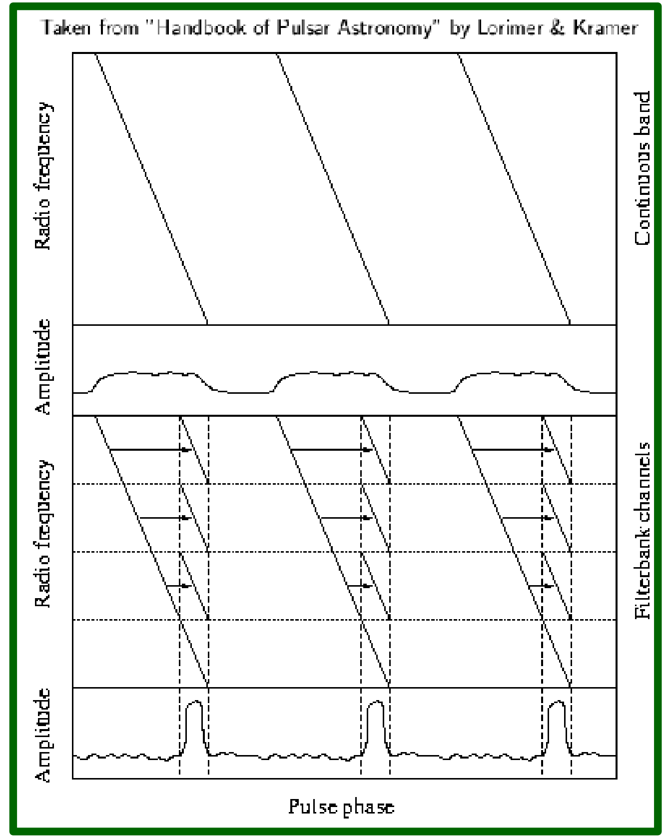
Dispersion

PSR B2021+51 **DM is off by only 3 pc/cc!**



Credit: Anya Bilous

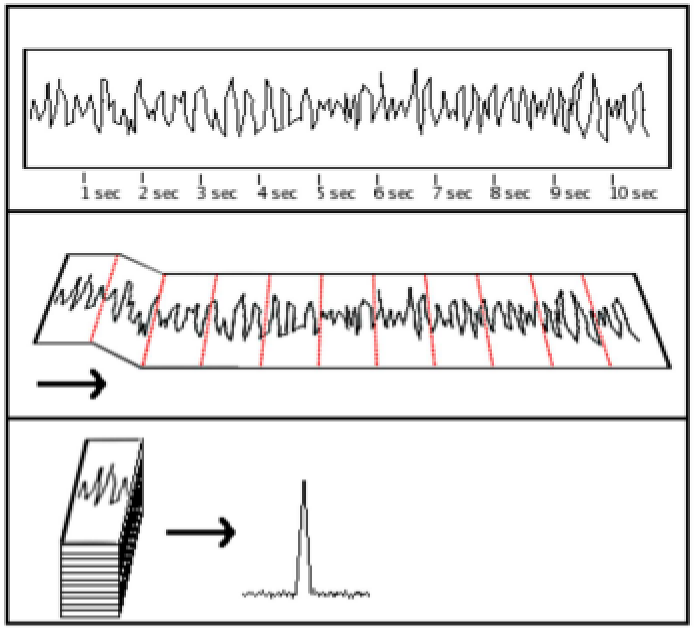
- DM [pc cm^{-3}] measures the integrated column density of free electrons along the line of sight
- Can be corrected using (in)coherent dedispersion



- Incoherent dedispersion – shifting channels in time
- Coherent dedispersion – requires complex-voltage data and is more computationally expensive

