Pulsar Timing with *Fermi* LAT

Aous Abdo
< Aous.Abdo@nrl.navy.mil >
Paul Ray
<Paul.Ray@nrl.navy.mil>
Naval Research Laboratory
Washington DC

Thanks to David Nice and Scott Ransom for useful material!

Pulsar Timing

- Precise monitoring of the rotation of the neutron star through the tracking of the times of arrival (TOAs) of EM pulses
- Accounts for every single rotation of the neutron star over long periods of times (years to decades)
- This allows for:
    - Performing extremely accurate astrometric measurements
    - Probe the interior physics of the neutron star
    - Test gravitational theories in strong gravitational fields

Pulsar Timing

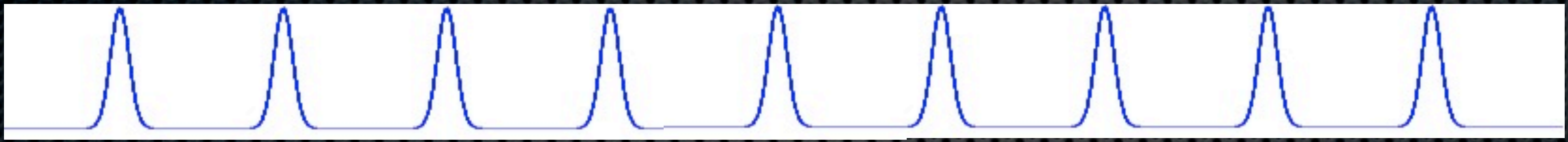
- Goal: determine a timing model that accounts for all of the observed pulse arrival times (TOAs)
- Although the shape of individual pulses vary, the shape of the average profile over a short time period is quite stable.
- This allows for the "folding" of individual pulses modulo the instantaneous pulse frequency (ν) of the neutron star.

Pulsar Timing

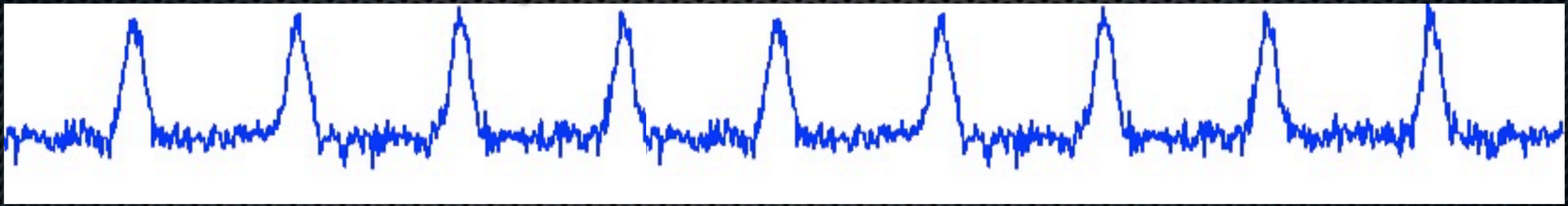
- Parameters that can be determined:
  - Spin parameters:
    - frequency (ν) and frequency derivatives (ν', ν'', ...)
      ➙ spin-down age, magnetic field, spin-down luminosity, torques
  - Orbital parameters:
    - orbital period (P_{orb}), projected semi-major axis (a sin (i)), longitude of periastron (ω), time of periastron (T_0), and the eccentricity of the orbit(e)
      ➙ General Relativity terms
  - Positional parameters:
    - Ecliptic longitude (λ), ecliptic latitude (β), proper motion

What the signal looks like

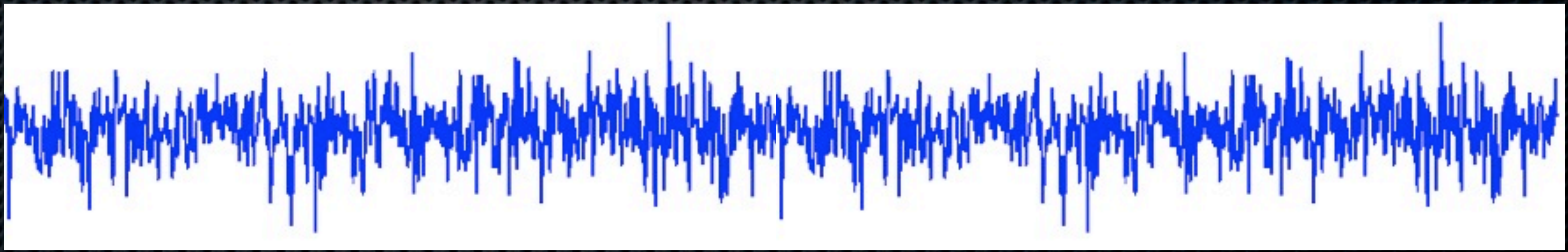
Ideally the signal will look like this



But there is always some noise



And often individual pulses are not visible at all



Time of Arrival

- In timing, the key quantity of interest is the time of arrival (TOA) of pulses at the telescope.
- It is defined as the arrival time of some fiducial point on the profile.
- Since the average pulse profile has a stable form, TOAs can be determined accurately by cross-correlating the observed profile with a high signal-to-noise template profile.

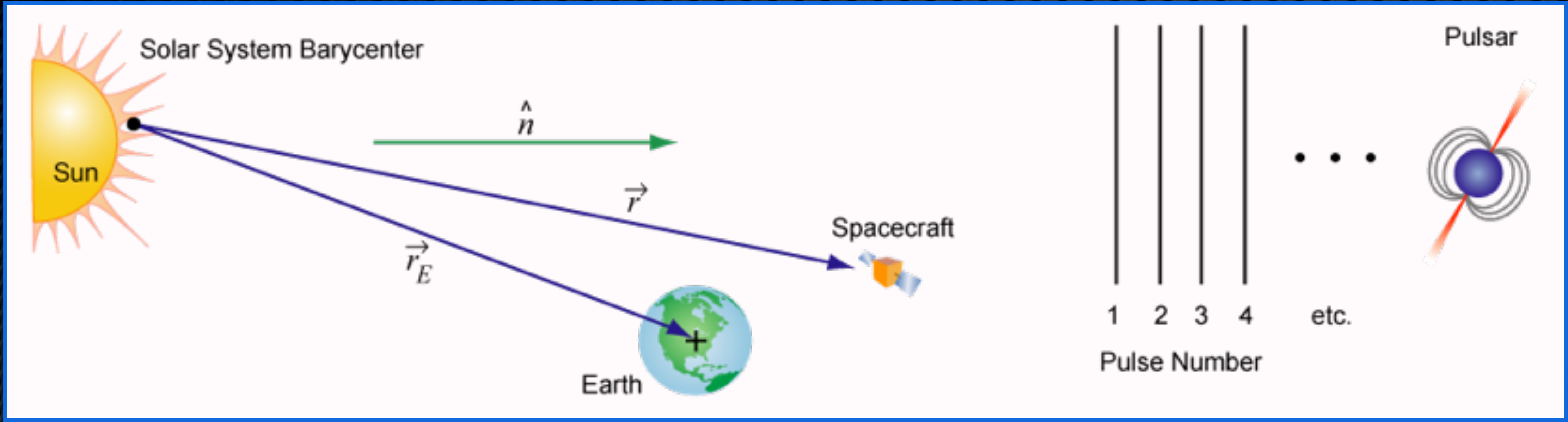
$$P(t) = a + bT(t - \tau) + N(t)$$

- $P(t)$: sampled profile, $T(t)$: template profile, $N(t)$: noise
  a: arbitrary offset, b: scaling factor, and τ is the time shift between the profile and the template

TOAs example

Observatory			Radio Frequency	Pulse Time of Arrival	Measurement Uncertainty		
a	3751	1518+49	370.000	50942.02369981804596	69.1	9-May-98	460.2
a	3751	1518+49	370.000	50942.02508871578912	74.9	9-May-98	460.8
a	3752	1518+49	370.000	50942.02710263928441	107.8	9-May-98	460.1
a	3752	1518+49	370.000	50942.02849153928888	68.4	9-May-98	463.7
a	3753	1518+49	370.000	50942.03050309034722	63.0	9-May-98	459.7
a	3753	1518+49	370.000	50942.03189199466585	71.4	9-May-98	468.7
a	3754	1518+49	370.000	50942.03389643284537	64.2	9-May-98	461.5
a	3754	1518+49	370.000	50942.03528532340819	57.4	9-May-98	456.3
a	3755	1518+49	370.000	50942.03728740139970	74.4	9-May-98	459.7
a	3755	1518+49	370.000	50942.03867629785610	65.1	9-May-98	461.5
a	3756	1518+49	370.000	50942.04067884384616	54.2	9-May-98	458.8
a	3756	1518+49	370.000	50942.04206774860490	87.3	9-May-98	470.2
a	3757	1518+49	370.000	50942.04406981298474	88.9	9-May-98	461.2
a	3757	1518+49	370.000	50942.04545870833792	71.8	9-May-98	463.1
a	3758	1518+49	370.000	50942.04748447411745	110.3	9-May-98	463.0
a	3758	1518+49	370.000	50942.04887336536594	78.6	9-May-98	461.1
a	3759	1518+49	370.000	50942.05089865820880	60.2	9-May-98	463.4
a	3759	1518+49	370.000	50942.05228755033977	131.1	9-May-98	463.1
a	3760	1518+49	370.000	50942.05428961858992	63.4	9-May-98	460.9
a	3760	1518+49	370.000	50942.05567851214494	93.2	9-May-98	462.8
a	3761	1518+49	370.000	50942.05768105475176	116.2	9-May-98	461.0
a	3761	1518+49	370.000	50942.05906994776154	75.0	9-May-98	463.0
a	3762	1518+49	370.000	50942.06108244410689	72.2	9-May-98	465.9
a	3762	1518+49	370.000	50942.06247133259781	76.9	9-May-98	463.6
a	3763	1518+49	370.000	50942.06450988581265	86.1	9-May-98	461.4
a	3763	1518+49	370.000	50942.06589877480622	61.9	9-May-98	460.4
a	3764	1518+49	370.000	50942.06790794988299	90.1	9-May-98	460.5
a	3764	1518+49	370.000	50942.06929683956486	67.2	9-May-98	460.8
a	3765	1518+49	370.000	50942.07129227137214	63.5	9-May-98	460.6
a	3765	1518+49	370.000	50942.07268116130441	139.5	9-May-98	461.8

Barycentering TOAs



- Our observing frame (Earth or satellite) is not inertial. We need to convert the TOAs to a nearly inertial frame before attempting to fit a timing model.
- The center-of-mass of the solar system, Solar System Barycenter (SSB), can be regarded as an inertial frame.
- We transform the observed TOAs to this frame.
- Correct for relativistic time delay due to the presence of masses in the Solar System.

