

Observing pulsars with NenuFAR



Observing pulsars with NenuFAR

Step 1: The cleaning



Observing pulsars with NenuFAR

Step 1: The cleaning



Code used: `python NenuPlot.py`

The essential options:

- freqappend** : addition of several frequency bands
- flat_cleaner** : flat cleaning
- nodefaraday** : do not correct the RM in the data
- arout** : output psrchive file (.clear)

Practical options:

- t** : reduce the number of temporal integration by the requested factor (before RFI mit)
- ta** : reduce the number of temporal integration by the requested factor (after RFI mit)
- small_pdf** : reduce the size of the quicklook pdf (using Ghostscrip)
- metadata_out** : ascii file with metadata (.metadata)
- iterative** : iterative cleaning (more precise but 3 time longer)

Observing pulsars with NenuFAR

Step 1: The cleaning



Example:

```
python NenuPlot.py -ta 6 -freqappend -flat_cleaner -nodefaraday -arout -small_pdf -metadata_out -iterative /datab
nenufar-pulsar/ES03/2020/06/B0950+08_D20200611T1607_59011_*.ar
```

Loop 1:

- The bandpass is **flattened**
- A **dirty template** is made
- A **dirty on-pulse** is calculated
- The dirty on-pulse is removed in the data set
- CoastGuard is run on the dedispersed data with a threshold of **8 sigma**
- A **new template** is generated

Loop 2:

- The bandpass is **flattened**
- A **beter on-pulse** is calculated from the last template
- The **new on-pulse** is removed from a new copy of the **original data set**
- CoastGuard is run on the dedispersed data with a threshold of **3.5 sigma**
- A **new template** is generated
- A **first mask** is generated

Loop 3:

- The bandpass is **flattened** using a **refined** bandpass measurement
- A **refined on-pulse** is calculated from the last template
- The **refined on-pulse** is removed from a new copy of the **original data set** on which is installed the new mask **beter**
- CoastGuard is run on the dedispersed data with a threshold of **3.5 sigma**
- A **refined template** is generated
- A **refined mask** is generated

recursively up to loop 12 with the option **-iterative**

Observing pulsars with NenuFAR

Step 1: The cleaning

Example:

```
python NenuPlot.py -ta 6—freqappend -flat_cleaner  
nenufar-pulsar/ES03/2020/06/B0950+08_D2020061
```

```
file(s) used: ['B0950+08_D20200611T1607_59011_250606_0028_BEAM1.  
=====
```

