Beam-formed Data with LOFAR

Richard Fallows, ASTRON
Beam-formed Data – Main Uses

- Pulsars – already covered
- Single-station science
- Dynamic spectra:
  - Solar radio bursts
  - Planetary emission
  - Scintillation
- “Imaging” diffuse objects:
  - Raster-style imaging
  - Useful for Sun and other wide objects
A Reminder of the Beam Types

From sub-array pointings to tied-array beams, LOFAR has some complex terminology.

Herewith a brief summary...
HBA Tile Beam

- Each HBA tile consists of a 4x4 grid of crossed-dipoles, controlled by an analogue beam-former
  - Can be “pointed” in only one direction at a time!
  - Baseline is 5m:
    - Beam diameter ranges from ~43° at 110MHz to 20° at 240MHz
- Central system: Pointing direction of tile specified by definition of first pointing.
- Single station use:
  - Define pointing how you wish
  - Sub-array station to point different tiles in different directions (with corresponding loss in sensitivity)
“Sub-Array Pointing” - Station Beam

- Dipoles/Tiles combined via station beam-former
  - Forms “beamlets” where each beamlet is a pointing for a single sub-band
  - Up to 488 beamlets can be formed (8-bit mode)
  - Beamlets can be pointed in any direction using any sub-band:
    - Use all beamlets to point in the same direction with full bandwidth
    - Have several/many different pointings with a corresponding trade-off in bandwidth.

- Any single pointing direction covering a set of sub-bands is known as a sub-array pointing, or station beam.

- If HBA is used, pointing only makes sense within the area covered by the tile beam.
• Central correlator beam-former combines data from a set of stations.
• Form a single incoherent beam for each sub-array pointing:
  – Use any combination of stations (core/remote/international)
  – Combination applies only geometrical delays between stations, no “phasing up”
Coherent "Tied-Array" Beam (TAB)

- Central correlator beam-former combines data coherently from a set of stations
- Currently restricted to core stations only as these are on the single clock
- Many tied-array beams can be formed:
  - Each has full bandwidth of sub-array pointing
  - Narrow beams: 1.2° for 10MHz to 0.05° for 240MHz
  - Pointing only makes sense within area covered by sub-array pointing
TAB Example: Crab pulsar scintillating through the solar wind
TAB Example: solar radio burst seen through a thunderstorm – superterp stations

20120705, 14:37:00-14:37:30 UT, Superterp
TAB Example: multiple TABs mapping the Sun

- Can form concentric 'rings' of beams on and around an object.
- Try it to see if solar radio bursts can be mapped with this technique.
- Advantage over traditional imaging of very high time and frequency resolution and ready access to a dynamic spectrum for each beam.
TAB Example: multiple TABs mapping the Sun
Recording Single Station Data

- “Fly's Eye” mode
- Correlator outputs sub-array pointing data from each station individually
- Possible applications:
  - Off-line correlation of raw complex voltages (VLBI for example)
  - Recording high time-resolution time series' for later correlation (interplanetary scintillation for example)
  - Future: Parallel observations, different stations pointing in different directions, or observing different frequency ranges.
Simultaneous LBA/HBA Observation of the Sun
The Beam-formed Data Format – Reading data using Python
Beam-formed Data Files

- Data are accessible via HDF5
- Consist of two files:
  - *.h5: HDF5 file containing the meta-data plus a data array linked to:
  - *.raw: The data themselves in a binary format.
- Data can be read using any HDF5 library routines or file reader (e.g., hdfview) by opening only the h5 file.
  - But both files must be stored in the same place and you must be reading them from that place
Beam-formed Data Files

File name structure:

<ObsID>_SAPxxx_BEAMxxx_Sx_Pxxx.h5/raw

- ObsID is the observation ID, Lxxxxxx
- SAPxxx is a sequential sub-array pointing number, e.g., SAP000
- BEAMxxx is a sequential beam number, e.g., BEAM000
- Sx refers to Stokes parameter: S0-S3 for Stokes I,Q,U,V respectively
- Pxxx is a sequential part number for cases where data for various sub-band blocks are split over several files.