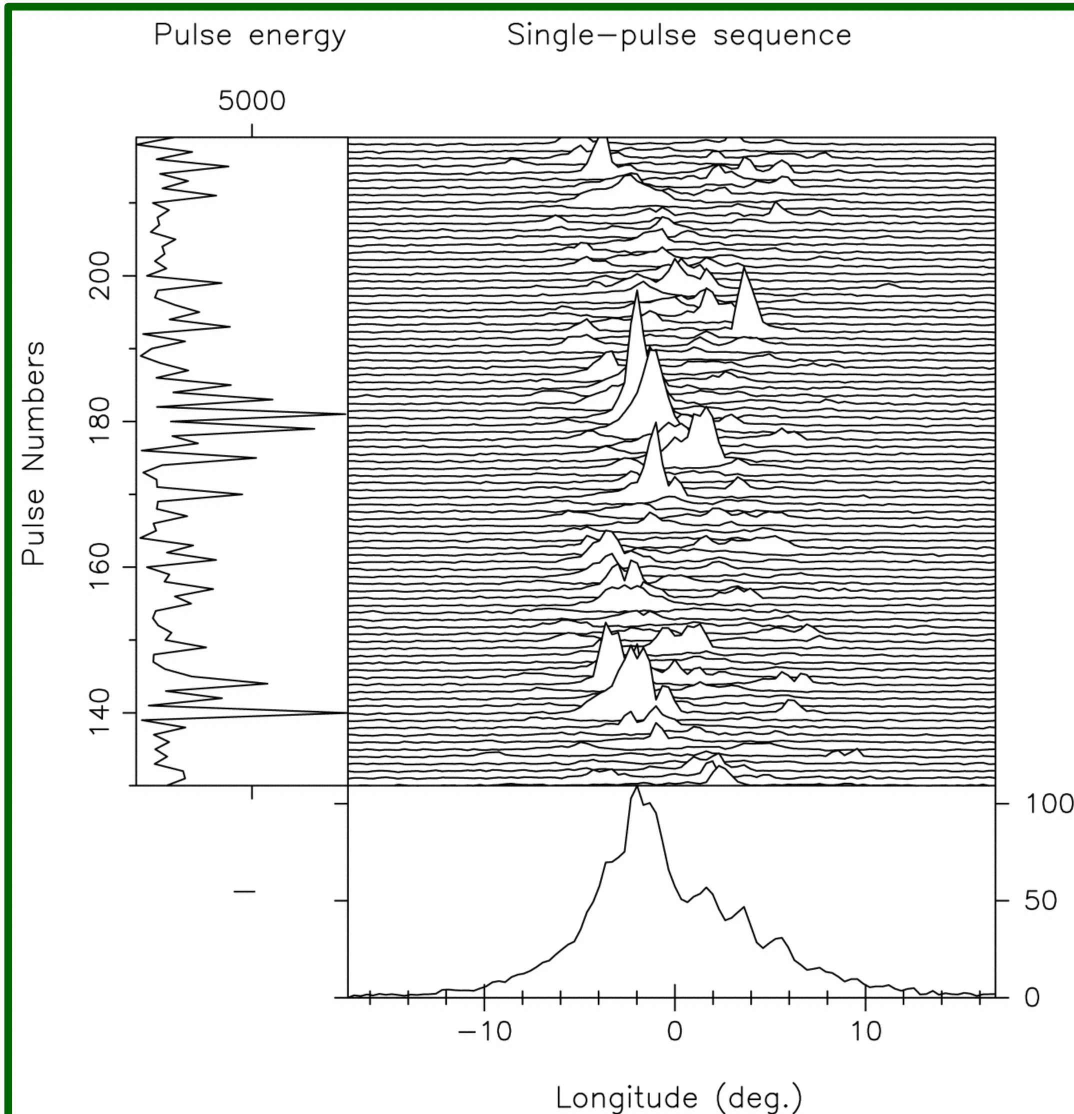
Folding

in a nutshell

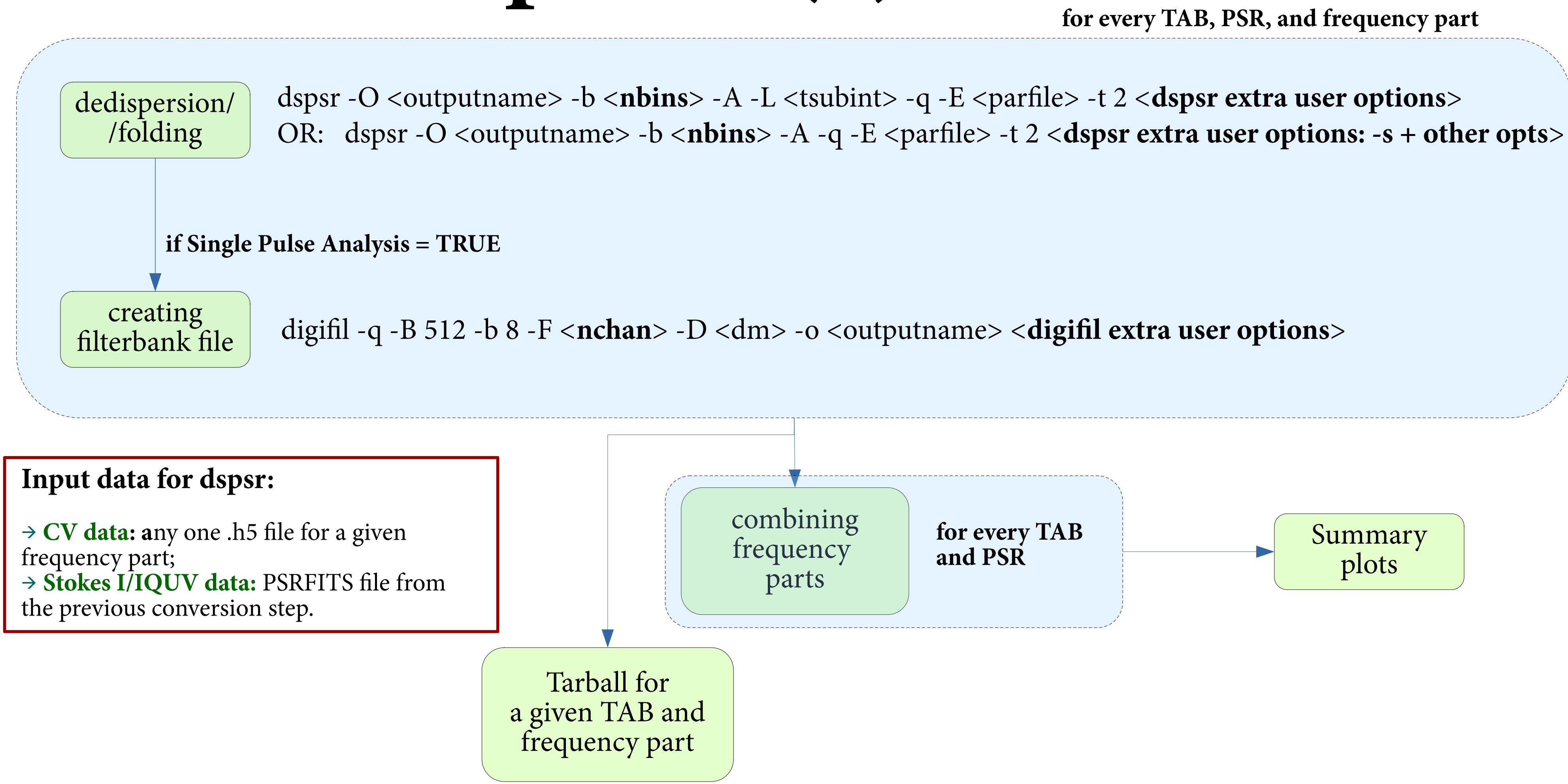


Deshpande & Rankin (1999)

PSR B0943+10



DSPSR Pipeline (1)



<**nbins**> - calculated automatically based on the sampling time and F0/P0 from the parfile.
Maximum possible <nbins>=1024

<**nchan**> - number of channels in a given frequency part. If number of channels = 1, then <nchan>=2

DSPSR Pipeline (2)