Timing Model

- In the initial frame of the pulsar, we can Taylor expand the spin frequency about an epoch t_0:

$$\nu(t) = \nu_0 + \dot{\nu}_0(t - t_0) + \frac{1}{2}\ddot{\nu}_0(t - t_0)^2 + ...$$

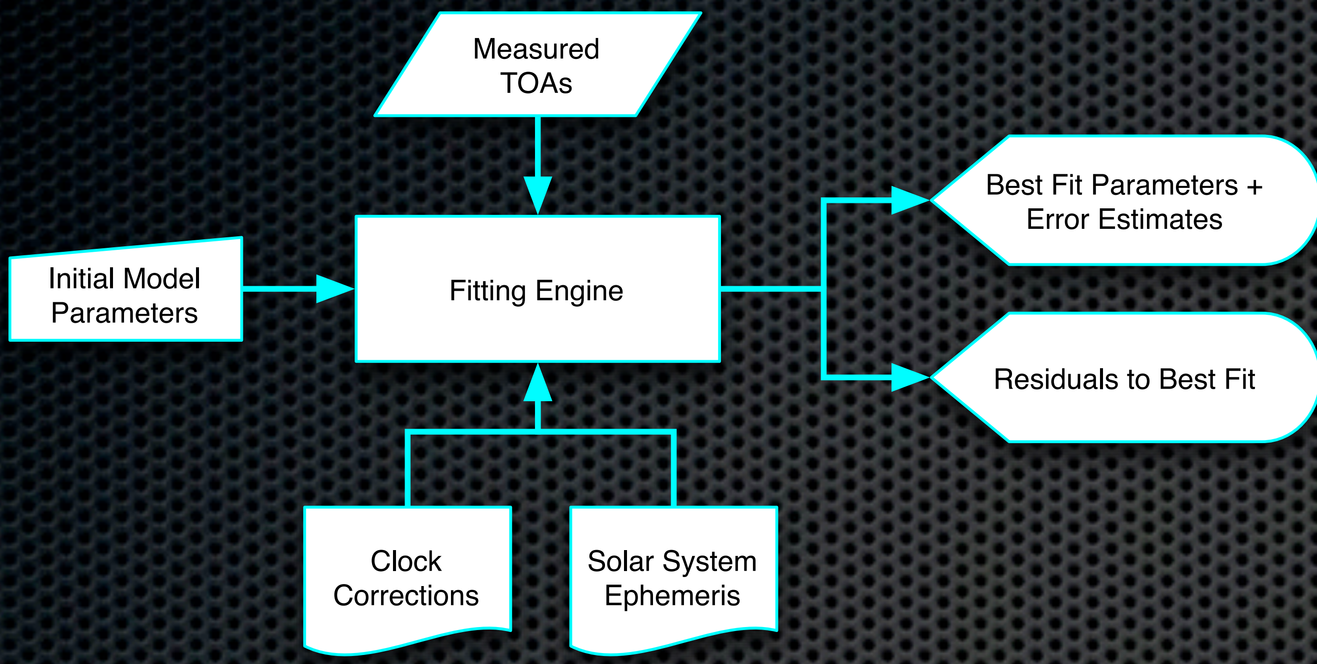
- Integrating the above equation we get the rotational phase:

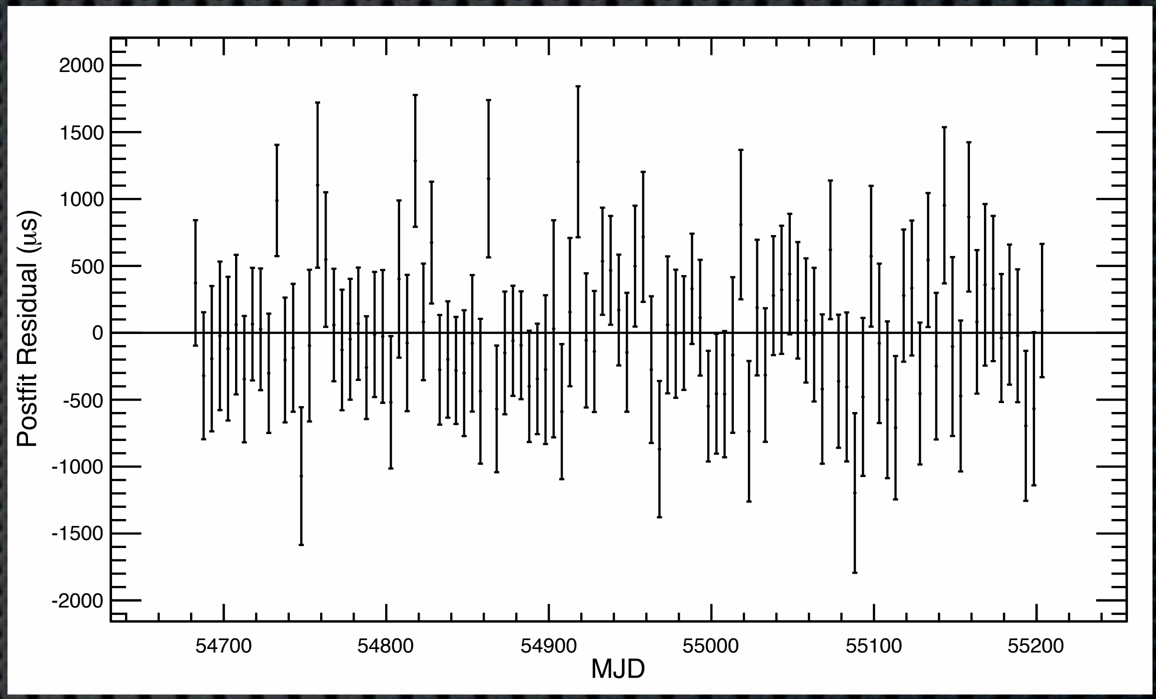
$$\phi(t) = \phi_0 + \nu_0(t - t_0) + \frac{1}{2}\dot{\nu}_0(t - t_0)^2 + \frac{1}{6}\ddot{\nu}_0(t - t_0)^3 + ...$$

  where ϕ_0 is the pulse phase at t_0.

- Full model can include spin, astrometric, binary, and other parameters

Fitting TOAs to a Timing Model





- Find parameters that minimizes the residuals between the data and the model.
- Ideally the residuals should have a mean of zero and show no systematic trends.

Tools for Fitting Timing Models

- Tempo < http://pulsar.princeton.edu/tempo/ >
  - Developed by Princeton and ATNF over 30+ years
  - Well tested and heavily used
  - Based on TDB time system
  - But, nearly undocumented, archaic FORTRAN code

Time Systems

TAI = Atomic time based on the SI second
UT1 = Time based on rotation of the Earth
UTC = TAI + "leap seconds" to stay close to UT1
TT = TAI + 32.184 s
TDB = TT + periodic terms to be uniform at SSB
TCB = Coordinate time at SSB, based on SI second

- Tempo2 < http://www.atnf.csiro.au/research/pulsar/tempo2/>
  - Developed at ATNF recently
  - Based on TCB time system (coordinate time based on SI second)
  - Well documented, modern C code, uses long double (128 bit) throughout
  - Easy plug-in architecture to extend capabilities

PRESTO

- Developed by Scott Ransom <http://www.cv.nrao.edu/~sransom/presto/>
- A large suite of pulsar search and analysis software
- The software is composed of numerous routines designed to handle three main areas of pulsar analysis:
  1. Data Preparation: Interference detection and removal, de-dispersion, barycentering (via TEMPO).
  2. Searching: Fourier-domain acceleration, single-pulse, and phase-modulation (or sideband) searches.
  3. Folding: Candidate optimization and Time-of-Arrival (TOA) generation.

Analysis Steps

1. Extract photons around a pulsar with a predetermined ROI, energy, and time ranges. (gtselect)
2. Bin the data and make counts maps (gtbin)
3. Barycenter the events (gtbary)
4. Fold the photons with prepfold
5. Obtain a good timing model with tempo2

1. Data Selection & Extraction

- Assuming one have downloaded:
  - An event file FT1.fits <http://fermi.gsfc.nasa.gov/cgi-bin/ssc/LAT/LATDataQuery.cgi>
  - A space craft history file FT2.fits <http://fermi.gsfc.nasa.gov/cgi-bin/ssc/LAT/LATDataQuery.cgi>
  - An ephemeris file for the pulsar <http://fermi.gsfc.nasa.gov/ssc/data/access/lat/ephems/>
- The first step is to apply data selection to the event file.
- Perform data selection using gtselect:

```
$ gtselect infile=FT1.fits outfile=psr_FT1_filt.fits ra=1.7565
dec=73.05225 rad=1.50 tmin=239557517 tmax=277689600 emin=100.0
emax=100000.0 evclsmin=3 zmax=105.0 evclsmax=3clobber=yes
```

- Notice that we selected a minimum energy cut of 100 MeV and an ROI of 1.5 degree around the source.

