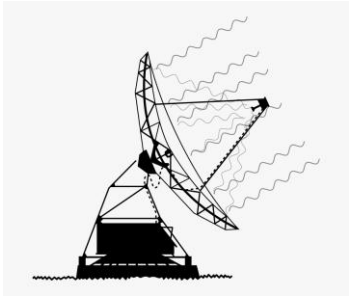
How to use Multi

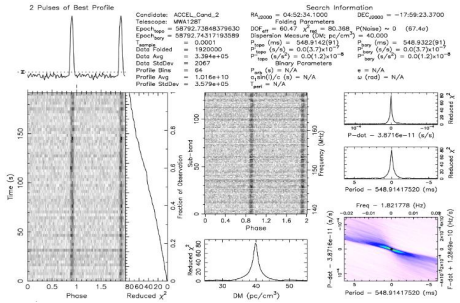
Qiuyang Fu

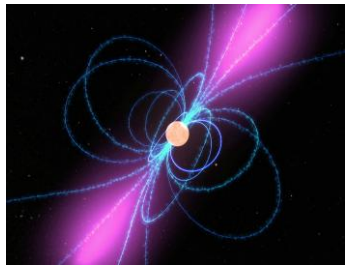
2025.12.02

National Astronomical Observatories, Chinese Academy of Sciences

Background



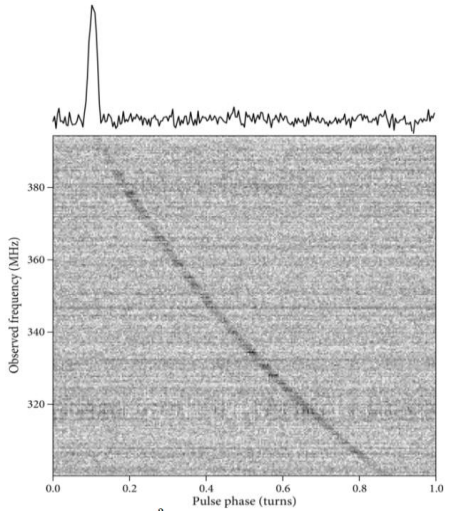




Data → Search algorithm → Pulsar candidates → sifting → Pulsar

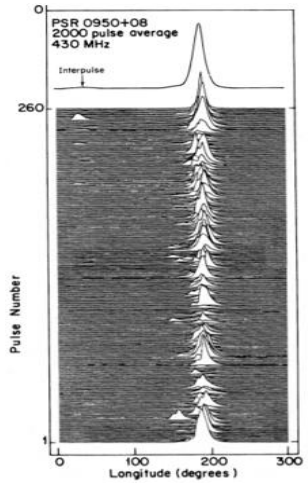
1.Dispersion Measure

$$DM = \int_0^d n_e dl$$



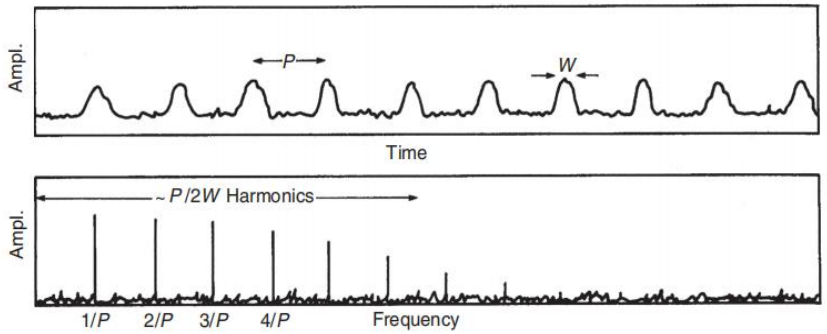
Credit: Essential Radio Astronomy

2. A single pulse signal is weak ⇨ Integrated pulse profiles according to the period



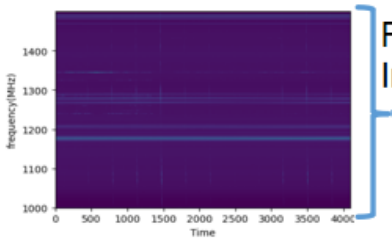