**combining frequency parts;
for every TAB, PSR**

add frequency parts together — psradd -R -m time -o <outputname>

↓ **if nchan/sub > 1**

zapping junk channels — paz -z <**list of channels to zap**> -m

↓ **if Skip RFI check = FALSE**

RFI excision — clean.py -F surgical -o <outputname>
OR: paz -r -e paz.ar **[if nsub * nchan > 512000]**

↓

Dedispersion — pam -D -m

↓

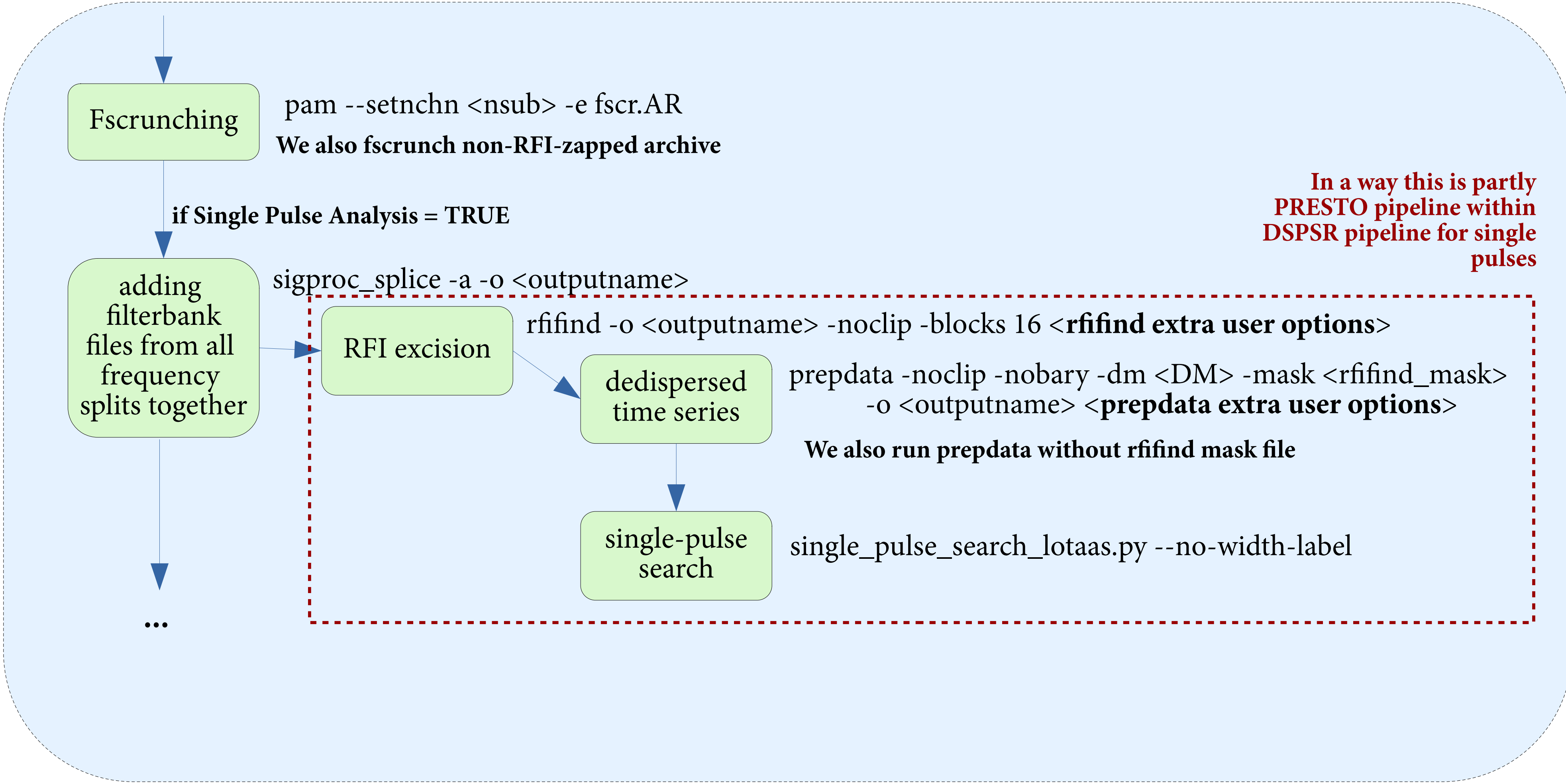
Fscrunching — pam --setnchn <nsub> -e fscr.AR
We also fscrunch non-RFI-zapped archive