2. Generating Counts Map

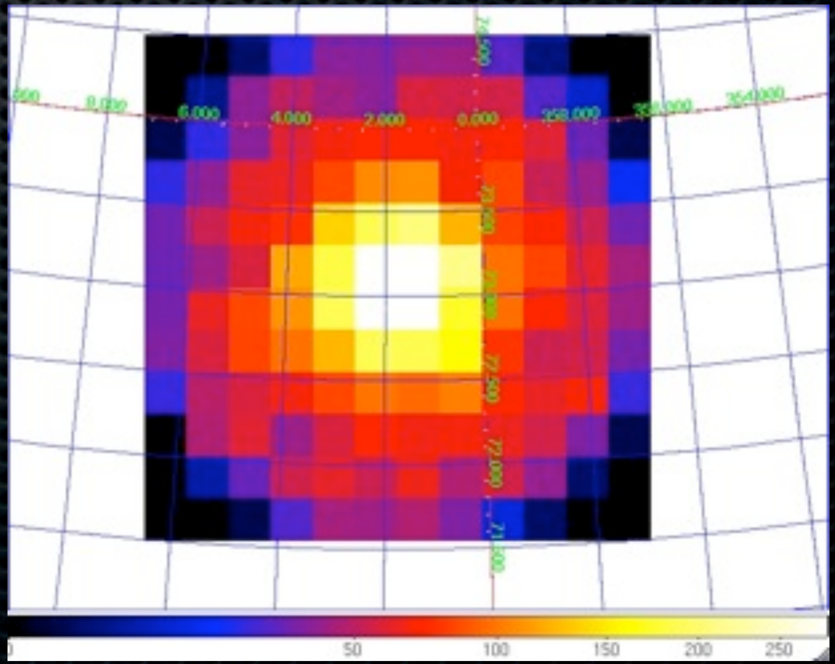
- This step could be skipped but it is always good to check how the region of interest looks like:

- Using the output file from gtselect create a counts map:

```
$ gtbin algorithm=CMAP evfile=psr_FT1_filt.fits scfile=FT2.fits
outfile=psr_cmap_3.fits nxpix=12 nypix=12 binsz=0.25 axisrot=0.0
coordsys=CEL xref=1.7565 yref=73.05225 proj=AIT
```

- Parameters :
  - binning algorithm: cmap for counts map
  - parameters that determine how to bin the image
  - coordinate system to bin the map in
  - reference points of our source in that coordinate system
  - type of projection

Counts Map of the region around the pulsar

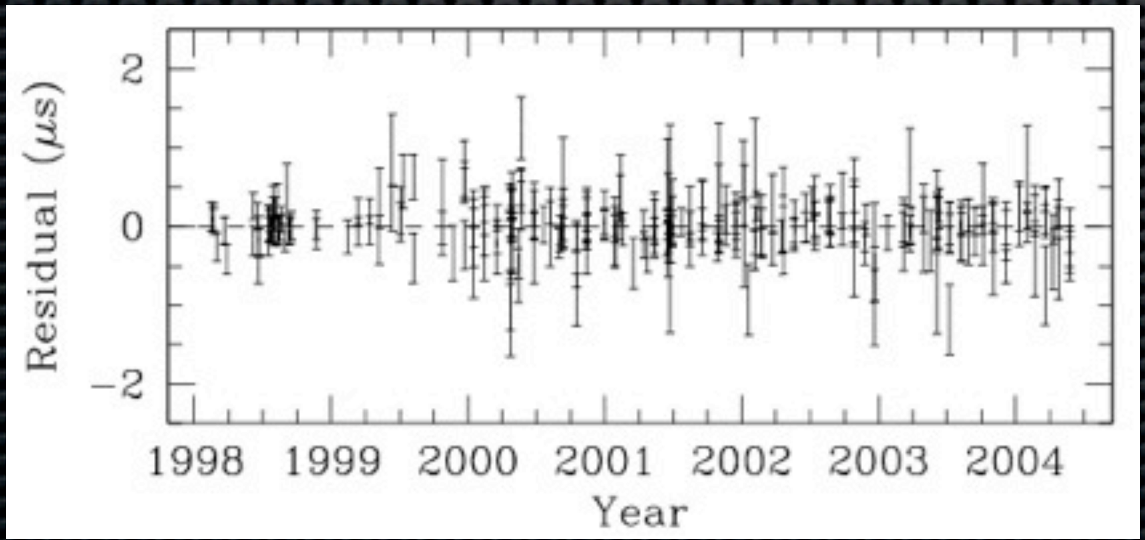


3. Barycentering the Events

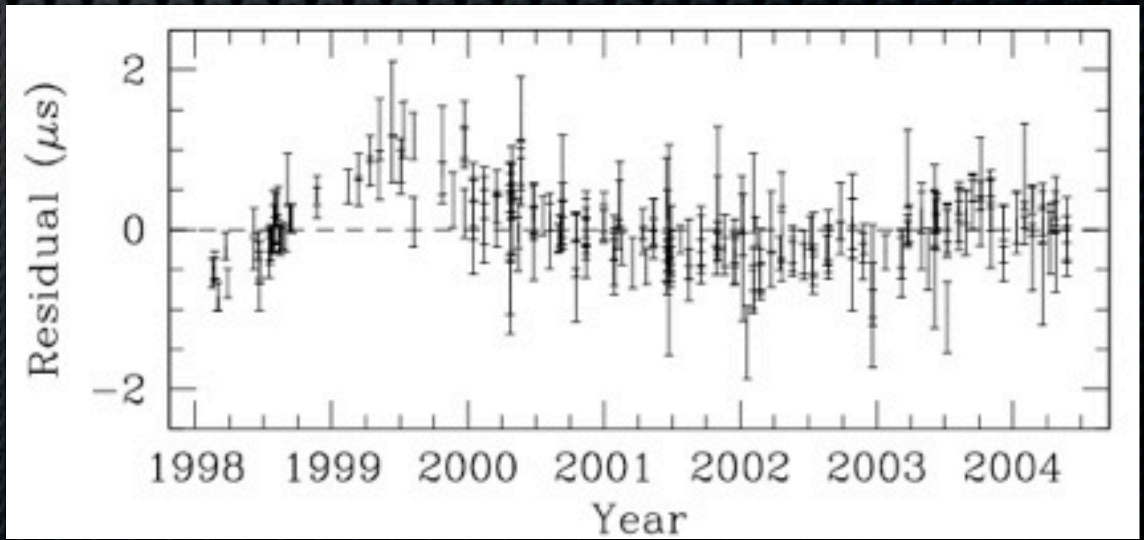
```
$ gtbary evfile=psr_FT1_filt.fits scfile=FT2.fits
outfile=psr_FT1_filt_bary.fits ra=1.7565 dec=73.05225 solareph="JPL
DE405"
```

- User needs to specify the R.A. and DEC of the pulsar as well as the planetary ephemeris to be used.
- Select JPL DE405 for planetary ephemeris
- With previous-generation planetary ephemeris, DE 200, there is a 300 m error in Earth's position which will result in ~ 1μs timing error.

PSR J1713+0747 analyzed using DE 405 solar system ephemeris



PSR J1713+0747 analyzed using DE 200 solar system ephemeris.



4. Folding the Events

- Now we have all the selected photons barycentered at the pulsar's location and ready to be folded. For this purpose we will use the PRESTO package
- Make a .bin and .inf files to be used with PRESTO's prepfold:

```
$ makebininf.py psr_FT1_filt_bary.fits
FT1 file:  psr_FT1_filt_bary.fits
psr_FT1_filt_bary
# TIMEZERO =  0.0
# MJDREF =  51910.0007429
Source Name :  psr
fits2bin psr_FT1_filt_bary.fits psr_FT1_filt_bary.bin 0.0
NAXIS2 = 9979
Got HDU of type 2
Using MJDREF 51910.000742870368, TIMEZERO 0.000000000000
Found TIME column number 10
Converted 9979 points.
     First was      54682.7017463557    Last was      55123.9273059241
```

- Makebininf.py is a contributed script <http://fermi.gsfc.nasa.gov/ssc/data/analysis/user/> that reads barycentered FT1 fits file and generate a .bin file containing double precision MJD event times and a .inf file that describes it. These two files form the input for prepfold and other PRESTO codes.

