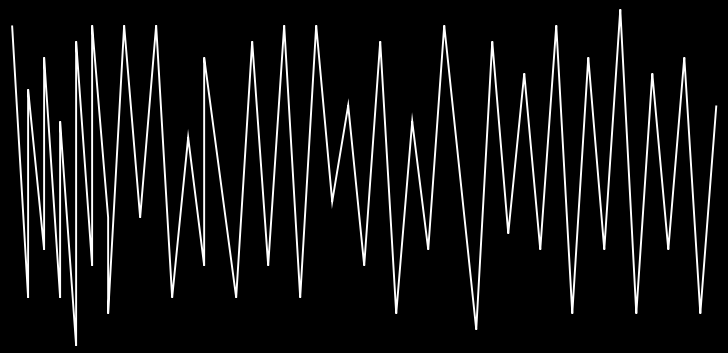
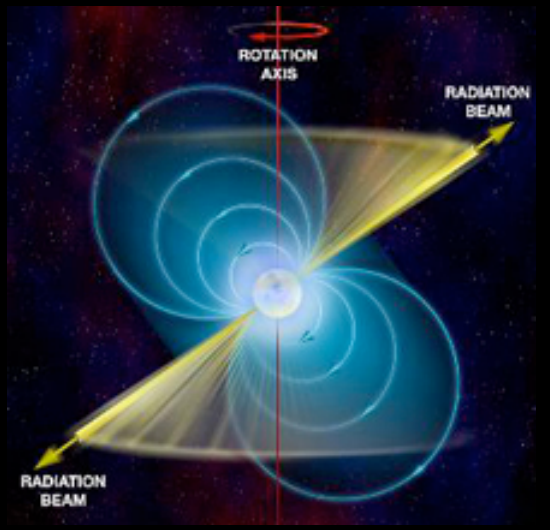
Analyzing LOFAR Beam-Formed Data

Jason Hessels

(ASTRON/UvA)





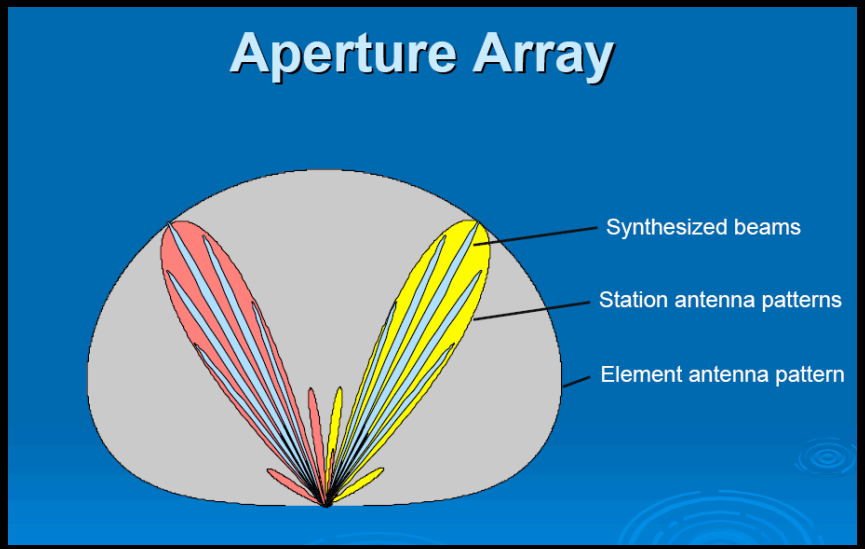
LOFAR Data Processing School

ASTRON - Feb. 11, 2009

Outline

- LOFAR raw beam-formed data: what it is, format.
- Accesssing these data: basic access/inspection of these data for system checks and to build analysis tools.
- Existing software: existing (scientific) software packages for analysis.
- New software: where we want to go; what we need to do.

Some Terms



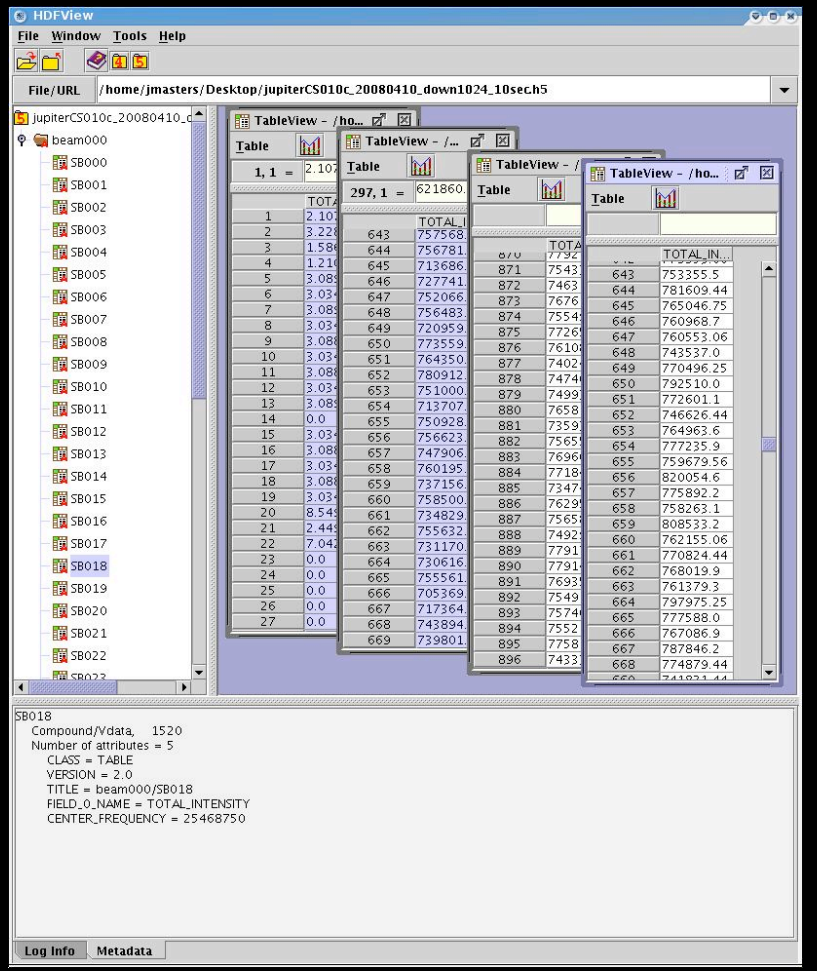
- Beam-formed data: non-imaging data, generally with high time and frequency resolution. Also termed "pulsar" data or "timeseries" data.
- (Dedispersed) timeseries: the signal (e.g. Stokes I) as a function of time at a given central frequency. This can be the combination of many spectral channels in which a correction for dispersive delay has also been applied.
- Filterbank/subband data: a collection of timeseries at a range of discrete frequency bands.
- PRESTO/PSRCHIVE: two large, previously existing suites of pulsar search and reduction routines.