↓

<**list of channels to zap**> - if there are 16 chan/sub, we need to zap every 16th channel, then list becomes «0 15 31 47...»
This is necessary, as when 2nd PPF is used, the first channel in each subband gets corrupted

DSPSR Pipeline (2, cont.)

**combining frequency parts;
for every TAB, PSR**



DSPSR Pipeline (3, cont.)

combining frequency parts;
for every TAB, PSR

...

make diagnostic plots

pav -SFTd -C **for CV data** *OR* pav -DFTpd -C **for non-CV data**
pav -GTpdf<**fscrunch_factor**> -C
pav -Yfpd -C
pav -j
rmfit -m -100,0,100 -D
rmfit -m 0,100,100 -D
and combine all them together in a single plng-file

optimizing Period/DM

pdmp -mc <Nsubs> -mb <min{128,Nbins}>

updating archive file with improved P/DM

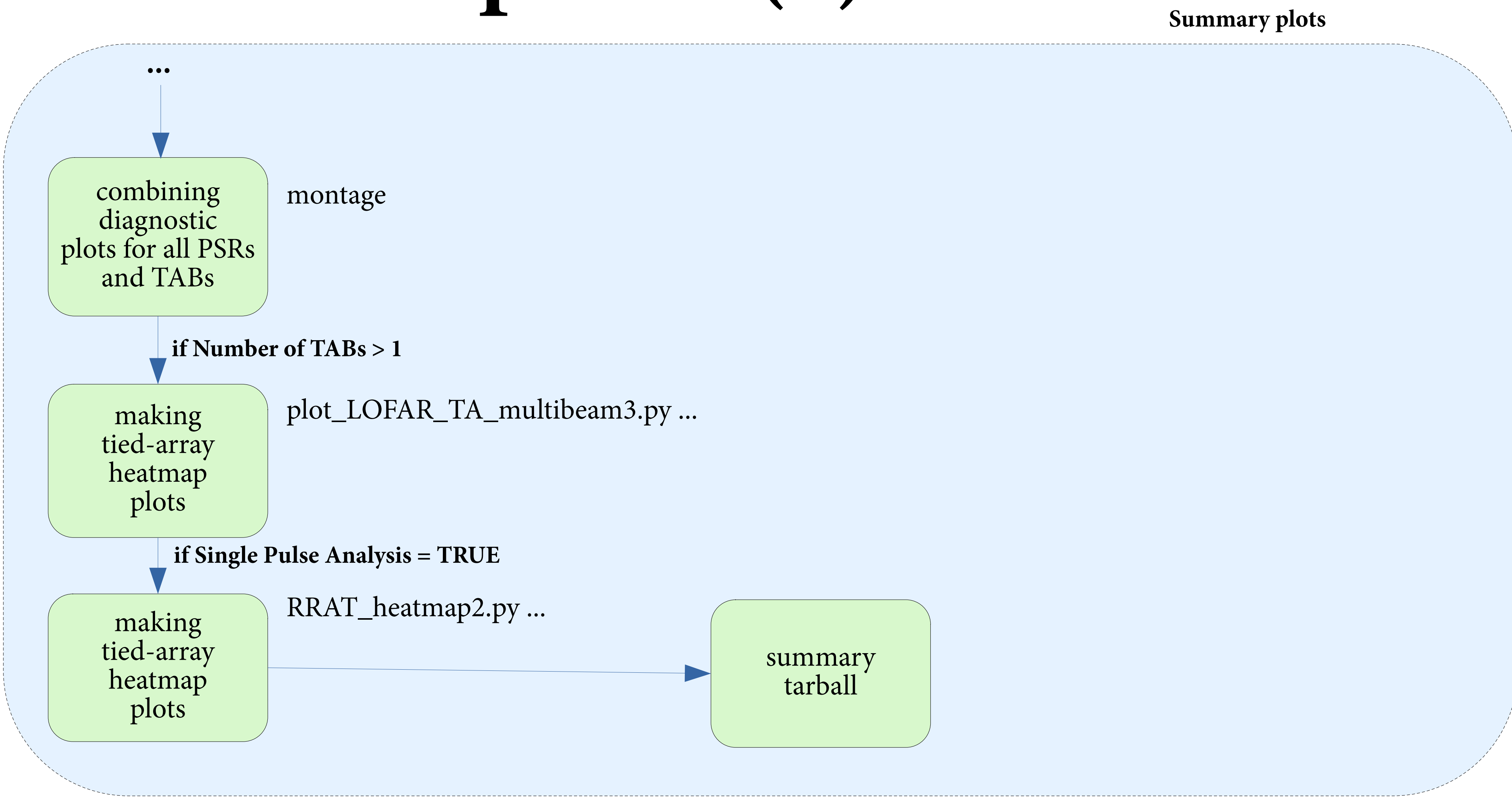
pam -e pdmp.AR -d <new DM> --period <new Period>