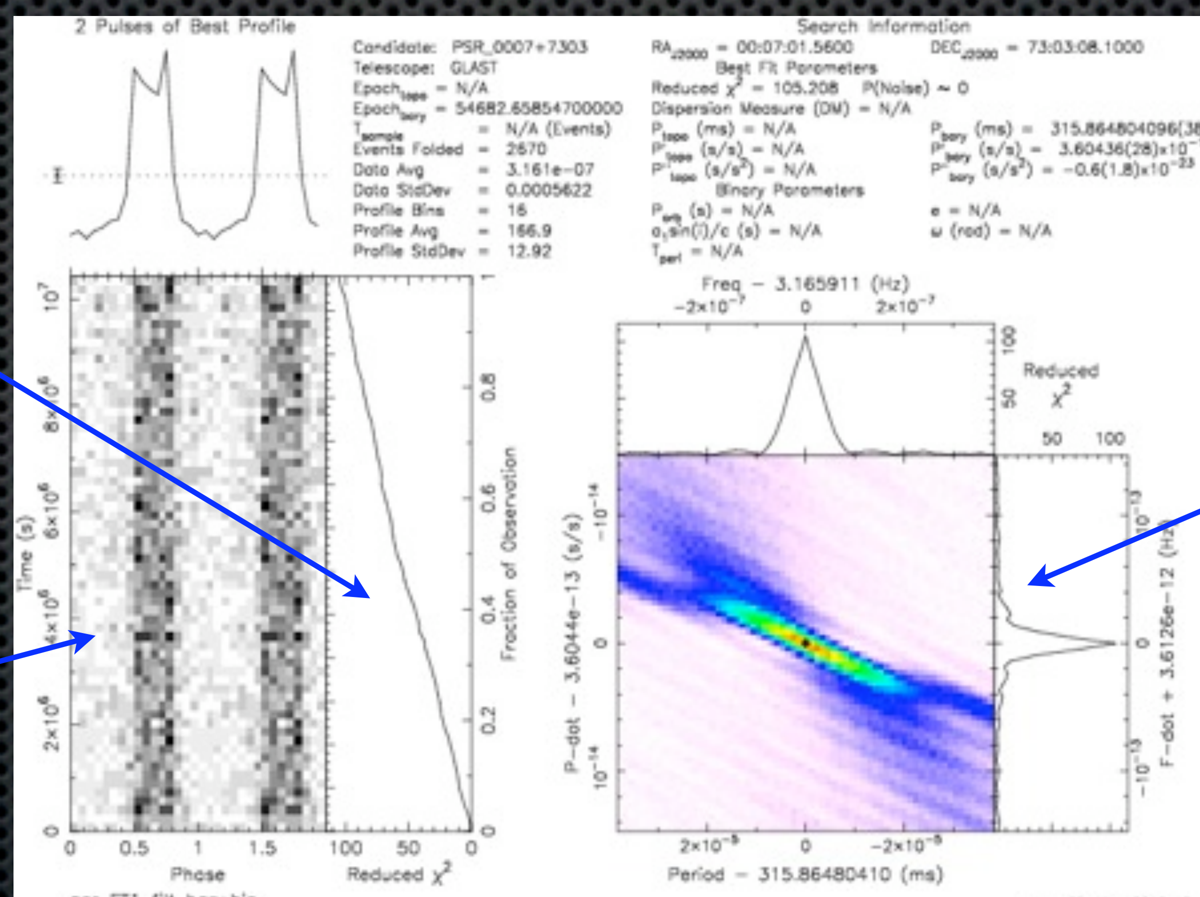
4. Folding the Events Cont'd

- Now we use prepfold to fold the photons at the frequency and frequency derivatives, and other parameters, given in the .par file

```
$ prepfold -events -mjds -double -par psr.par psr_FT1_filt_bary.bin
```

growth of reduced chi-square with time.

Pulse profile as a function of time (phaseogram)

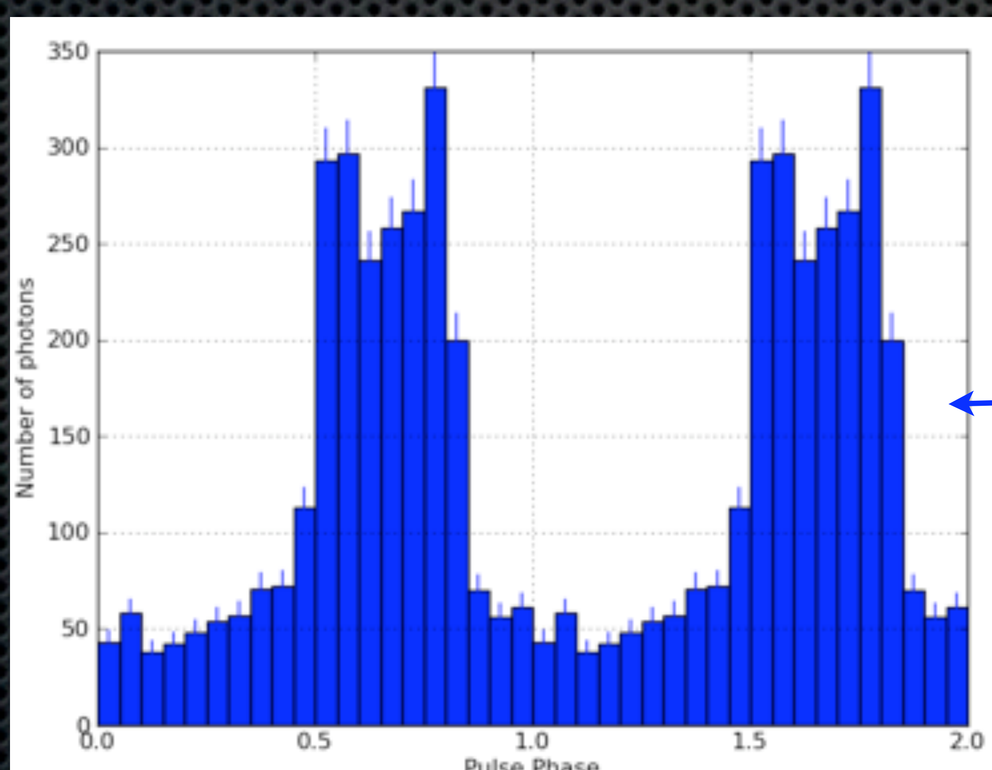


Distribution of reduced chi-square in the f-fdot space.

Thursday, May 13, 2010 (DOY 133)

Pulsar Light Curve

- Another output of prepfold is a text file containing the pulse profile numbers.
- One can use this to plot the pulse profile and we will use it later in the tutorial to show you how to create a gaussian profile template.

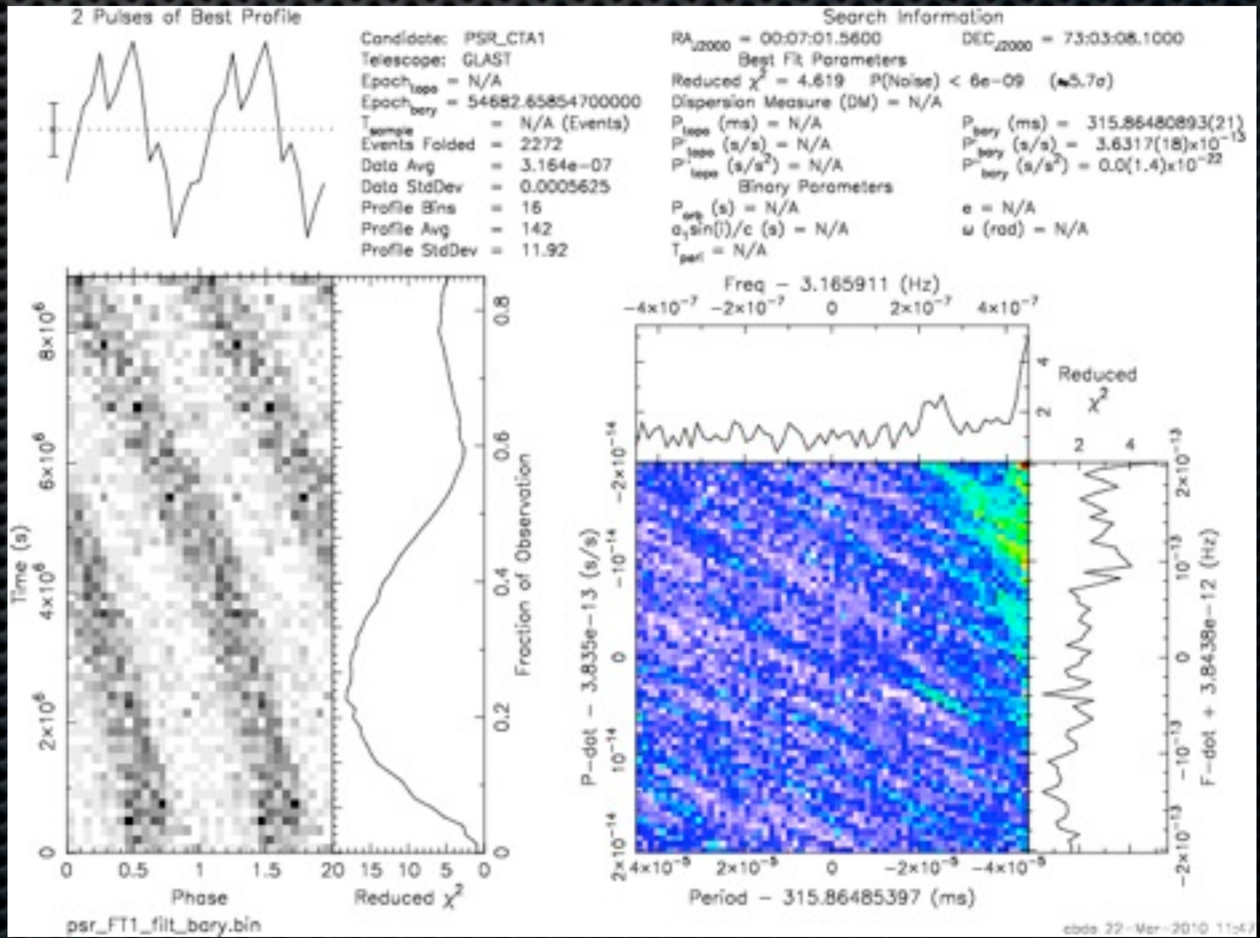


```
abdo@heselin:/Analysis/abdo/psrsearch/CTA1/CTA1II/timing/tutorial — ss...
abdo@hes...orial — ssh    ssh    bash
# Input file           =  psr_FT1_filt_bary.bin
# Candidate            =  PSR_0007+7303
# Telescope            =  GLAST
# Epoch_topo           =  N/A
# Epoch_bary (MJD)     =  54682.658546999999
# T_sample             =  0.0157933
# Data Folded          =  659887920
# Data Avg             =  2.02307083770253e-07
# Data StdDev          =  0.000449785597557607
# Profile Bins         =  20
# Profile Avg          =  133.500000710418
# Profile StdDev       =  11.5542200390341
# Reduced chi-sqr      =  84.664
# Prob(Noise)          <  0
# P_topo (ms)          =  N/A
# P'_topo (s/s)        =  N/A
# P''_topo (s/s^2)     =  N/A
# P_bary (ms)          =  315.86480409561  +/- 3.36e-08
# P'_bary (s/s)        =  3.60435833064993e-13 +/- 2.5e-17
# P''_bary (s/s^2)     =  -6.2072190591656e-24 +/- 1.55e-23
# P_orb (s)            =  N/A
# asin(i)/c (s)        =  N/A
# eccentricity         =  N/A
# w (rad)              =  N/A
# T_peri               =  N/A
######################################################
   0  43
   1  58
   2  38.00001
   3  42.00001
   4  48.00001
   5  54
   6  57.00001
   7  71.00001
   8  72.00001
   9  113
  10  293
  11  297
  12  241
  13  258
  14  267
  15  331
  16  200
  17  69.9999
  18  56
  19  61
```

Bin #
cts/bin

Dealing with Timing Noise

- The previous case was an ideal case where the timing model was up-to-date and the phaseogram looked straight throughout the time period over which we are timing the pulsar.