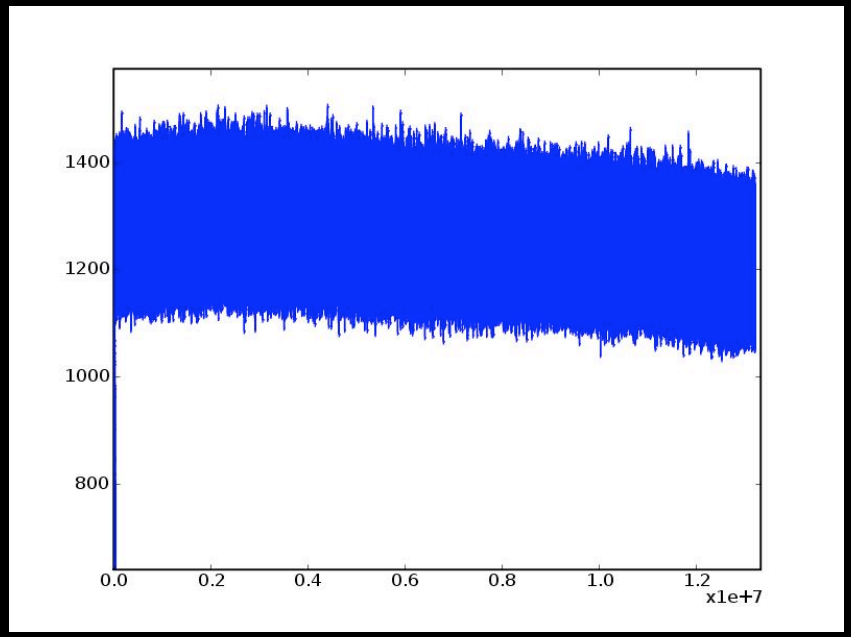
Raw data: Format

- HDF5 (Hierarchical Data Format): flexible file format for raw beam-formed (and other) LOFAR data.
- Keeps track of data (and derived data products?) across file systems. Important for large (>1TB) data sets.
- Extensive LOFAR header structure defined (easily extended and modified in future... input welcome!).

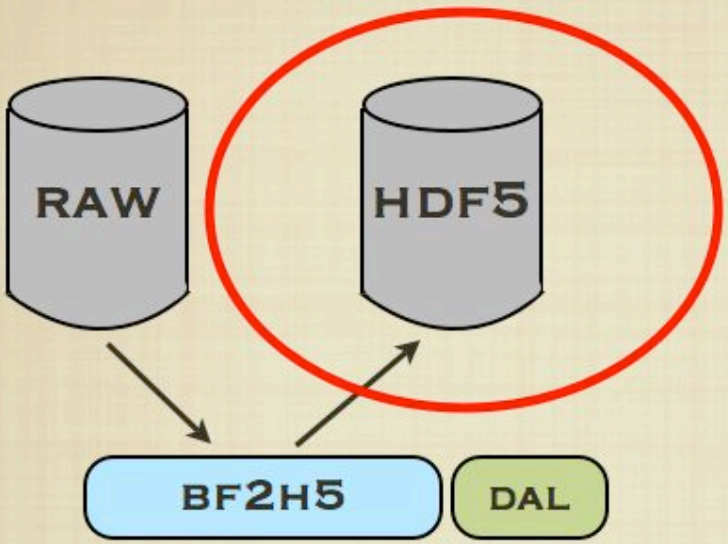
Raw data: Visualization

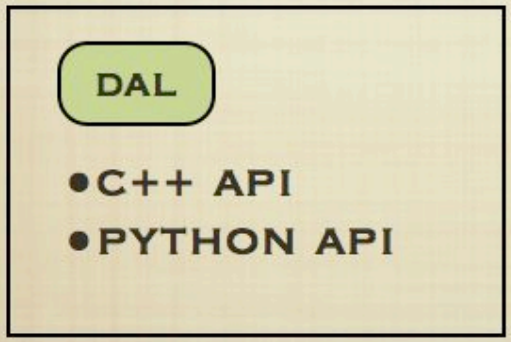
- Existing tools for visualizing and reading HDF5 (e.g. VISIVO).