Credit: Hankins &Cordes, 1981

3. Fourier transform ⇨ Pulsar period



Credit: Pulsar Astronomy

Method



Frequency Integration

De-dispersion DMi

FFT i

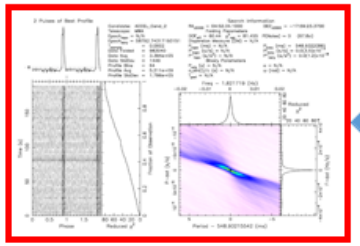
Candidates (DM, Period)

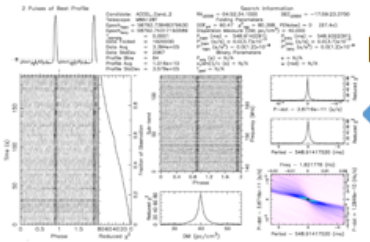
Sift&Fold

Time Frequency

Time Series i

Time

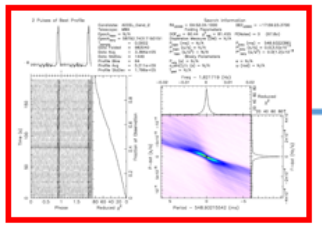




Filtering "snapshot" candidates generated from folding time series data

Reducing the number of folding full data

Accelerating folding to reduce computational costs

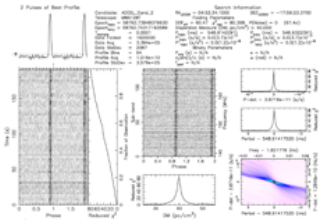


AI Filter

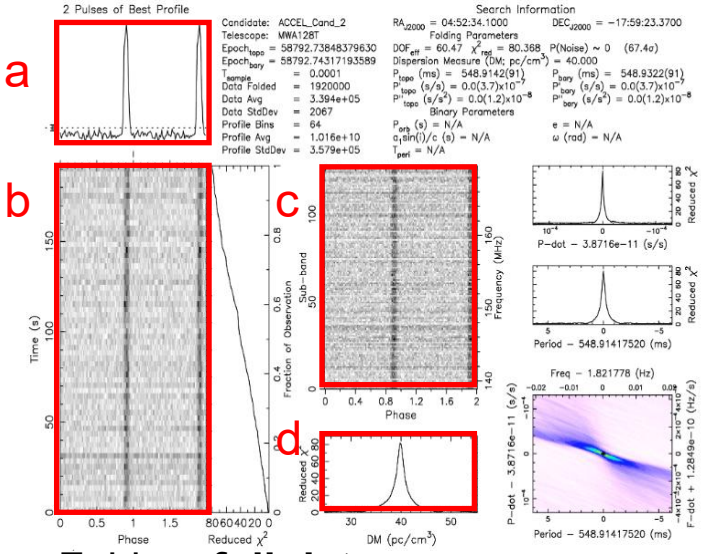
Candidates (DM, Period)