make diagnostic plots

same commands and diagnostic plots as for original archive file

<**fscrunch_factor**> = Nsubs / X, where X = highest common denominator of Nsubs between 1 and min{Nsubs, 63}

DSPSR Pipeline (4)



PRESTO Pipeline (1)

for every TAB, and frequency part

RFI excision, zapping junk channels

rfifind -o <outputname> -psrfits -noclip -blocks 16 -zapchan 0:X:Y <**rfifind extra user options**>
X = Nchan-1
Y = Nchan/sub

↓ **if Skip dynamic average = FALSE**

making diagnostic dynamic spectrum

subdyn.py -psrfits -saveonly -n <samples_to_average> ...

↓

for every PSR

Dedispersion/ /Folding

Prepfold -noscales -nooffsets -noxwin -psr <psrname> -par <parfile> -n <**nbins**> -nsub <**nsubs**>
-fine -nopdsearch -mask <rfifind_mask> -o <outputname> <**prepfold extra user options**>

We also run prepfold without rfifind mask file

↓

make diagnostic plots

convert
montage → profile thumbnails

↓

...

<**nbins**> - calculated automatically based on the sampling time and F0/P0 from the parfile.
Maximum possible <nbins>=1024

<**nsubs**> - if Nchan > 512, nsubs = 512. Otherwise, nsubs = Nchan

PRESTO Pipeline (2, cont.)

for every TAB, and frequency part

...

if Single Pulse Analysis = TRUE

for every PSR

dedispersed time series

prepdata -noscales -nooffsets -noclip -nobary -dm <DM> -mask <rfifind_mask>
-o <outputname> <**prepdata extra user options**>

We also run prepdata without rfifind mask file

single-pulse search

single_pulse_search_lotaas.py --no-width-label

<**nsubs**> – greatest common denominator of Nchan between 1 and 1024
<**lodm**> – DM - 0.5*dmstep*numdms. If lodm<=0, then lodm=0.01

if RRATs analysis = TRUE

dedispersed time series for a range of DMs

prepdata -noscales -nooffsets -noclip -nobary -dm 0.0 -mask <rfifind_mask> -o <outputname>
<**prepdata extra user options**>

We also run prepdata for DM=0.0 without rfifind mask file

prepsubband -noscales -nooffsets -noclip -nobary -nsub <**nsubs**> -lodm <**lodm**> -dmstep 0.01
-numdms 1000 -mask <rfifind_mask> -o <outputname> <**prepsubband extra user options**>