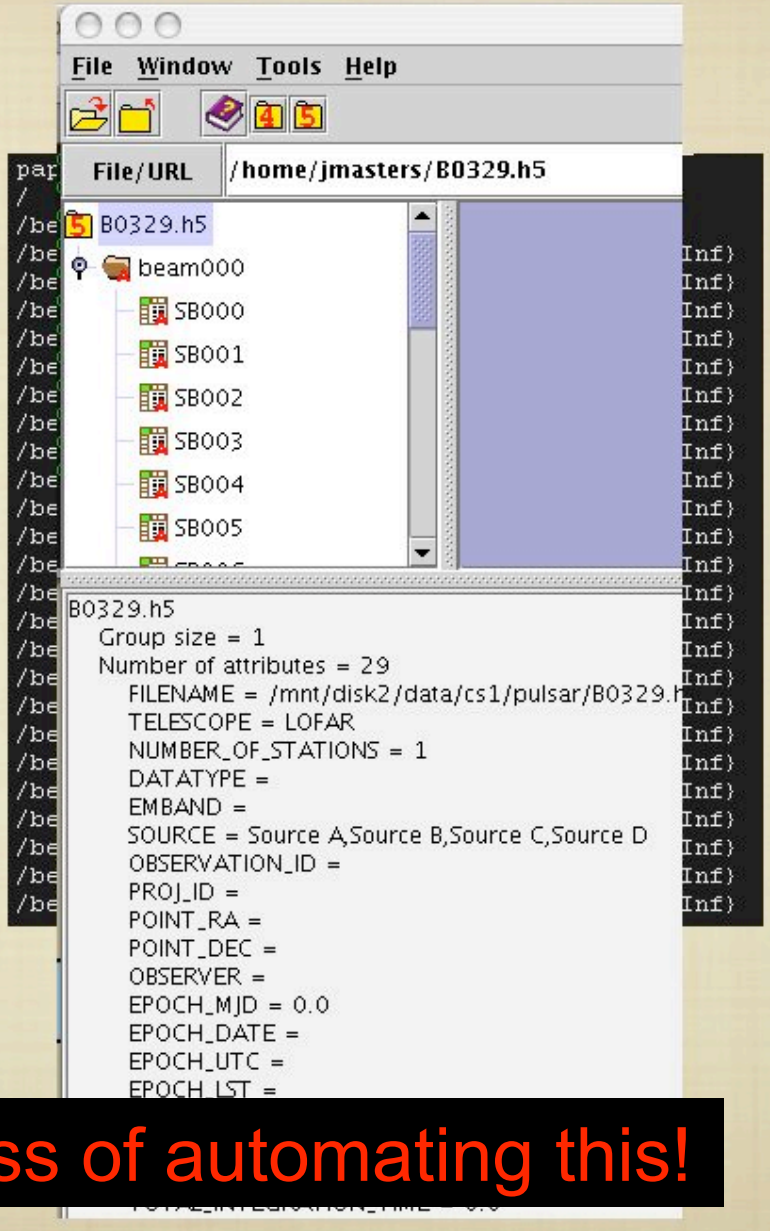




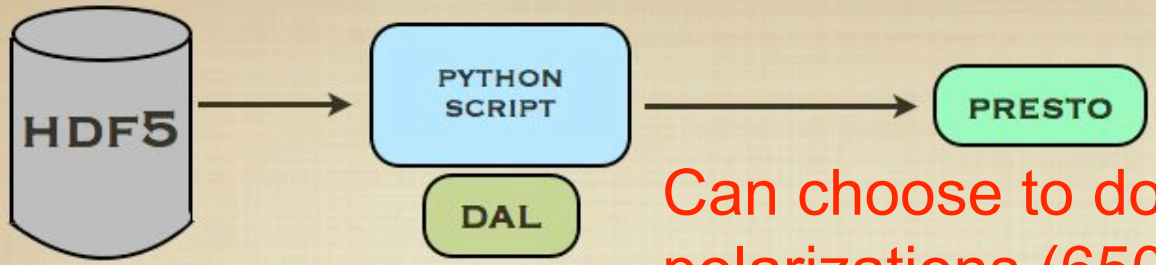
Current Status





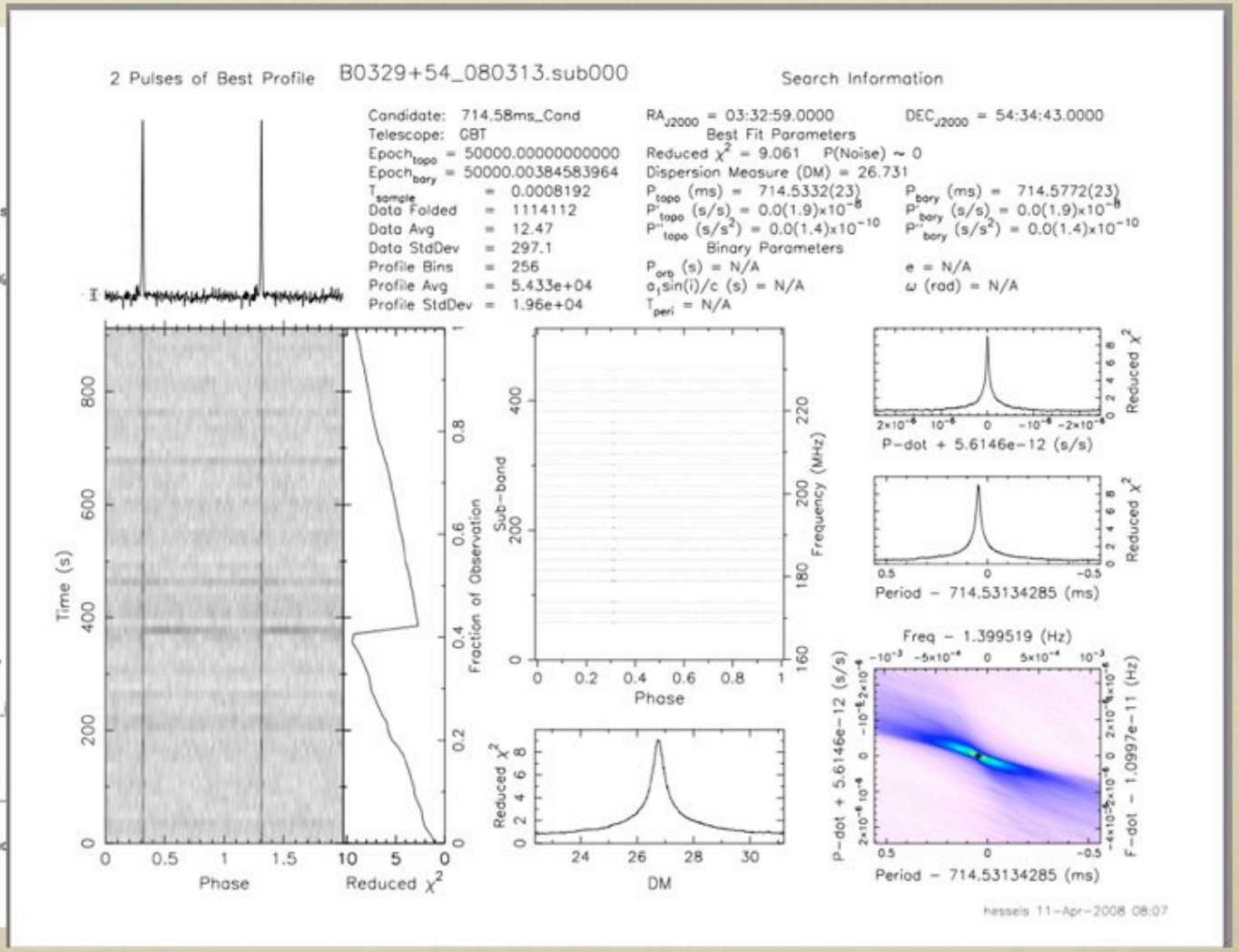


We're in the process of automating this!