- In real life pulsars have timing noise which means that older timing models need to be updated when new data is added.

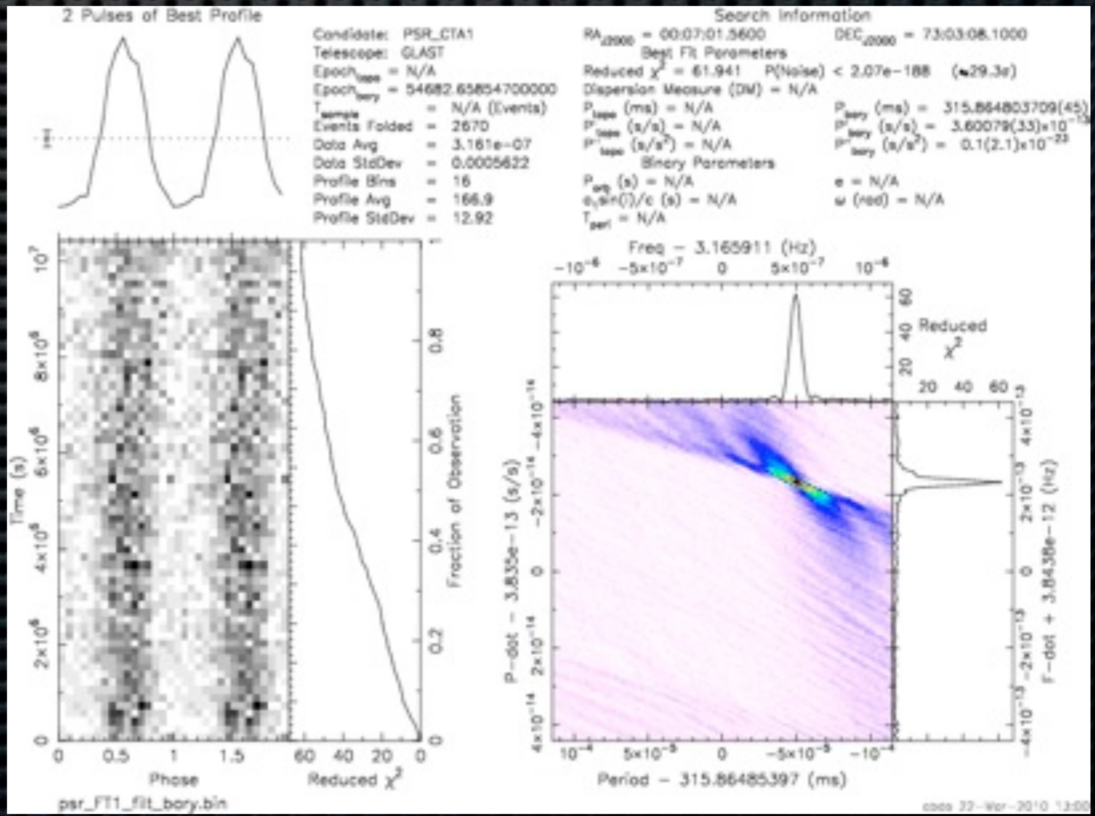
- This could be seed in the curving of the phaseogram as seen on the right.

Dealing with Timing Noise cont'd

- Notice that in the last slide the optimal reduced chi-square was outside the search window in the f-fdot space.
- One can expand the search window by using the ***-coarse*** option in prepfold.

```
$ prepfold -events -mjds -double -par psr.par psr_FT1_filt_bary.bin -
coarse
```

- The pulsar is detected now and the phaseogram looks more or less like a straight line.
- The next step will be to get a good timing model which covers the full time span. We will do this with Tempo2



5. Pulse Timing with Tempo2

- First we will use prepfold with the -timing option (in place of -par option) to generate folds using the best ephemeris from folding searches.

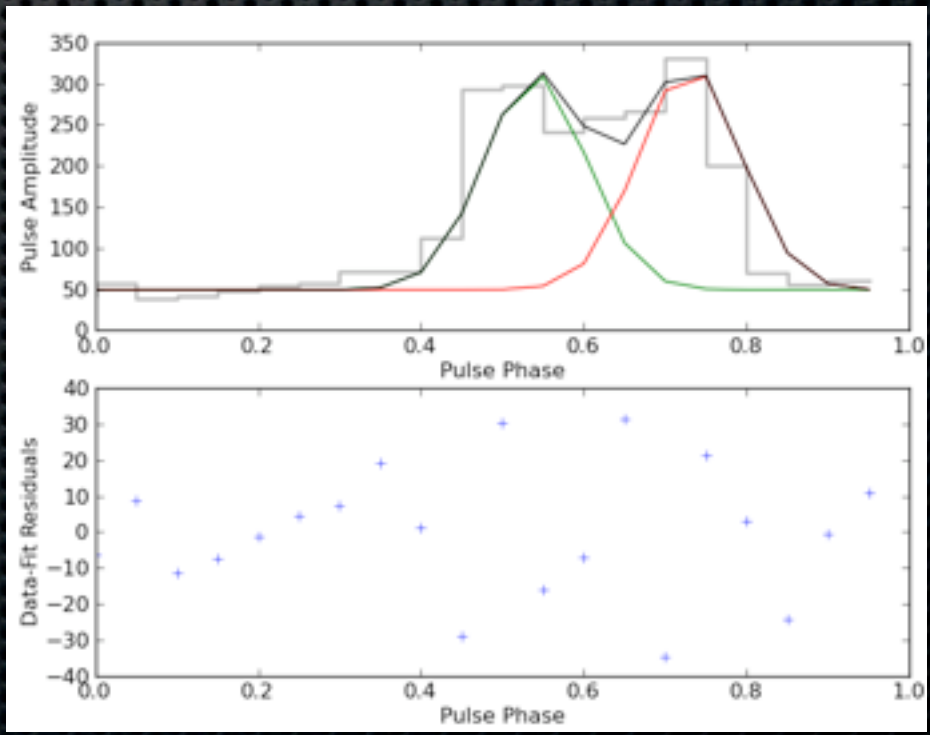
```
$ prepfold -events -mjds -double -timing psr.par psr_FT1_filt_bary.bin
```

- Next we will use the .bestprof file generated by prepfold to generate a template profile:

```
$pygaussfit.py psr_FT1_filt_bary_PSR_0007+7303.pfd.bestprof
```

Draw a rectangle around the regions to be fit by gaussian curves using a left-click and drag. After selecting the regions, a middle click will do the fitting. Output will look something like this:

```
[abdo@heselin tutorial]$ pygaussfit.py psr_FT1_filt_bary_PSR_0007+7303.pfd.bestprof
--------------------------------------------------------------------
Multi-Gaussian Fit by pygaussfit.py of 'psr_FT1_filt_bary_PSR_0007+7303.pfd.bestprof'
--------------------------------------------------------------------
mpfit status: 1
gaussians: 2
DOF: 13
chi-sq: 46.26
reduced chi-sq: 3.56
residuals  mean: -1.26e-07
residuals stdev: 17.6
--------------------------------------
 const = 49.41038 +/- 3.67581
 phas1 = 0.54045 +/- 0.00419
 fwhm1 = 0.14752 +/- 0.00972
 ampl1 = 41.17216 +/- 2.68273
 phas2 = 0.73043 +/- 0.00405
 fwhm2 = 0.14820 +/- 0.00941
 ampl2 = 42.91746 +/- 2.68894
--------------------------------------
[abdo@heselin tutorial]$ []
```



Take everything between the last two sets of "--------"'s and put it into a .gauss file:

Generate TOAs

- Now that we have a profile template we can use it to get TOAs.
- We will use get_TOAs.py from PRESTO to do this:

```
$get_TOAs.py -n 10 -e -g template.gauss psr_FT1_filt_bary_PSR_0007+7303.pfd >
psr_FT1_filt_bary_PSR_0007+7303.tim
```

