ExoFit User's Guide

Sreekumar Thaithara Balan & Ofer Lahav

2008-05-21

1 Introduction

ExoFit is a software for extracting orbital parameters of extra-solar planets from radial velocity data. It uses Markov Chain Monte Carlo(MCMC) method to estimate orbital parametes and their uncertainties. Radial velocity model and MCMC method is discussed in [1].

2 Installing *ExoFit*

- **Step 1:** Download the tar ball ExoFit.v1.tar.gz from

  http://zuserver2.star.ucl.ac.uk/~lahav/exofit.html

- **Step 2:** Go to directory where ExoFit.v1.tar.gz is downloaded and in the terminal type:

  ```
  tar xvzf ExoFit.v1.tar.gz
  ```

  This will extract all the source files into a directory named Exofit.vX.XX.

- **Step 3:** *ExoFit* needs GSL - GNU Scientific Library. In a typical installation of GSL, GSL header files are located in a directory named `gsl` under `/usr/include/` and the `lib` files are located in `/usr/lib/`. If you have installed gsl somewhere else please specify the path to `gsl` in `Makefile`. The first two lines of the `Makefile` should be edited to specity `gsl` header files and libraries. For example if you have installed GSL in `/home/visitor/usr/local/`, the first two lines should be modified to:

  ```
  INCDIR=/home/visitor/usr/local/include
  LIBDIR=/home/visitor/usr/local/lib
  ```

  Now you can complie the source files.

- **Step 4:** We assume that you have GCC,the GNU Compiler Collection. To complie and make the executable type:

  ```
  make
  ```

This will create two executable files *exofit* and *plotmaker*. Now we are ready to go. You may put these executable files in `~/bin` or any other directory in your `PATH` for ease of use.

3 Using *ExoFit*

To use exofit type[1]

`exofit` *path/rvdata.dat*

Where *rvdata.dat* is the file containing radial velocity data and *path* is the path to the file *rvdata.dat*. For example if your radial velocity data is in */home/data/HD187085.dat* the you should type:

```
exofit /home/data/HD187085.dat
```

4 Input Data

The input to *exofit* is the radial velocity data and the state data.

4.1 Radial Velocity Data

JD (-2451000)	RV (ms^{-1})	Uncertainty (ms^{-1})
120.9170	-12.1	5.3
411.0753	-2.5	6.5
683.1693	16.1	5.7
743.0494	9.3	5.5
767.0046	8.8	4.7
769.0652	5.3	4.4
770.1153	5.3	5.2
855.9477	8.6	8.9
1061.2140	-4.5	5.3

Table 1: Form of the radial velocity data. The entries shown here are from [2] for HD187085.

Radial velocity data is a simple text file with format shown in Table[1]. The data has 3 columns. Column 1 is the time coordinate, Column 2 is the radial velocity in ms^{-1} and Column 3 is the uncertainty in measurement also in ms^{-1}.

[1] We assume that `exofit` is in your `PATH`. Otherwise specify the full path to `exofit` like `/home/visitor/programs/exofit`. You can copy `exofit` and `plotmaker` to the present working directory and type `./exofit /home/data/HD187085.dat`

4.2 State Data

The state data *state.dat* defines the starting points of the Markov Chain and the prior boundaries. If the file *state.dat* is present in the same directory as the *exofit*, then the values from *state.dat* is taken as input parameters. If it is not found then *exofit* runs with default state values which are identical to the values shown in the table.If you are not familiar with MCMC do not keep this file in the same directory as *exofit*. *The program works fine even if this file is not present.* Each column in *state.dat* has the following format:

Parameter
Minimum
Start Value
Maximum
Step Size

T	K	V	e	w	X	s
0.2	0.00001	-2000.0	0.0	3.0	0.0	0.0
7500.0	1000.0	0.0	0.50	4.145	0.5	1000.0
15000.0	2000.0	2000.0	0.99999	6.28318	0.99999	2000.0
1500.0	200.0	400.0	0.1	0.628	0.1	200.0

Table 2: Form of the text file *state.dat*, were T, K, V, e, w, X, s stads for *period* in days, *amplitude*, in ms^{-1}, *systematic velocity* in ms^{-1}, *eccentricity*, *longitude of periastron* in radians, *periastron passage factor* and *noise factor* in ms^{-1} resprctively.