Can choose to downsample and sum polarizations (650GB --> 3GB)

```
#! /usr/bin/env python

from pydal import *
from scipy import *
import sys
from pylab import *

def downsample(vector, factor):
  """
  downsample(vector, factor):
    Downsample (i.e. co-add consecutive numbers) a short s
      of a vector by an integer factor.
  """
  if (len(vector) % factor):
    print "Lenght of 'vector' is not divisible by 'factor'=%d!" %
    return 0
  newvector = reshape(vector, (len(vector)/factor, factor))
  return add.reduce(newvector, 1)

# The BeamFormed object represents the file.
#  The parameter is the name of the beam-formed file.
if (len(sys.argv) > 1):
 file = BeamFormed(sys.argv[1])
else:
 print "Please provide a beam-formed hdf5 file as input."
 sys.exit(1)

# get beam 0
beam = file.getBeam(0)

n_samples = 512 * 128

for subband in range(24):
    start = 0
    downsampled_data = array([],'f')
    for count in range(2200):
        data = beam.getSubbandData_XY( subband, start,
        xx_intensity = abs(data[0])
        yy_intensity = abs(data[1])
        tot_intensity = sqrt(xx_intensity*xx_intensity + yy_

        data = downsample( tot_intensity, 128 )
        start += n_samples

        downsampled_data = concatenate( (downsampled_


    tmpfile = "B0329+54_080313_subband%02d" % subbanc
    fileobj = open(tmpfile, mode='wb')
    fileobj.write(downsampled_data)
    fileobj.close()
    print "Wrote ",tmpfile
```



Data Access:
Data Access Layer (DAL)

- Library of routines to read LOFAR HDF5 beam-formed data (C++).
- Handy python module called “pydal” (python implementation of C++ code).
- Can use DAL to e.g. read headers, build basic tools (bandpass, timeseries) and reduction steps, link with other software made very easy.

Info: Joe Masters, Lars Bähren

HDF5 QUICK LOOK

```
$ python
> from pydal import *
> file = BeamFormed("myfile.h5")
> file.summary()
> beam = file.getBeam(0)
> data = beam.getSubbandData_XY( 5, 0, 100 )
> file.number_of_beams()
> file.point_ra()
> file.number_of_samples()
```

Excellent for visualizing data, building analysis scripts, and linking to existing reduction packages.

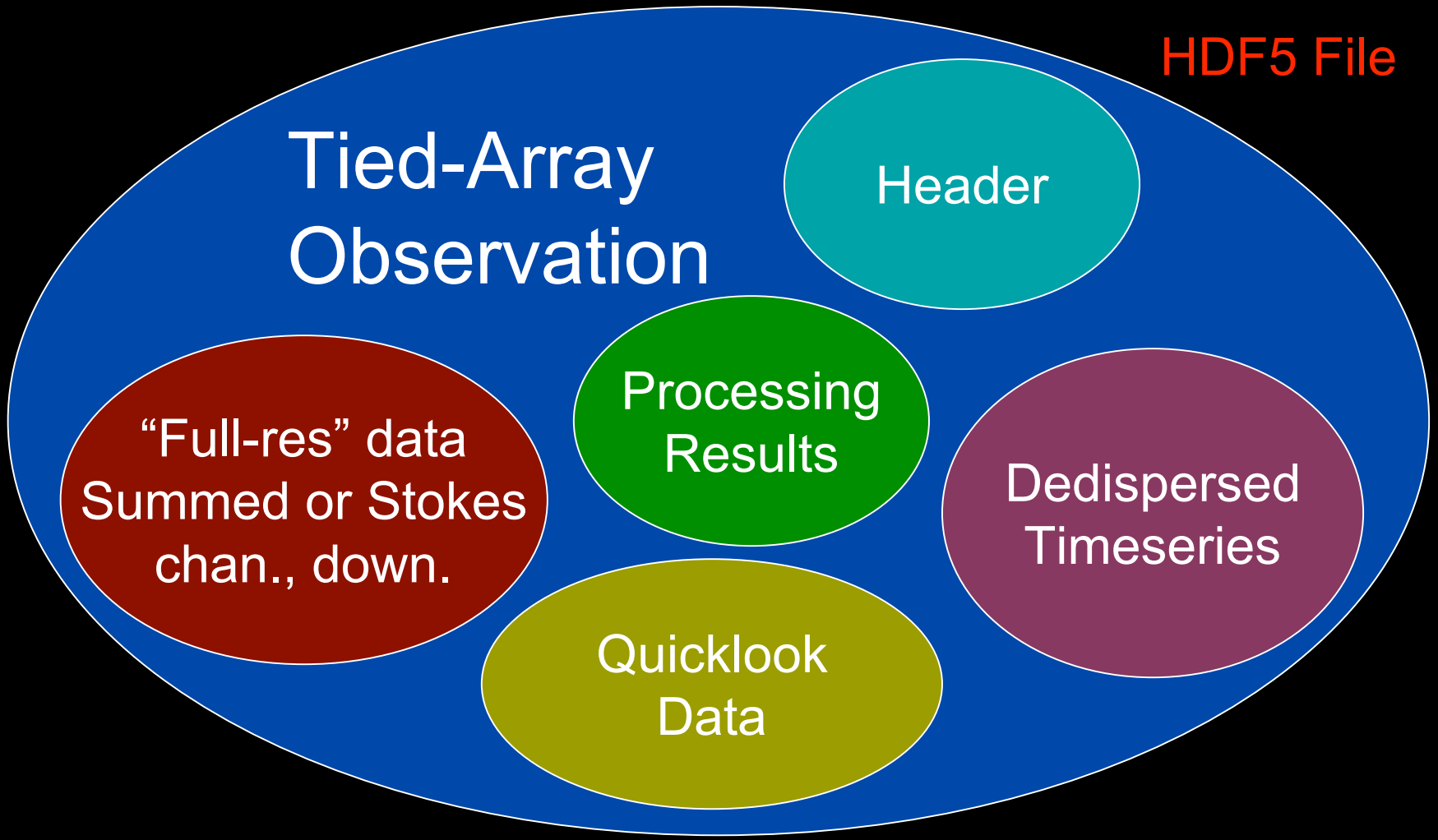
Raw data: Initial Reduction and Customization of Data Products

The following is at least partially implemented (there is already a very crude pulsar pipeline... or perhaps “pulsar conveyer belt”)

- Output total power (e.g. pulsar search) or full Stokes.
- Allow downsampling of powers.
- Create ~16-256 channels per subband at the expense of the native time resolution.
- Directly provide (coherently) dedispersed timeseries (prohibitive in realtime for pulsar search) and/or folded profiles.
- Provide multiple tied-array beams.