Resulting TOAs

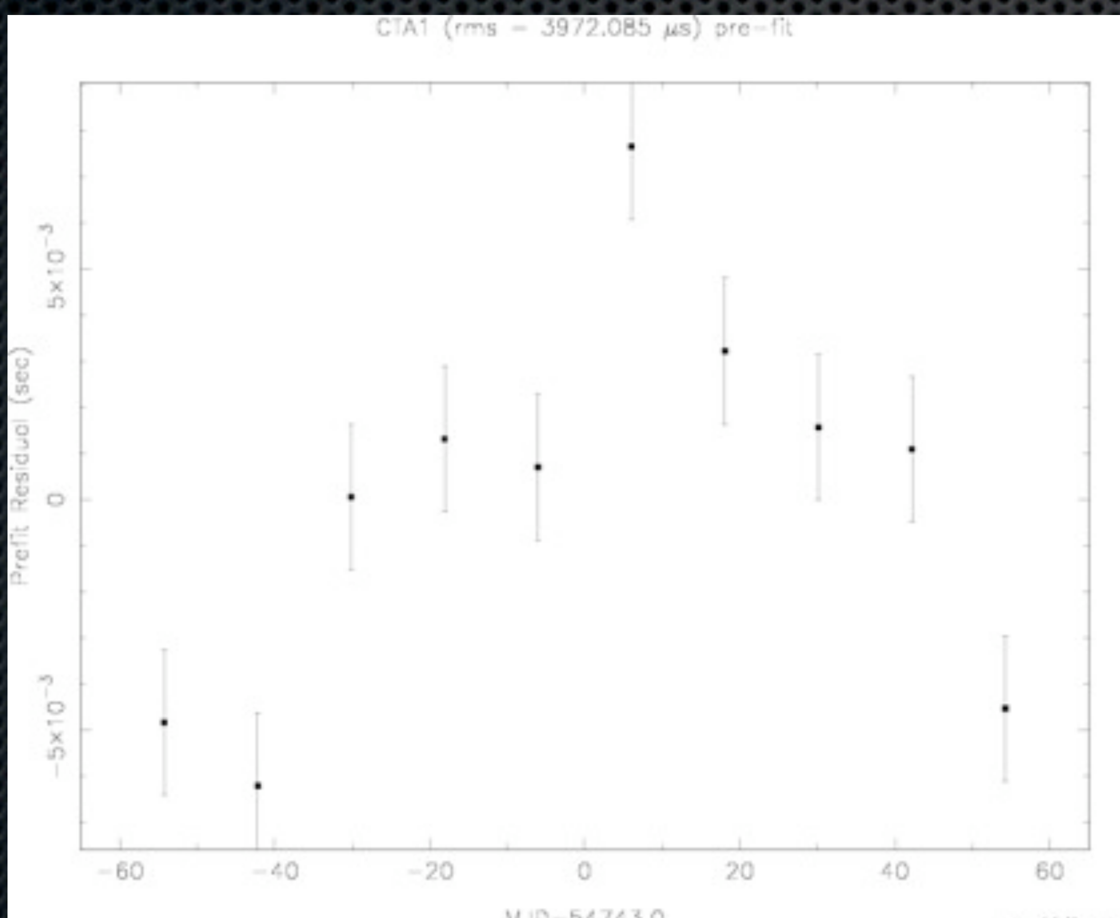
```
[abdo@heselin tutorial]$ more psr_FT1_filt_bary_PSR_0007+7303.tim
@                 0.000 54688.6896847794559   1579.32
@                 0.000 54700.7519617745390   1579.33
@                 0.000 54712.8142385821599   1579.33
@                 0.000 54724.8765187116424   1579.33
@                 0.000 54736.9387948875565   1579.33
@                 0.000 54749.0010745309989   1579.33
@                 0.000 54761.0633501105931   1579.34
@                 0.000 54773.1256291023027   1579.34
@                 0.000 54785.1879078316871   1579.34
@                 0.000 54797.2501825694527   1579.34
[abdo@heselin tutorial]$
```

Fit with Tempo2

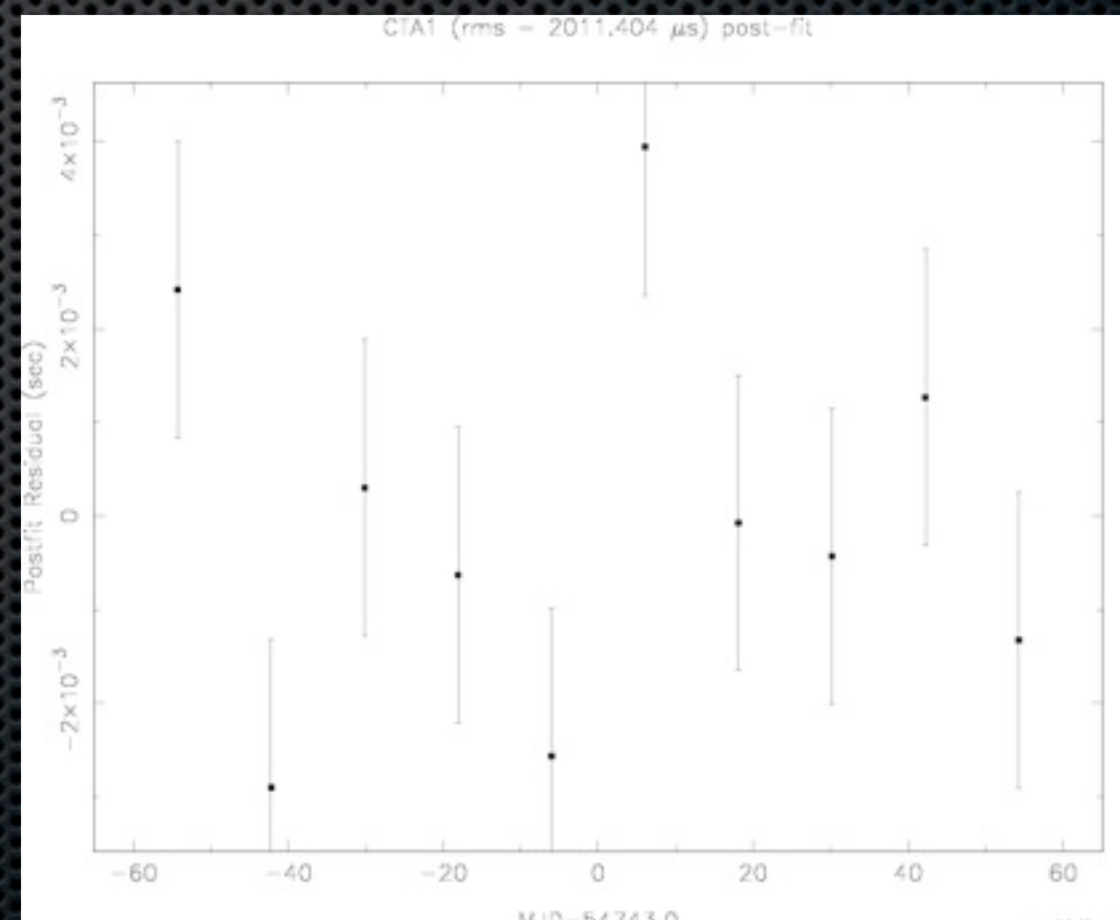
- We now use tempo2 to fit the TOAs and get a good timing model:

```
$ tempo2 -gr plk -f psr-old.par  psr_FT1_filt_bary_PSR_0007+7303.tim  -epoch
54850
```

Pre-fit residuals (RMS~4 ms)



Post-fit residuals (RMS ~ 2ms)



Best Fit Ephemeris

From Folding

PSRJ	CTA1
RAJ	00:07:01.56000000
DECJ	73:03:08.1000002
F0	3.165922646277616 1
F1	-3.8438458885488623e-12 1
PEPOCH	54647.441238999999999

From Timing

PSRJ	CTA1
RAJ	00:07:01.56000000
DECJ	73:03:08.1000002
F0	3.16592243157486899949 1 0.00000005909285465756
F1	-3.59294937710333818 7e-12 1 4.98714659500349121 94e-14
PEPOCH	54647.441238999999999

Note significant change in F1

Assigning Photon Phases

- Now that we have a timing model that we are happy with, we will need to assign phases for photons in our data file so we can do things like phase-resolved spectroscopy and phase-resolved images.
- Before we do that we will generate a new file with a larger ROI. This is better since we are usually interested in ROIs on the orders of degrees. We will rerun gtselect with a larger ROI, say 15 degrees first.

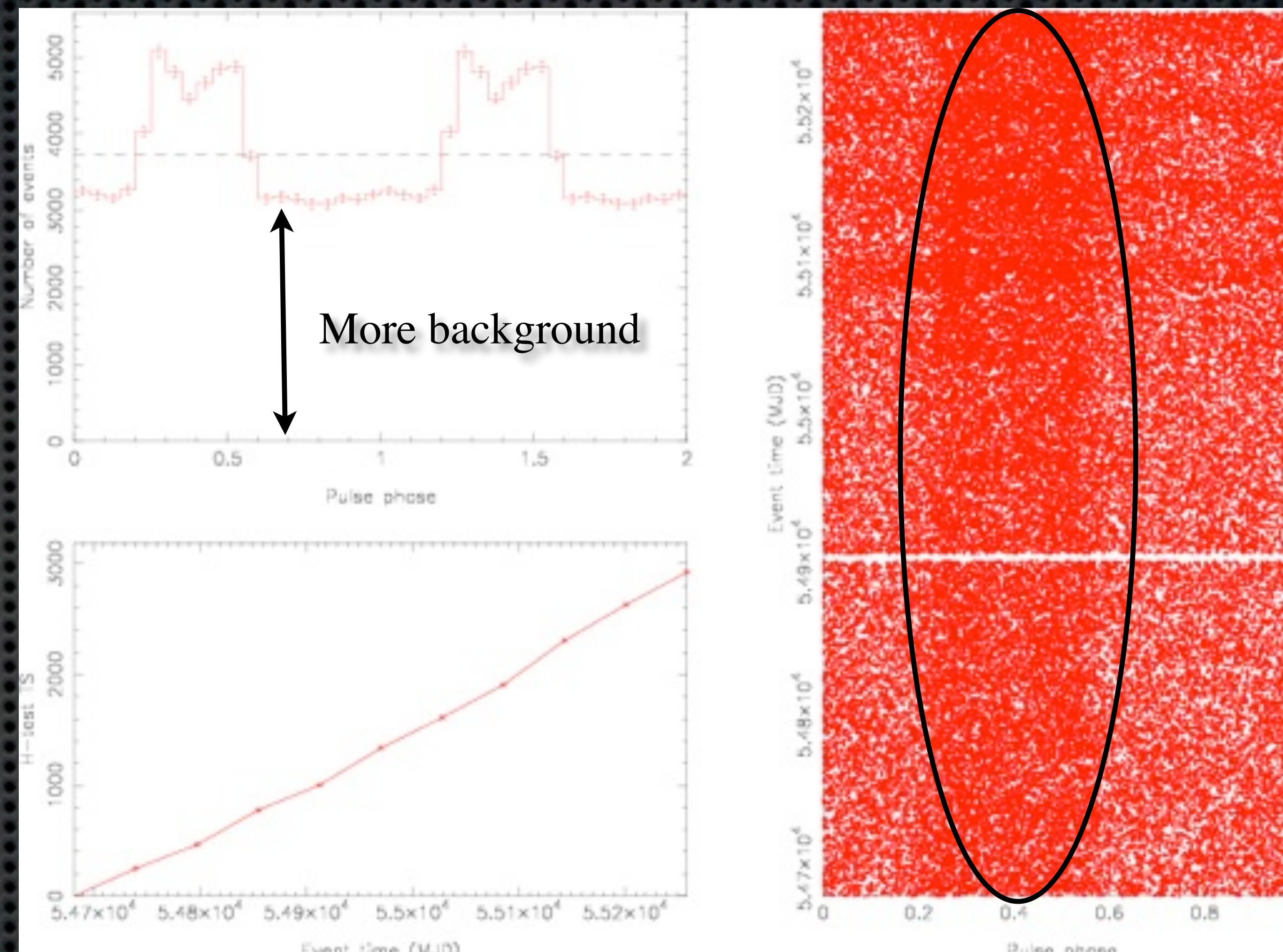
```
$ gtselect infile=FT1.fits outfile=psr_FT1_15Deg.fits ra=1.7565
dec=73.05225 rad=15.0 tmin=239557517 tmax=277689600 emin=100.0
emax=100000.0 evclsmin=3 zmax=105.0 clobber=yes
```

Assigning Photon Phases

- We will then use the Tempo2 plugin* developed by Lucas Guillemot with the new file to assign the photon phases.

```
$ tempo2 -gr fermi -ft1
psr_FT1_filt.fits  -ft2 FT2-
pro.fits -f psr-new.par -
phase
```

- This will add a PULSE_PHASE column to the fits file.
- The plugin prints out a graph of the pulsar light curve, a phaseogram, and the behavior of the h-test over time for the pulsar



Notice that the phaseogram is less obvious now. This is due to the large background of events that we got when we used a larger ROI (15 deg. compared to 1.5). This is also visible in the light curve figure on the upper left.

