3. Creating a tempo2 plugin to search for gravitational waves

Aims and objectives

- Learn how to create a tempo2 plugin to process multiple pulsar data sets
- Analyse pulsar timing residuals using various statistical tests
- Develop your knowledge of the programming languages C or C++ (the basis of many pulsar software packages)
- Access the IPTA data challenge

Background reading

- TEMPO2 documentation online
- TEMPO2 tutorial on making plugins
- Use your web browser to find sites to help you to program in C and C++
- The IPTA data challenge is available from http://www.ipta4gw.org/?page_id=214
- Hellings & Downs (1983; ApJ) "Upper limits on the isotropic gravitational radiation background from pulsar timing analysis"
- Hobbs, G et al. (2009; MNRAS) "TEMPO2: a new pulsar timing package - III. Gravitational wave simulation"
- Jenet et al. (2005; ApJ) "Detecting the Stochastic Gravitational Wave Background Using Pulsar Timing"

Who to find if you get stuck during the student week

If you have trouble writing the plugin contact: David Champion, Nathan Garver-Daniels, George Hobbs, Mike Keith, Ryan Shannon

If you have trouble with the statistical analysis: Fredrick Jenet, Yuri Levin, Duncan Lorimer, Dick Manchester, Maura McLaughlin, Ingrid Stairs, Dan Stinebring

Experts to discuss these topics with during the science week

The following people will be at the science workshop who have developed tempo2 plugins: Sarah Buchner, Sarah Burke Spolaor, David Champion, Xinping Deng, Justin Ellis, George Hobbs, Gemma Janssen, Michael Keith, Vikram Ravi, Ryan Shannon, Rutger van Haasteren, Joris Verbiest, Jingbo Wang, Xiaopeng You

The following people have experience in parameterising data sets and searching for gravitational waves: Matthew Bailes, William Coles, Jim Cordes, Paul Demorest, Fredrick Jenet, Kejia Lee, Yuri Levin, Andrea Lommen, Dick Manchester, Scott Ransom and Ryan Shannon.

The IPTA data challenge has been created by Fredrick Jenet, Michael Keith and Kejia Lee.

1 Creating and installing your first plugin

The purpose of this worksheet is to analyse the data sets made available as part of the IPTA data challenge. To do this, we will first write a simple TEMPO2 plugin, obtain the data and then update the plugin as required.

Usually you will not need to modify the standard TEMPO2 software. However, you may find that TEMPO2 does not provide you with all the functionality that you need for your project. In such cases you will need to write a TEMPO2 plugin. Such plugins can be written in C or C++.

To start you should find the tempo2 source code on your laptop and go into the "plugin" directory. Then you must open a text editor (emacs/vim etc.) and type in the following template (if you are feeling lazy the code is also available from /data/module3/code/mycode_plug.C, but make sure that you understand what each line in the code is for. If you do this, please explain what each function is doing by adding more detailed comments.):

```
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#include <math.h>
#include "tempo2.h"

using namespace std;

extern "C" int graphicalInterface(int argc,char *argv[],pulsar *psr,int *npsr)
{
 char parFile[MAX_PSR][MAX_FILELEN];
 char timFile[MAX_PSR][MAX_FILELEN];
 int i;
 double globalParameter;

 *npsr = 0;  // Number of pulsars to process

 // Check the command line arguments
 for (i=1;i<argc;i++)
  {
   if (strcmp(argv[i],"-f")==0)
   {
    strcpy(parFile[*npsr],argv[++i]);
    strcpy(timFile[*npsr],argv[++i]);
    (*npsr)++;
   }
  }

 readParfile(psr,parFile,timFile,*npsr);  // Read the timing models
 readTimfile(psr,timFile,*npsr); // Read the arrival times
 preProcess(psr,*npsr,argc,argv);

 // Now form the barycentric arrival times, residuals, do the fit and print the output
 for (i=0;i<2;i++)
  {
   formBatsAll(psr,*npsr);
   formResiduals(psr,*npsr,1);
   if (i==0) doFit(psr,*npsr,0);
   else textOutput(psr,*npsr,globalParameter,0,0,0,(char *)"");
  }

 return 0;
```

```
}
```

Save this file with a filename such as "mycode_plug.C". You must now compile and install this plugin. To do this type:

```
$ g++ -fPIC -shared  -l tempo2 -o $TEMPO2/plugins/mycode_{$LOGIN_ARCH}_plug.t2 mycode_plug.C
```

(If you are using your own computer, you will need to set `$LOGIN_ARCH` appropriately.

To check whether it has been installed correctly type in:

```
$ tempo2 -h
```

and see if your plugin is listed. If it is not listed then request help from George Hobbs, Ryan Shannon, Mike Keith or David Champion.

2 Getting some data files and running your plugin

For this worksheet we will make use of the IPTA data challenge. We want to process the first data set downloaded from `http://www.ipta4gw.org/?page_id=126`. These data can also be obtained from /data/module3/datachallenge/.