Info: Pulsar Working Group, Jan-David Mol, John Romein

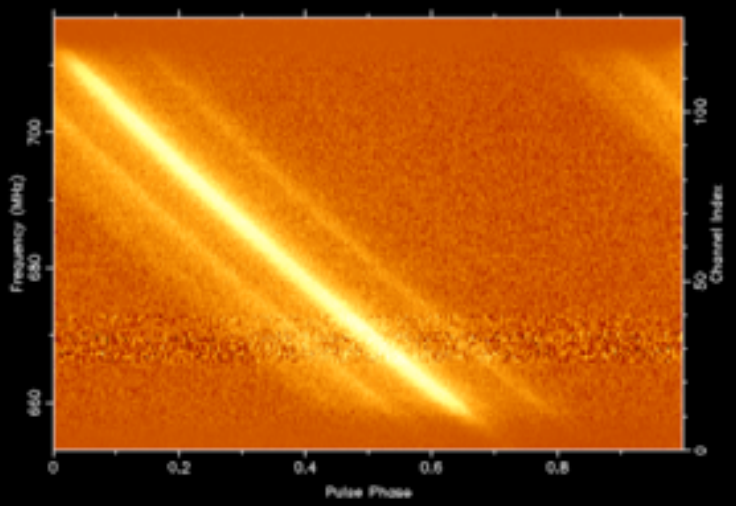
Raw data: Customizing, Initial Reduction



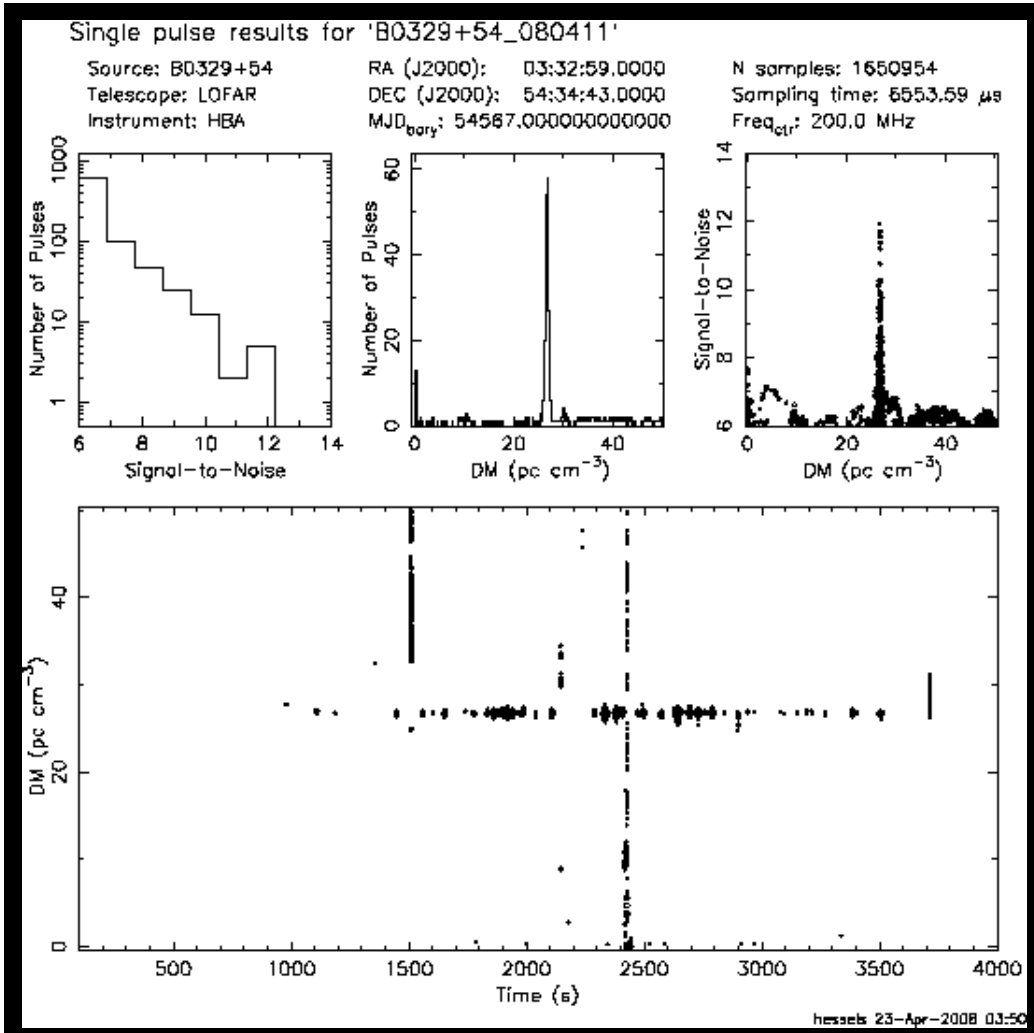
Linking Software

We already have software like PRESTO and PSRCHIVE to e.g.:

- Excise RFI.
- Create dynamic spectra.
- Search for bursts.
- Coherently dedisperse.
- Fold data.
- Measure polarization.
- Search for periodicities.



We use the DAL to read the T-A data into these...
Scientific reduction of the data is a great way to check the system.