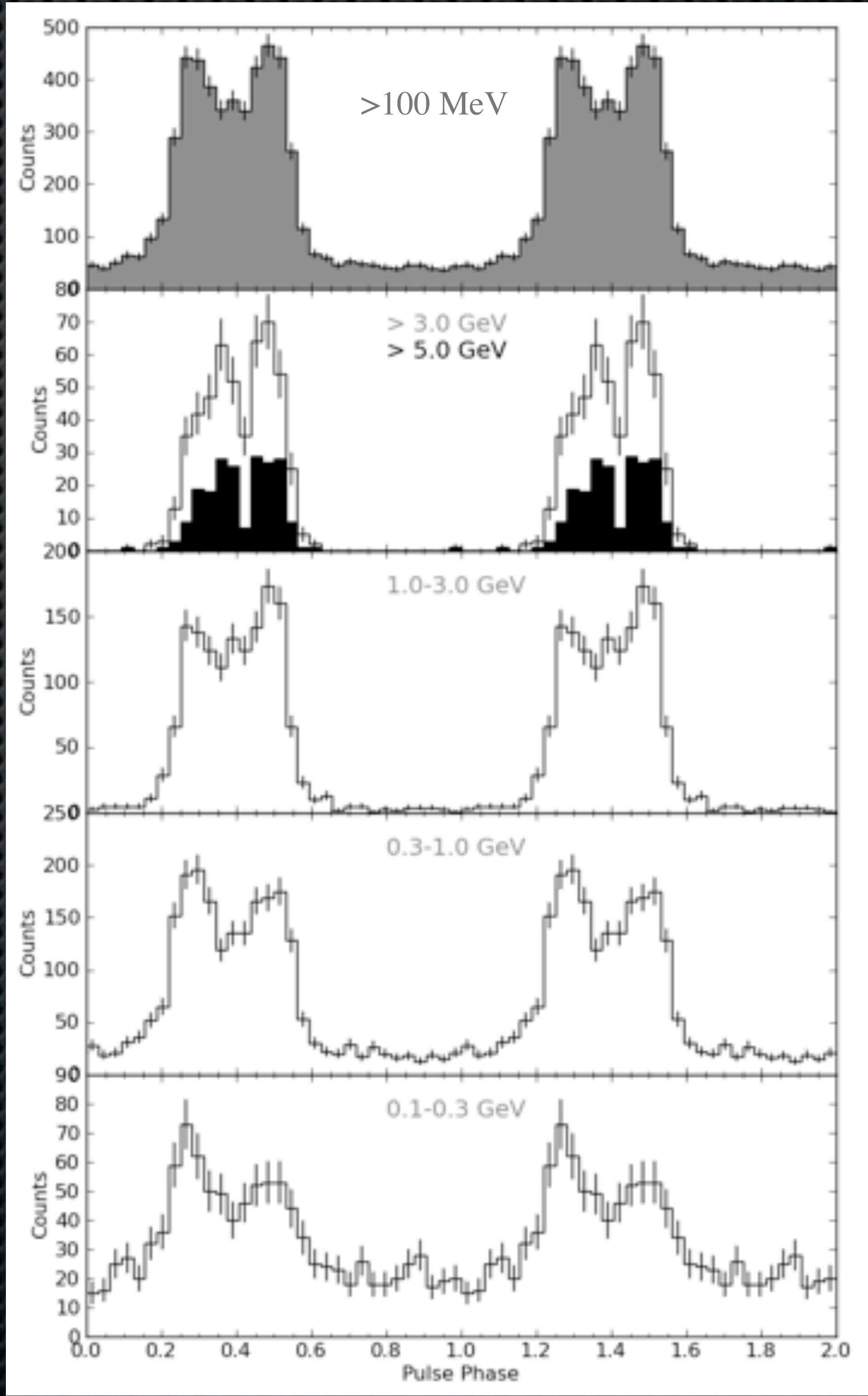
*A user contributed s/w available at: http://fermi.gsfc.nasa.gov/ssc/data/analysis/user/

Energy-Dependent Light Curves

- Now we have a 15 degree ROI file with phases assigned.
- We can make energy-dependent light curves of our pulsar using the user contributed script makelcs.py < http://fermi.gsfc.nasa.gov/ssc/data/analysis/user/>:

```
$makelcs.py psr_FT1_15Deg.fits
```

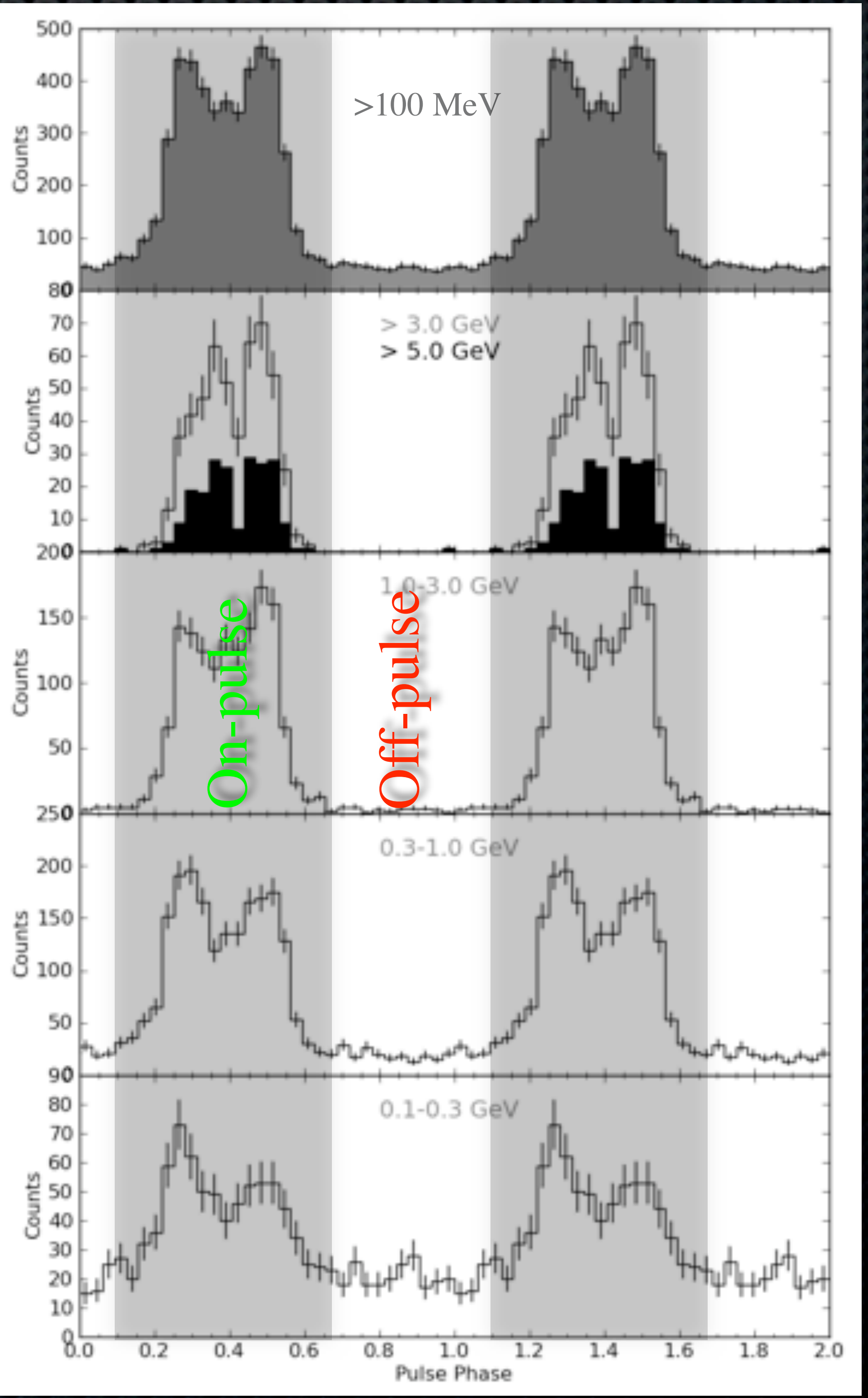
- This script creates pulse profiles for different energy bands as shown on the right.
- The script has many options like:
  - Number of bins in the pulse profile
  - Type of selection for the ROI (cookie cutter or energy-dependent cut)
  - Start and stop time cuts
  - Energy cuts