Fold

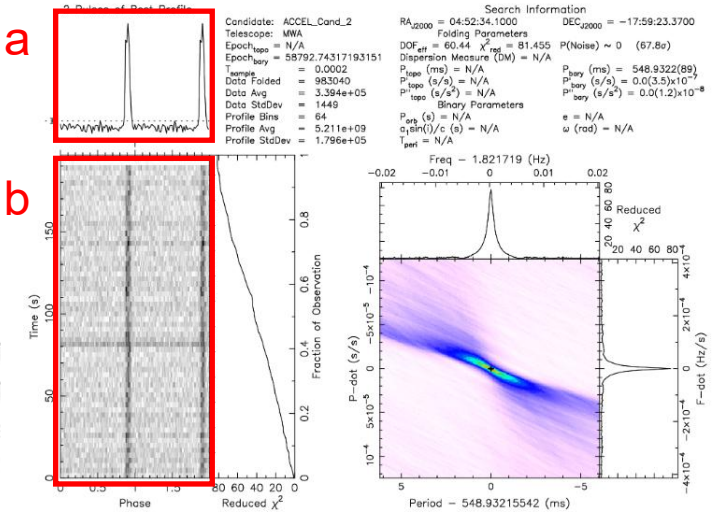
Time Frequency



Software: PRESTO



Fold on **full data**



Fold on **time series data**
"snapshot candidates"

Usage

Code: https://github.com/ifuqy/Multi

This model's code is implemented using Tinygrad.

The testing environment for this program is as follows:

- tinygrad==0.10.2

git+https://github.com/tinygrad/tinygrad.git@v0.10.2 or https://github.com/tinygrad/tinygrad/releases (tinygrad 0.11.0 Latest)

- matplotlib==3.8.4
- numpy==1.22.4
- pyyaml==5.4.1
- scikit-learn==1.1.3
- tqdm==4.66.1
- scipy==1.11.3

Please avoid cloning the repository directly using **git clone https://github.com/tinygrad/tinygrad.git**, as this may introduce unexpected bugs. Instead, download and use **Release version**, which currently supports the latest **v0.11.0**.

To install required dependencies, run: **pip install -r requirements.txt**

Usage

First, modify the file
`/Multi/blob/main/config/global_cfg.yaml`
and change:
`DEVICE: NV:1`
to:
`DEVICE: "Default"`
This allows the model to automatically select the most suitable device for computation.

```
# Support Device: ['WEBGPU', 'REMOTE', 'QCOM', 'PYTHON', 'NV', 'NULL', 'NPY',
#                  'METAL', 'LLVM', 'HIP', 'GPU', 'DSP', 'DISK', 'CUDA', 'CPU', 'AMD']
DEVICE: NV:1  # If a default value needs to be retained, keep "Default".
```

Usage

Enter the project directory:

```
cd Multi/
```

Then run:

```
python predict.py \
  --ckpt ./trained_model/weight_0.9954_0.9830.pth \
  --outfile ./test_pfdfile.txt \
  --pfd_dir ./test_pfds/ \
  --use_prob \
  --chunk_size 500
```

Here:

- `--ckpt` specifies the model checkpoint to use.
- `--outfile` is the file where the prediction scores for all .pfd candidates will be saved.
- `--pfd_dir` is the directory containing the .pfd candidate files.
- Reduce the `chunk_size` value if you encounter RAM or GPU out-of-memory issues.

```
/mwa_vcs/MWA_cnnic/Multi/M5_knownpulsar/M5_DM29.35_JERK_Cand_226.pfd    0.9964163303375244
/mwa_vcs/MWA_cnnic/Multi/M5_knownpulsar/M5_DM29.40_JERK_Cand_335.pfd    0.5524632930755615
/mwa_vcs/MWA_cnnic/Multi/M5_knownpulsar/M5_DM29.30_JERK_Cand_202.pfd    0.9933492541313171
/mwa_vcs/MWA_cnnic/Multi/M5_knownpulsar/M5_DM29.45_JERK_Cand_108.pfd    0.9944450259208679
/mwa_vcs/MWA_cnnic/Multi/M5_knownpulsar/M5_DM29.35_JERK_Cand_154.pfd    0.997330904006958
/mwa_vcs/MWA_cnnic/Multi/M5_knownpulsar/M5_DM30.05_JERK_Cand_1.pfd      0.9907776117324829
/mwa_vcs/MWA_cnnic/Multi/M5_knownpulsar/M5_DM29.35_JERK_Cand_476.pfd    0.9847046136856079
```

DM: 29.35
Index: 226
Score: 0.996

Thanks