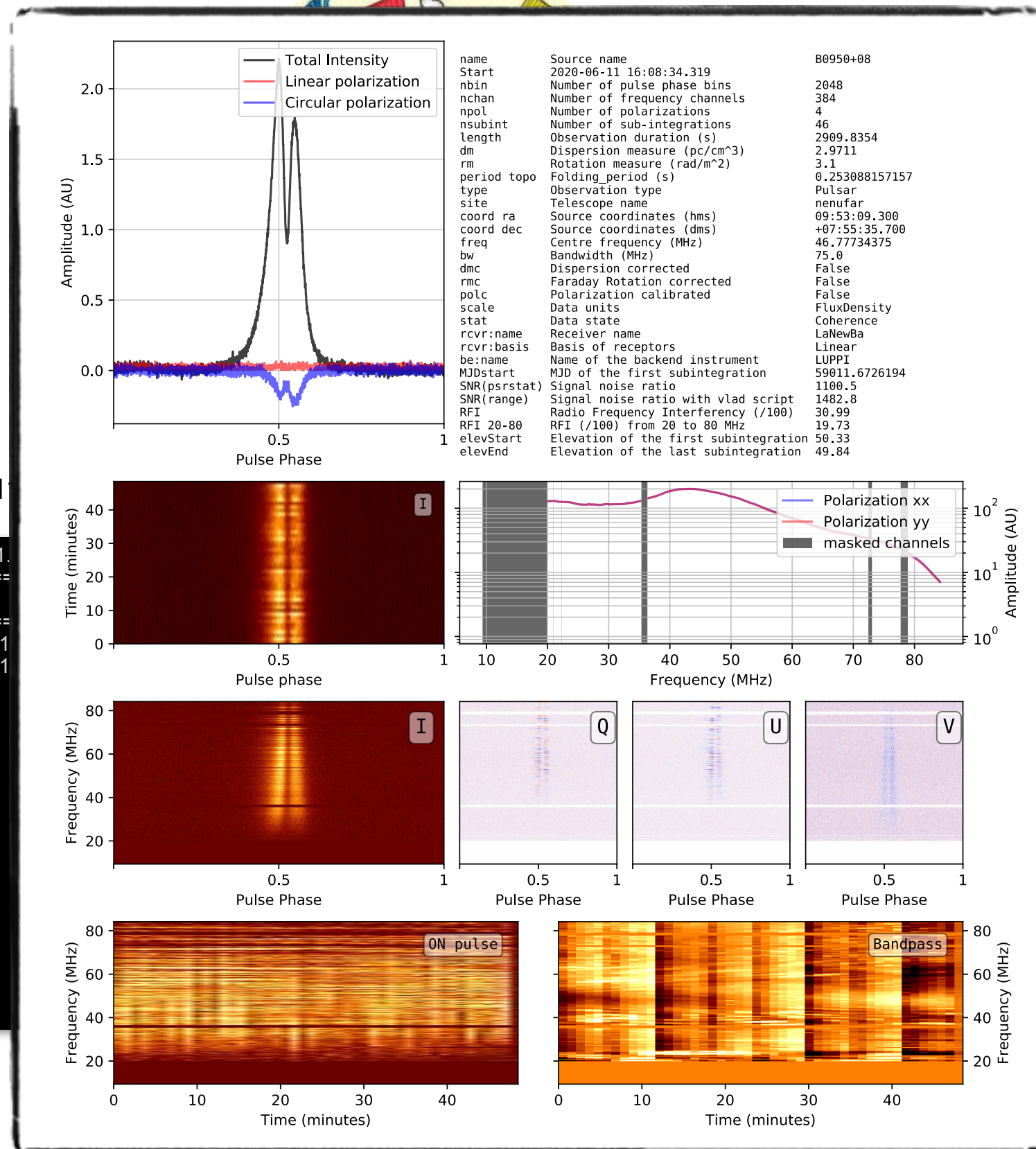
File(s) to be processed:

```
=====
```

/data/f2/nenufar-pulsar/ES03/2020/06/B0950+08_D20200611T1607_59011_250606_0028_BEAM1.ar
/data/f2/nenufar-pulsar/ES03/2020/06/B0950+08_D20200611T1607_59011_250606_0028_BEAM1.metadata
Loop: 1
ON pulse window is 1949-125
chanthresh = 8.0 subintthresh = 8.0
RFI fraction is 17.51 percent
[...]
Loop: 12
ON pulse window is 1920-127
chanthresh = 3.5 subintthresh = 4.0
RFI fraction is 28.27 percent
Cleaning was interrupted after 12 loops

```
> ls -tr  
B0950+08_D20200611T1607_59011_250606_0028_BEAM1.ar.clear  
B0950+08_D20200611T1607_59011_250606_0028_BEAM1.metadata  
B0950+08_D20200611T1607_59011_250606_0028_BEAM1.pdf
```

Output files



Observing pulsars with NenuFAR

Step 1: The cleaning

Example:

python NenuPlot
nenufar-pulsar/E

```
file(s) used: ['B0950+08']  
=====
```

```
=====
```

```
/data/f2/nenufar-pulsar  
/data/f2/nenufar-pulsar  
Loop: 1  
ON pulse window is 1  
chanthresh = 8.0 sub  
RFI fraction is 17.51 p  
[...]  
Loop: 12  
ON pulse window is 1  
chanthresh = 3.5 subintthresh = 4.0  
RFI fraction is 28.27 percent  
Cleaning was interrupted after 12 loops
```

```
> ls -tr  
B0950+08_D20200611T1607_59011_250606_0028_BEAM1.ar.clear  
B0950+08_D20200611T1607_59011_250606_0028_BEAM1.metadata  
B0950+08_D20200611T1607_59011_250606_0028_BEAM1.pdf
```

Output files