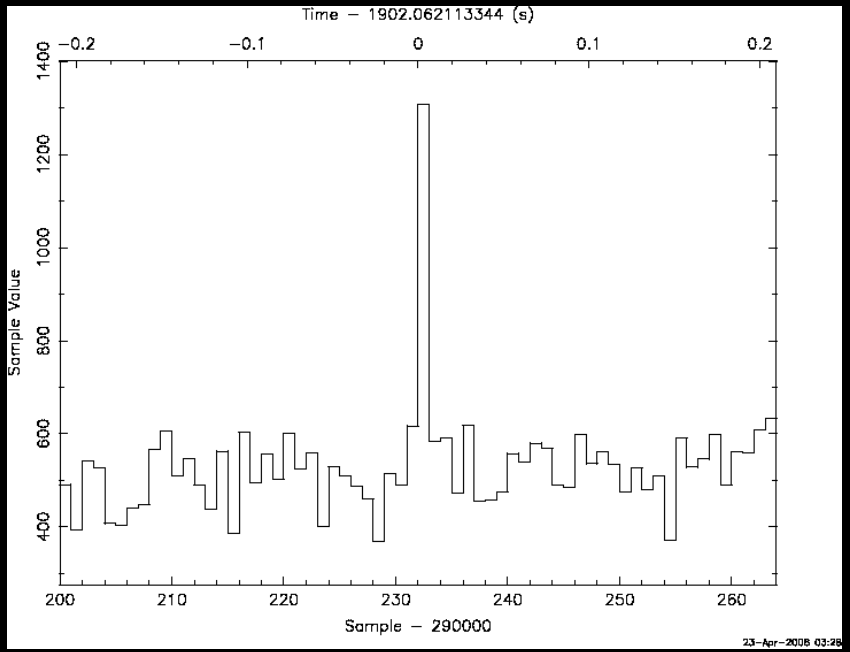


PSR B0329+54

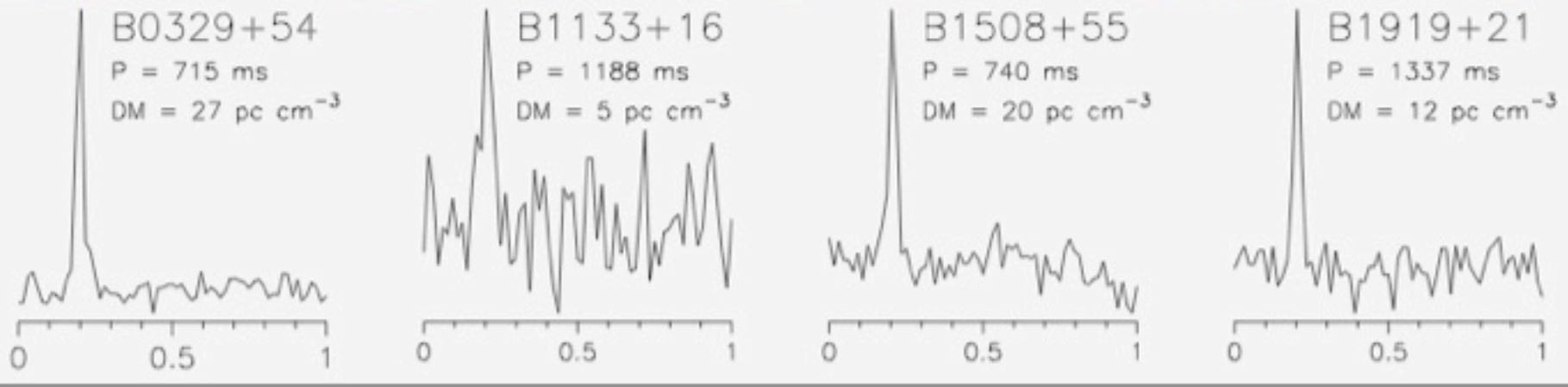
Period = 714.5ms

Bright single pulses detected with LOFAR HBA (160-240MHz)

One of the brightest pulses →

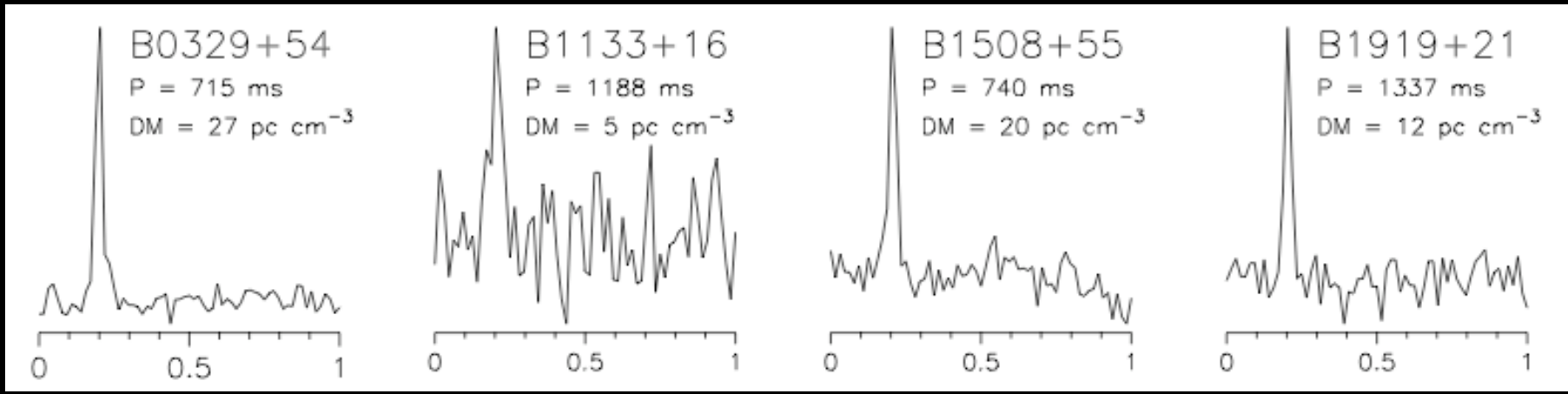


LOFAR Pulsar "Busy Week"



Results of Pulsar “Busy Week”

Pulsars other than B0329+54 detected for the first time with LOFAR



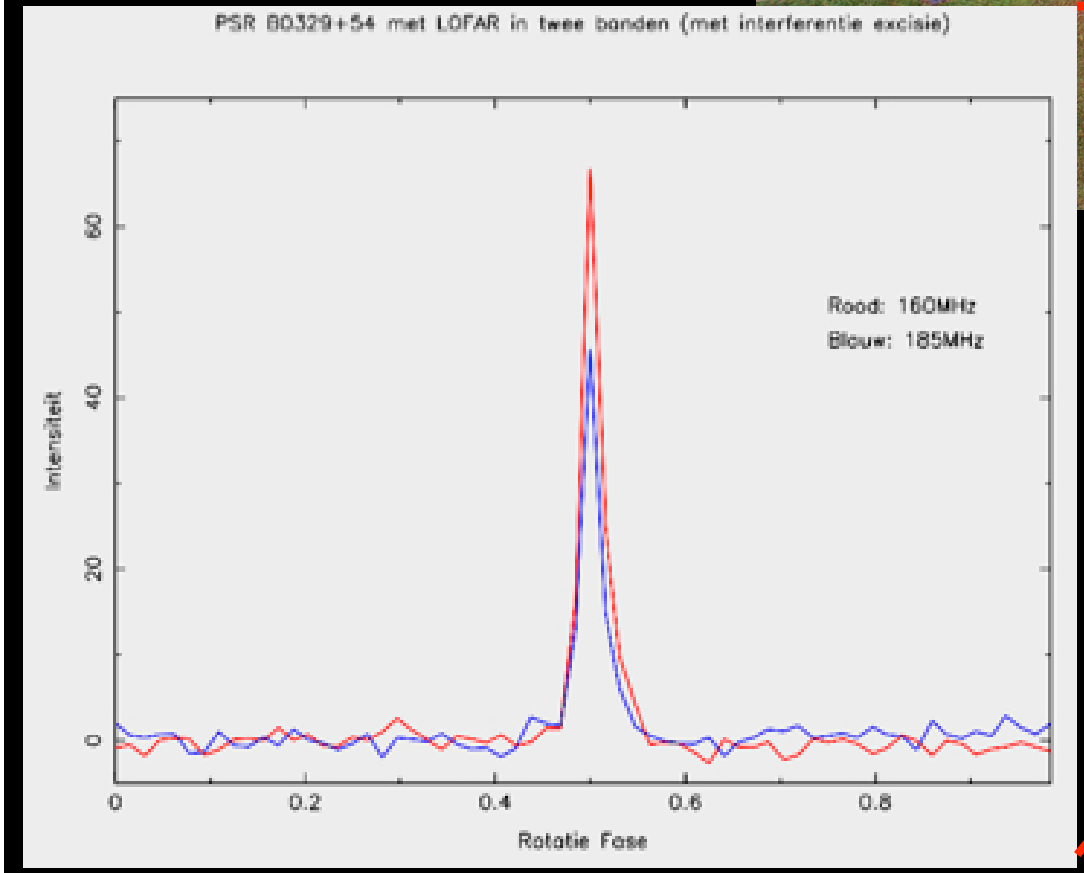
All these with only 1 HBA tile (1-2 hr integrations)!