To start just view the timing residuals for a few pulsars, e.g.,

```
$ tempo2 -gr plk -f J0030+0451.par J0030+0451.tim
```

You can run your plugin with

```
$ tempo2 -gr mycode -f J0030+0451.par J0030+0451.tim -f J0218+4232.par J0218+4232.tim
  -f J0437-4715.par J0437-4715.tim ... -npsr 40 -nobs 1000 -fit f0 -fit f1
```

(Use the -h option to understand the -npsr and -nobs command line arguments.)

You will also need to update the observatory.dat (located in the directory $TEMPO2/observatory/) file to include the fake telescope by adding the following line:

```
6378138.00 0.0 0.0  AXIS axi
```

What do the columns in the observatory.dat file represent? (Challenge: Why did the data challenge creators use a fake telescope at these particular coordinates?)

What is a simple way to stop you needing to type in 36 pulsar names?

List a few ways in which the data challenge data sets differ from actual observations

---

---

3 Updating the plugin

The plugin written above will only carry out some basic parameter fits for each pulsar and then display the resulting parameter values to the screen. For this worksheet you need to be able to carry out mathematical calculations on the resulting residuals. Try adding the following near the end of your plugin:

```
double sum;
int p;

for (i=0;i<*npsr;i++)
 {
  printf("The name of pulsar %d is %s\n",i,psr[i].name);
 }

 for (p=0;p<*npsr;p++)
 {
  printf("Pulsar %d has %d observations\n",p,psr[p].nobs);
  sum=0.0;
  for (i=0;i<psr[p].nobs;i++)
   {
     printf("The site-arrival-time  for pulsar %d, observation %d is %g\n",
            p,i,(double)psr[p].obsn[i].sat);
     printf("The post-fit timing residual is %g\n",(double)psr[p].obsn[i].residual);
     sum = sum + (double)psr[p].obsn[i].residual;
   }
   printf("The summation of the timing residuals for pulsar %d is %g\n",p,sum);
 }
```

You should now save that file, compile it and run it again. Now you should get significantly more information about the pulsar and each observation. Make sure that you understand how the code above works before moving on to the next section.

How would you print the 20th timing residual for the 5th pulsar?

____
____
____

Look at the source code or tutorials to determine how you would print the observing frequency for the 5th observation of the first pulsar?

____
____
____

4 Parameterising the data sets

Update your plugin to produce the following for each pulsar:

- The number of observations
- The data span
- The time between observations
- The mean of the post-fit timing residuals
- The standard deviation of the post-fit timing residuals

One of the main goals of the IPTA project is to make the first direct detection of gravitational waves. For a gravitational wave background it is expected that this detection will be made by measuring how correlated the pulsar timing residuals are between different pulsar pairs and comparing those correlations with the expected functional form that was derived by Hellings & Downs (1983).

Can you update your plugin to calculate the correlation coefficient between the timing residuals for each pulsar pair?

To see the expected functional form that was first derived by Hellings & Downs (1983) you will need to plot these correlation coefficients as a function of the angle between each pulsar pairs. To get the position of any pulsar you can use

```
double raj_in_radians = (double)psr[p].param[param_raj].val[0];
double decj_in_radians = (double)psr[p].param[param_decj].val[0];
```

There are quite a few ways to find the angle between any two objects in the sky. For one approach you can use:

```
double psrangle(double ra1,double dec1,double ra2,double dec2)
{
 double dlon,dlat,a,c;
 double deg2rad = M_PI/180.0;

 dlon = (ra2-ra1);
 dlat = (dec2-dec1);
 a = pow(sin(dlat/2.0),2) + cos(dec1)*cos(dec2)*pow(sin(dlon/2.0),2);
 if (a==1) c= M_PI;
 else c = 2.0*atan2(sqrt(a),sqrt(1.0-a));

 return c/deg2rad;
}
```

Can you plot the correlation coefficients as a function of angle between the pulsars?

Can you detect the Hellings & Downs (1983) expected functional form?

Will your algorithm work if the data were not regularly spaced?

It's possible that you will be able to obtain a low-significance detection of the Hellings-Downs functional form using the above analysis. However, there are lots of ways to improve your analysis. Speak with one of the experts to discuss ways in which your code could be improved (Note that the basic problem is that the gravitational wave background has a steep spectral shape and this leads to difficulties when you calculate the correlation coefficients – to consider this in more detail try the challenge below).

5 Challenge 1 - this is not easy!

What is the expected shape of the spectrum of the timing residuals induced by a gravitational wave background?

The challenge is to obtain the average power spectrum for all the timing residuals. You can do this by writing your own spectral estimation code, looking at the TKspectrum routines in tempo2 (see the TKspectrum.h functions with the main tempo2 source code) or using tempo2 to write out the post-fit residuals and then loading the answers into your favourite data analysis software package (such as MATLAB or IDL).

Is it valid to measure a power spectrum for post-fit timing residuals?

Does the resulting averaged power spectrum have the expected spectral exponent?

What is the amplitude of the gravitational wave background in these data?

Would your spectral estimation code work if the observations were not regularly sampled?

Would your spectral estimation code work if each pulsar had a different white noise level?

6 Publication

Find a publication on ADS that is related to this worksheet and summarise the basic conclusions of that paper.

7 Mistakes

How many mistakes did you find in this worksheet?