- If sub-bands are all in the same files, only P000 is used.
Beam-formed Data Files

For example, Stokes Q data from observation with ID number L246143 for the 10th labelled beam number of the second station beam will have the file names:

L246143_SAP001_BEAM009_S1_P000.h5
L246143_SAP001_BEAM009_S1_P000.raw

If the observation contains 400 sub-bands recorded as 20 sub-bands per file, part numbers would be P000 to P019.
Beam-formed Data Files

- Note on BEAM numbers:
  - The order is important:
    - Coherent tied-array beams are numbered first
    - Specified pointing directions numbered ahead of those calculated from tied-array `rings'.
    - Incoherent Stokes beams are numbered next
    - Fly's Eye beams numbered last.
  - No numbers are reserved for particular cases, so all available beams will be numbered sequentially and the experiment setup needs to be known to determine which numbers refer to which coherent, incoherent or fly's eye beam.
H5 File Structure

- HDF5 file structure is a hierarchical structure of “folders”/“directories” each containing meta-data and/or data:
  - /SUB_ARRAY_POINTING_xxx
  - /SUB_ARRAY_POINTING_xxx/BEAM_xxx
  - /SUB_ARRAY_POINTING_xxx/BEAM_xxx/STOKES_x
  - /SUB_ARRAY_POINTING_xxx/BEAM_xxx/COORDINATES

- The data are found by opening the STOKES_x folder.
Reading Data Using Python

- Use h5py library to read HDF5 files
  - ```python
  >> import h5py
  >> f = h5py.File('BF_file.h5')
  ```

- To list folders:
  - ```python
  >> f['/'].items()
  Out[..]:
  [(u'SUB_ARRAY_POINTING_000',<HDF5 group "/SUB_ARRAY_POINTING_000" (2 members)>),
   'SYS_LOG', <HDF5 group "/SYS_LOG" (0 members)>)]
  ```
Reading Data Using Python

- To list meta-data (“attributes”):
  - `>> f['/'].attrs.items()
    Out[..]:
    [(u'DOC_VERSION', '2.5.0'),
     (u'OBSERVATION_NOF_SUB_ARRAY_POINTINGS', 1), ..
    ...]

- To get a known attribute:
  - `>> f['/'].attrs.get('TELESCOPE')
    Out[..]: 'LOFAR'

- To list keys:
  - `>> f['/'].attrs.keys()
    Out[..]:
    [u'DOC_VERSION',
     u'OBSERVATION_NOF_SUB_ARRAY_POINTINGS', ..
    ...]
Reading Data Using Python

- Put meta-data into Python dictionary:
  - `>> parameters = dict(f['/'].attrs.items())`
  - `>> parameters['TELESCOPE']`
    
    Out[..]: 'LOFAR'

- Read data:
  - `>> data =
    f['/SUB_ARRAY_POINTING_000/BEAM_005/STOKES_0']`
  - `>> data.shape`
    
    Out[..]: (3232896, 6400)

- Data are stored as a 2-D array of [time,frequency]
Reading and Visualising Beam-formed Data using the Dynspec Package
The Dynamic Spectrum Toolkit

The Dynamic spectrum Toolkit Container (DTC) is a complete package of tools to reduce, process and visualise LOFAR beam-formed data:

- “Quicklook” tool to obtain quicklook images
- Extract particular time and frequency ranges of interest
- Rebin the data to average in time and/or frequency
- “Subtraction” tool to subtract one beam from another (for example to subtract an off-source beam from an on-source) using $Z=X-(k*Y)$ where $k$ is a free scaling factor
- “LinPol” tool, to convert $I, Q, U, V$ dynamic spectra to $I, \text{linear}, \text{PA}, \text{Total}$ dynamic spectra
- Visualisation tool to explore the data
The Dynamic Spectrum Toolkit

• Full details for installing and running can be found at:

Quicklook Plots

• Run Beam2Dynspec-Quick

Usage: Beam2Dynspec-Quick --id=Lxxx --obsDIR= --outputDIR= --percentTimeData=[0->1] --percentSpectralData=[0->1] --transpose=[yes or no] --nofPart=

• Uses percentage of data to create jpg files of dynamic spectra.
Rebin Tool

- Run Beam2Dynspec-Rebin

 Usage: Beam2Dynspec-Rebin --id=Lxxx --obsDIR=
       --outputDIR= --tmin= --tmax= --tscale= --fmin=
       --fmax= --chanPerSubband= --RAM=(x in Gb) --Npart=
       --RebinAll=(yes or no)
       [if Rebinall=yes default => 0: else –NSAP=]

- Rebins data to desired time/frequency resolution and
  outputs data from all beams in a “dynspec” HDF5 format.
Visualisation Tool

Run as Dynspec-Visu