Excellent test sources for commissioning

Nice spread in RA so tests can be done at any time

Note with 4 tiles added we can get the same detection in ~1/16 the time (i.e. 5-10 minutes)!

From humble beginnings...

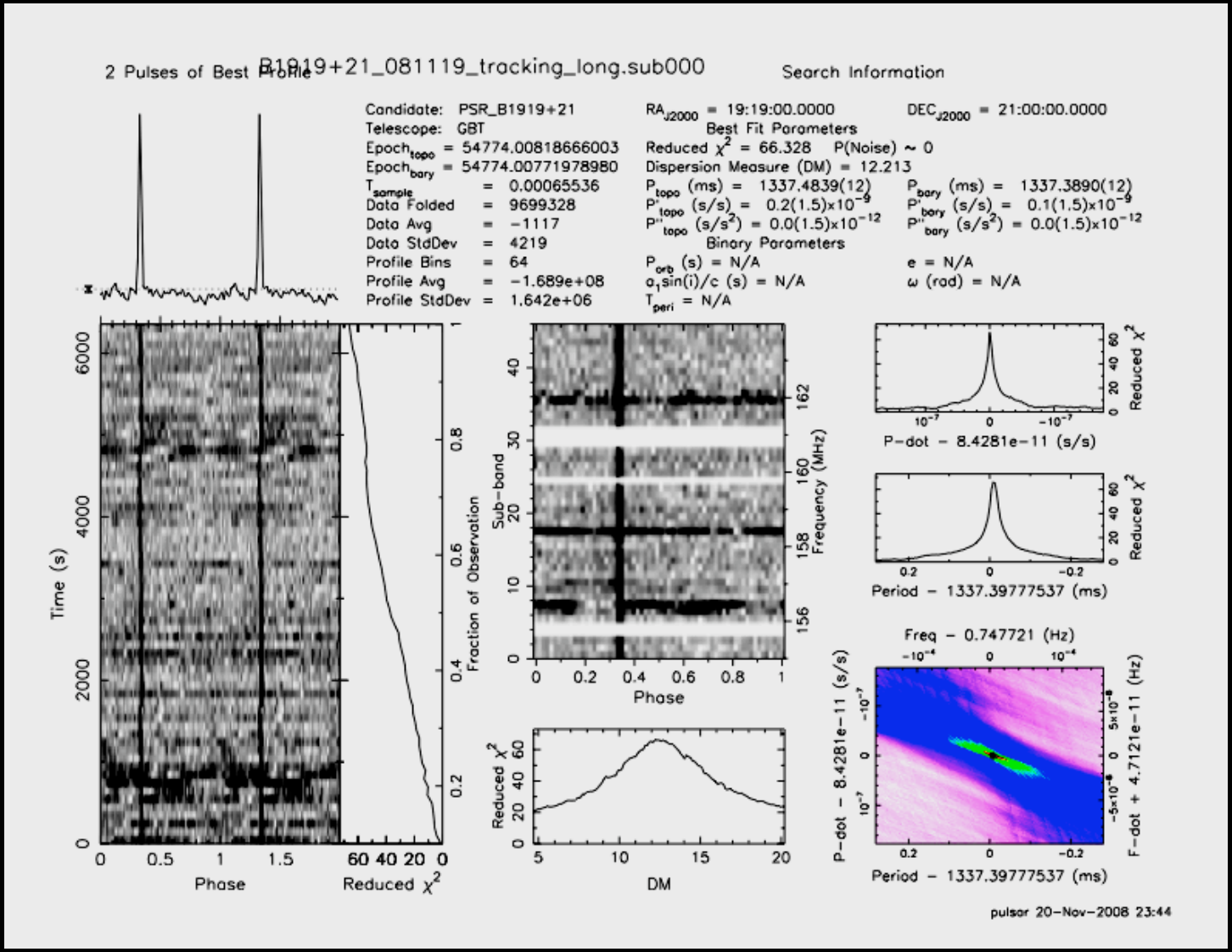


...can use same software for full station, two stations becomes trickier...

Results of Pulsar "Busy Week"

Successful 2-hr track of B1919+21 with 4 tiles added together

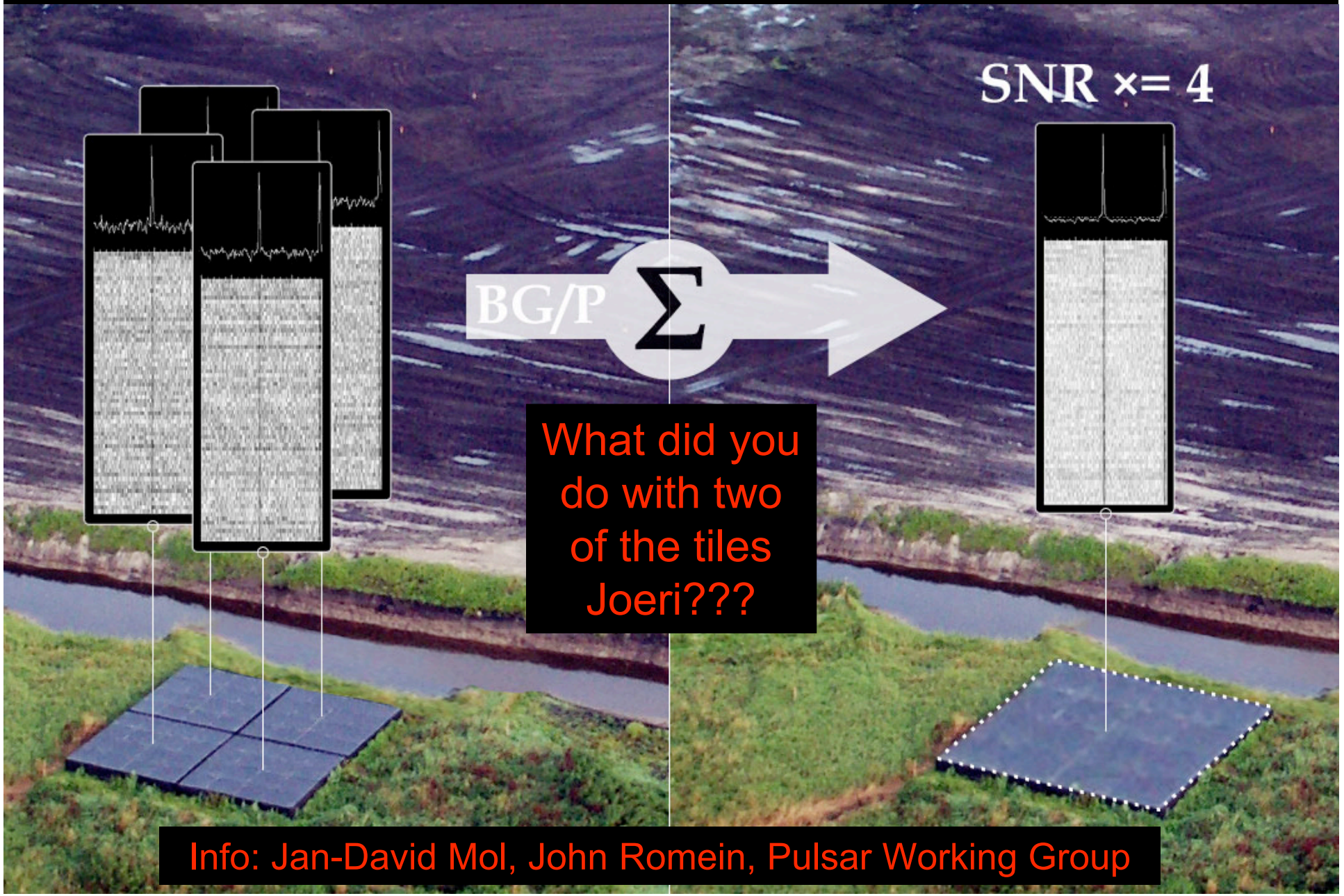
Note that RFI masking could have been done much better here (my laziness...)