Phase Selection

- In our case we would like to look at the region of the sky around the pulsar when the pulsar is "on" and when the pulsar is "off"
- To do this we will look at the light curve and try to extract visually the definitions of these on- and off-pulse periods.
- Here we defined the on-pulse interval as [0.1-0.65] and the off-pulse interval to be its complementary
- Now we select the phase intervals using fselect:

```
$fselect infile=ft1.fits outfile=off-pulse-100MeV.fits
expr="( !(PULSE_PHASE .ge. 0.1 && PULSE_PHASE .le. 0.65))"
$fselect infile=ft1.fits outfile=on-pulse-100MeV.fits
expr="( (PULSE_PHASE .ge. 0.1 && PULSE_PHASE .le. 0.65))"
```

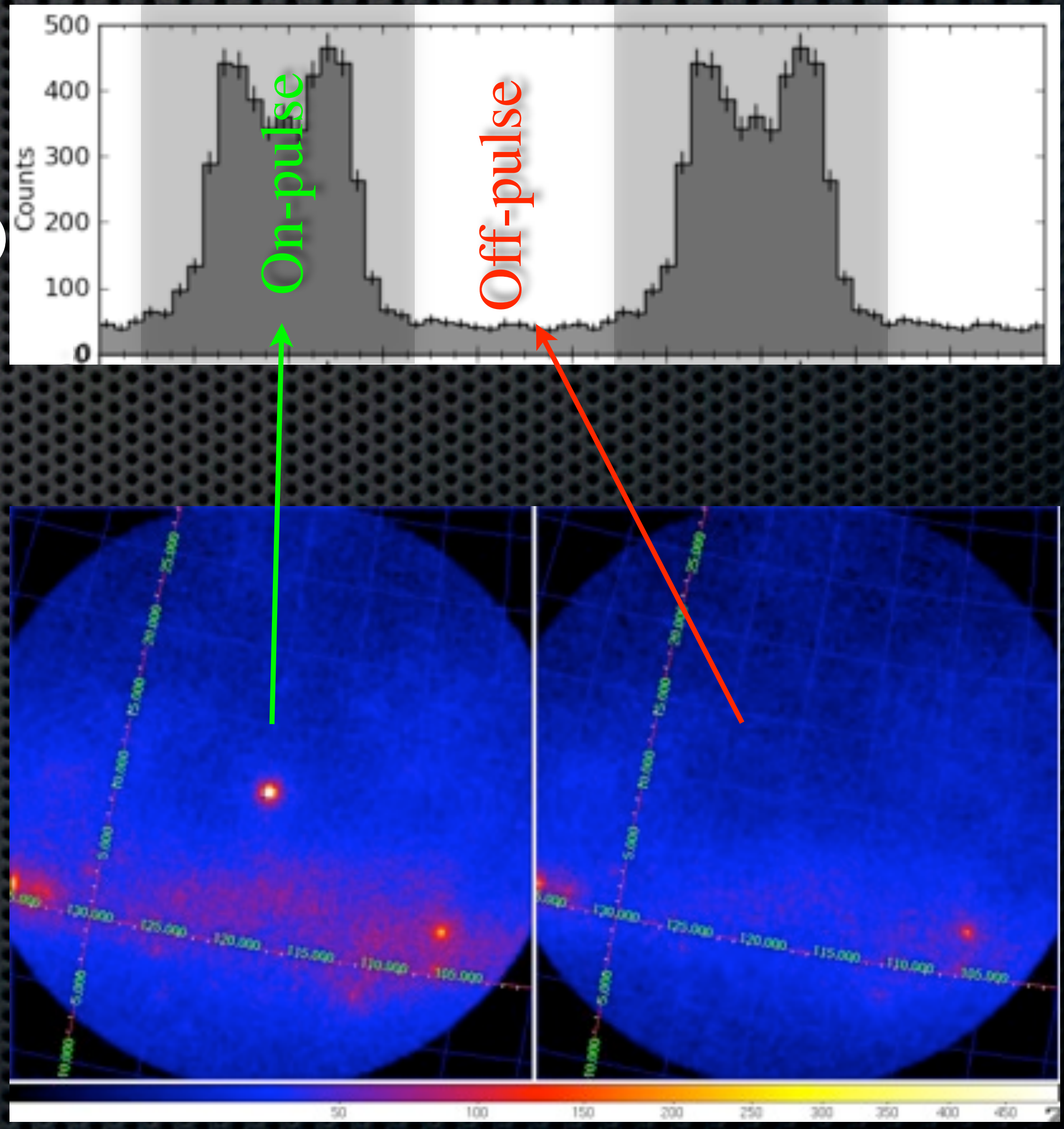


Phase Selection

- Now we have two fits files one that contains events which fell in the on-pulse part of the phase (on-pulse-100MeV.fits) and one with events which fell in the off-pulse part (off-pulse-100MeV.fits)
- We can use these two files to do things like phase-resolved spectroscopy (refer to talk on likelihood analysis)
- A good practice is to look at the counts maps for these two phase parts
- We will use gtbin again to generate these cmaps.

```
$ gtbin algorithm=CMAP evfile=on-pulse-100MeV.fits
scfile=FT2.fits outfile=on-pulse-100MeV-cmap.fits
nxpix=120 nypix=120 binsz=0.25 axisrot=0.0 coordsys=CEL
xref=1.7565 yref=73.05225 proj=AIT
```

- And we do the same for the off-pulse part



Phase-resolved counts maps.
Not exposure corrected

Correcting for Exposure

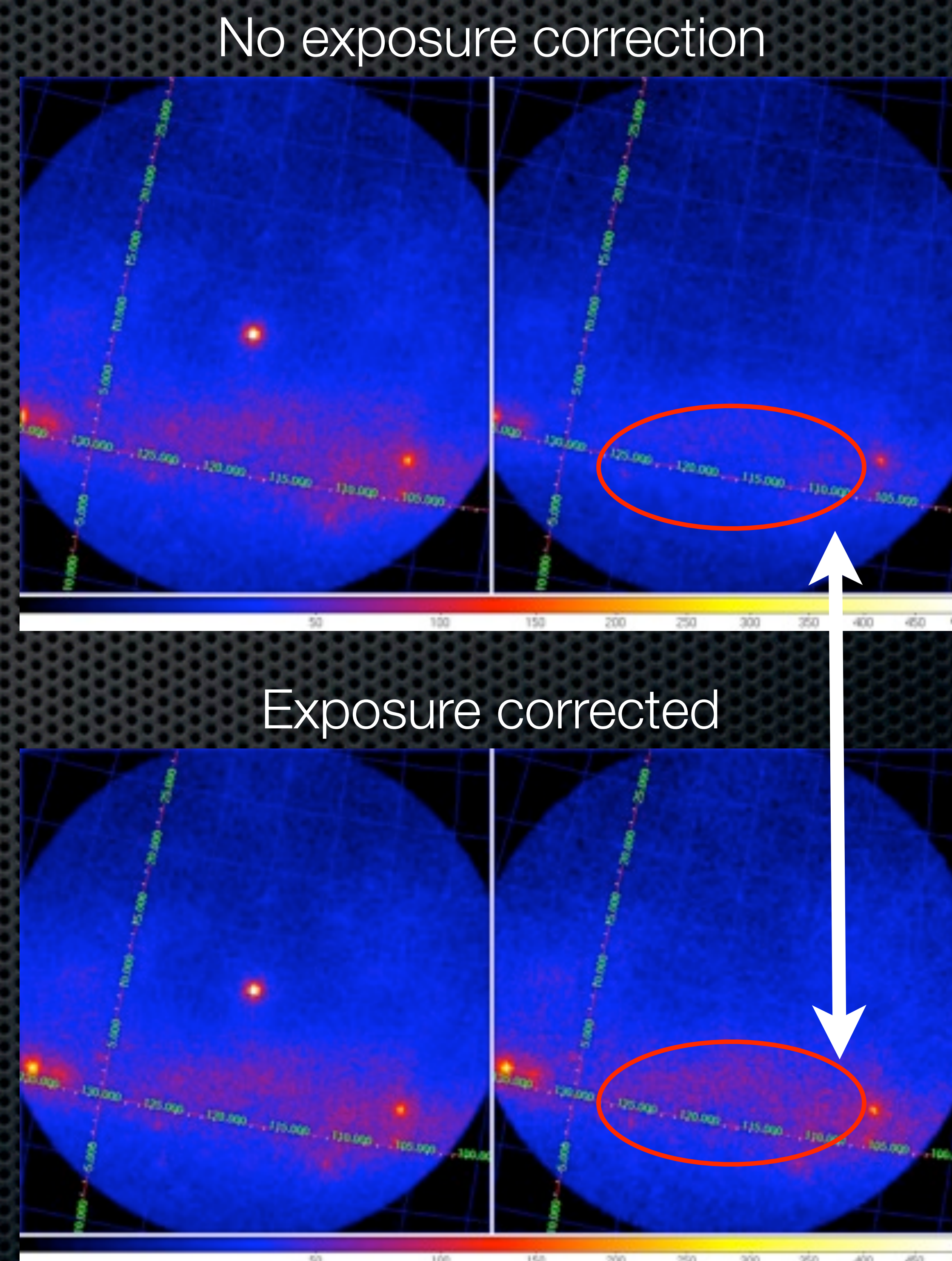
- To correct for the exposure in the counts maps we need to multiply the number of counts in each bin in the off-pulse map by a factor equal to the ratio of the on to off pulse phase interval widths:

$$Ratio = \frac{f_{on}}{f_{off}}$$

- In our case R is equal to 0.55/0.45 = 1.22

- This scaling can be done with the farith tool:

```
$farith off-pulse-100MeV-cmap.fits 1.22 off-
pulse-100MeV-cmap-scaled.fits *
```



Correcting for Exposure

- In the case of performing a likelihood analysis on phase selected data one needs to take into account the phase selection:
  - Correct for the exposure in the model xml file. This included correcting some spectral parameters for all the sources in the model. For example, multiplying the “integral” parameter for sources modeled with power law by the correct ratio.
  - Regenerating exposure maps with gtexpmap

- Now we do examples ....