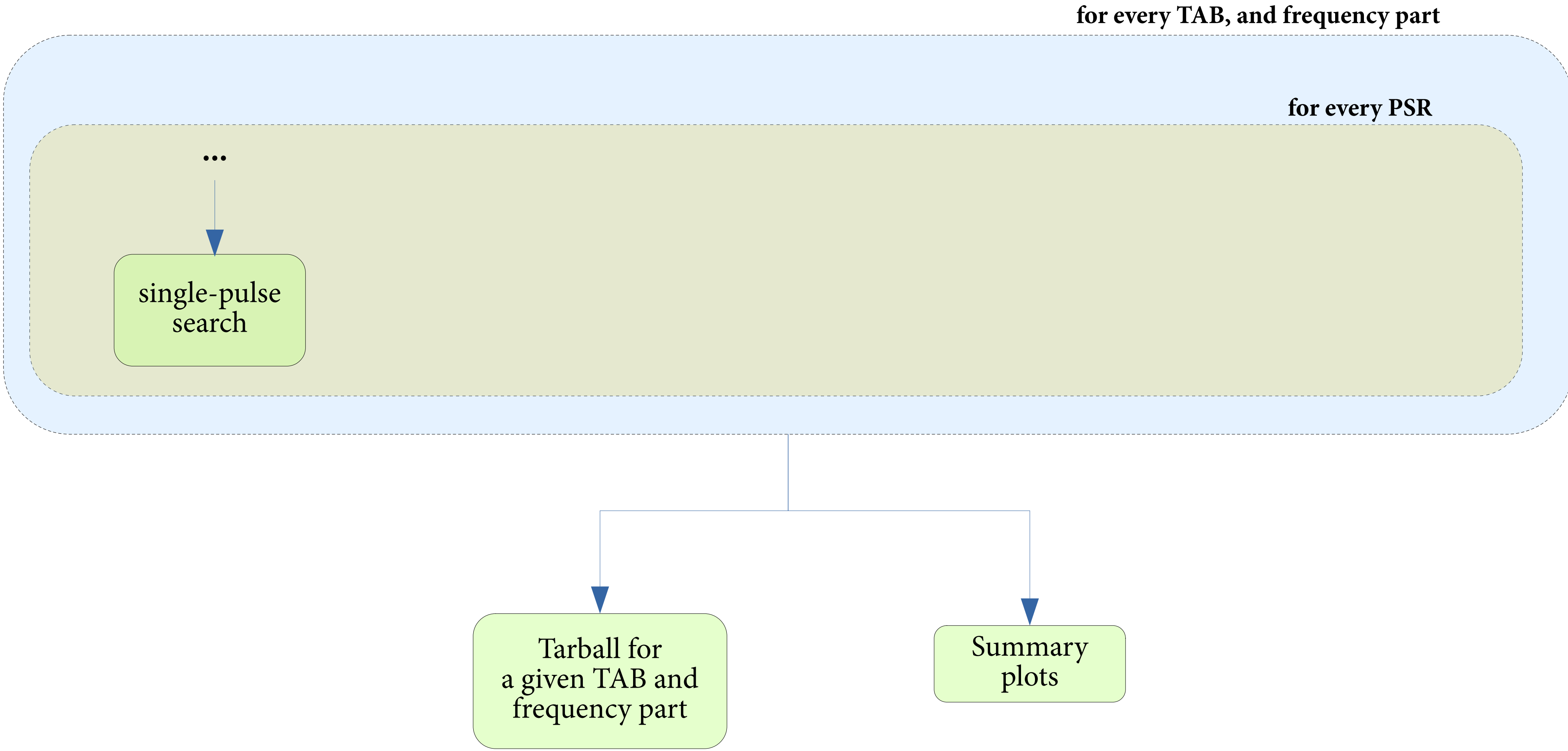
We also run prepsubband without rfifind mask file

single-pulse search

single_pulse_search_lotaas.py -p -g *.dat
single_pulse_search_lotaas.py -t 5.5 —no-width-label -g *.singlepulse

Also running similar command but excluding DM=0

PRESTO Pipeline (3)



PRESTO Pipeline (4)