Tracking problem solved!

Ultimately, single pulses from B1919+21 will be significantly brighter than this!!!

Tied-Array Mode



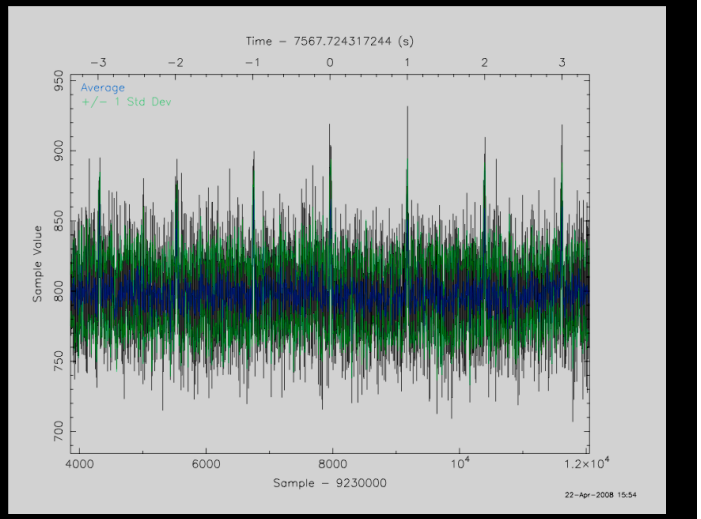
Info: Jan-David Mol, John Romein, Pulsar Working Group

Coming Soon

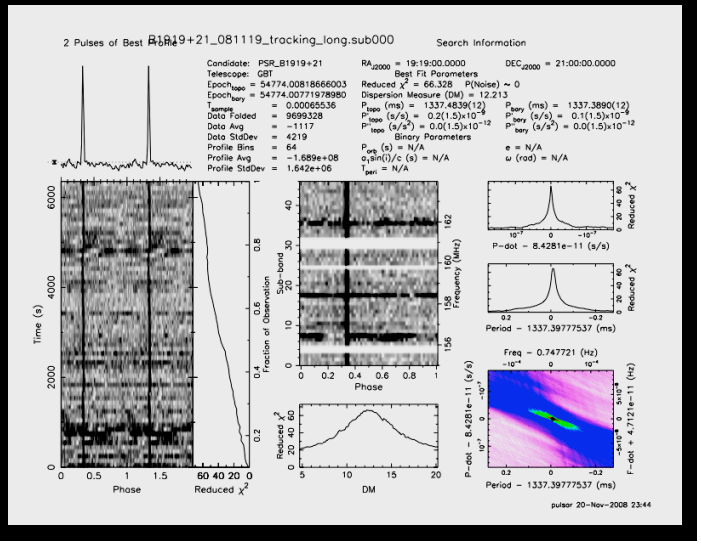
- Another “Pulsar Busy Week”: 1) implement online data-taking straight to HDF5 or PRESTO format, 2) further test tied-array mode.
- Will likely need to extend some of the existing analysis packages like PRESTO and PSRCHIVE to optimize them for LOFAR (approximations that aren’t valid at low frequency?) and write new software.
- Keep the DAL in mind when developing things. Don’t reinvent the wheel unless it is a much better wheel!

This afternoon's exercises:

Exercise 1: Interactively investigate time series and their Fourier transforms

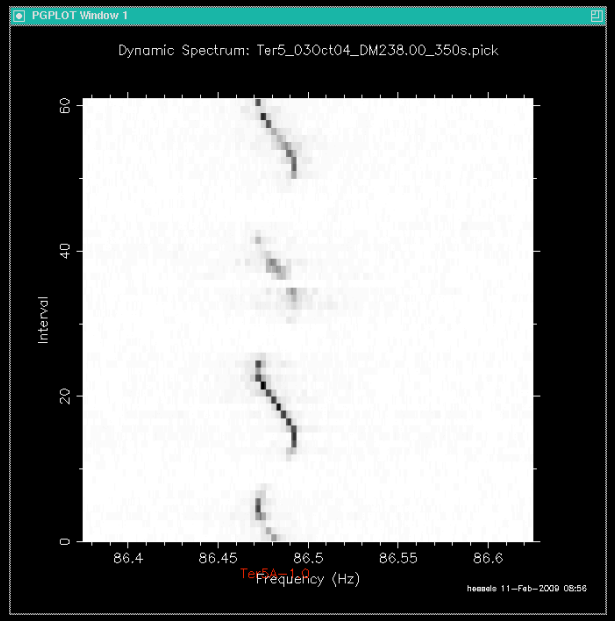


Exercise 2: Fold and dedisperse data to detect a pulsar signal.



This afternoon's exercises:

Exercise 3: Create and investigate dynamic spectra.



Remember: if pulsars bore you, there are still *many* other applications for beam-formed data and such data also gives sometimes unique tests/diagnostics of the system as a whole (e.g. clocks).