A sample state data is shown in Table[2]. The fist row in the data file is called a *header* and it contains the names of each parameter in the state. The second row contains the minimum values of each parameter, third row contains the starting points of MCMC[2], the forth row specifies the maximum values of each parameter and the fifth row defines the step sizes parameters. Each column should be separated by white spaces.

5 Output

The output of *exofit* is again a text file called *extract.dat*. This file, as shown in Table[3] has a header which contains the names of the parameters in the state and current strength of the state (G in Table[3]). After the header each row represents the state of the parameters (in the same order as in the header) at each iteration in the MCMC and the strength at that iteration. We call this an MCMC extract. The program also produces two other files namely *burn.dat* and *diag.dat*. *burn.dat* has the same format as the MCMC extract. However these values are considered as burn in and should not be used for the calculation of densities. *diag.dat* has again the same format, but shows the mean and standard deviation of each parameter for every 10000 iterations. These files shows the progress of the Markov Chain towards its stationary distribution. *extract.dat*

[2] Starting values of MCMC should be between minimum and maximum vales of the parameter.

is the final output of *ExoFit*. You may use your own statistical visualisation programs to analyse *extract.dat* or make use of the scripts provided below.

T	K	V	e	w	X	s	G
1037.084	14.368	-2.0020	0.27606	0.40457	0.06613	5.3891	-162.11
1003.861	15.571	-0.9926	0.18873	0.04108	0.10656	4.9867	-162.20
1080.522	15.104	-0.7553	0.32822	0.30887	0.15516	5.2289	-160.85
1005.389	17.591	0.33853	0.29412	0.22377	0.03030	4.6077	-165.58

Table 3: A sample MCMC extract

6 Statistics and Visualisation

6.1 Basics

The output of the *exofit* is used to estimate the orbital parameters and their uncertainties. This can be done with the help of any statistical packages available. We used R to find out the statistical summaries and produce their visualisation. R is a robust environment for statistical computation which is freely available. It can be obtained from http://www.r-project.org/. We have provied an R script that calculates the statistical summaries, plots the posterior densities of each parameter and finlay make the radial velocity curve using median of samples from the MCMC extract. This file could be found under the directory `scripts`. The procedure is explained below. We assume that your machine has R installed on it.

- **Step 1:** Start an R session. Open a terminal session and type

  ```
  R
  ```

  This will start a new R session. You should get an output as shown in Figure[1]. To make the density plots and errorbars, the following packages

-173.1464

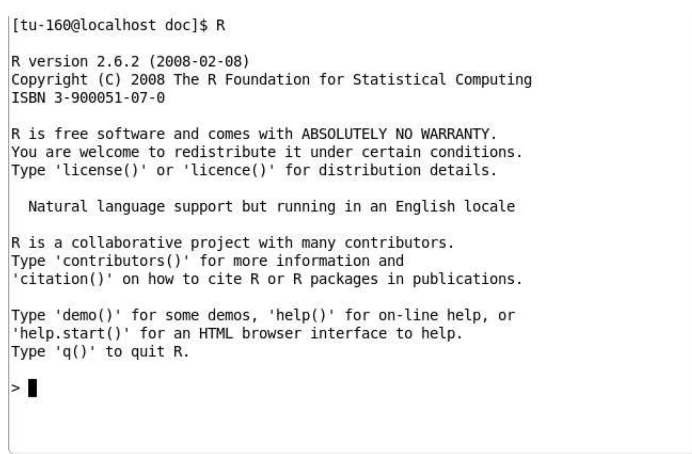


Figure 1: R session

must be present in R.

1. **lattice**
2. **gird**
3. **MASS**
4. **Hmisc**
5. **coda**
6. **hdrcde**
7. **ash**

To install these packages use the command:

```
available.packages()
```

This will open up a dialogue box as shown in Figure[2]. Choose an appropriate mirror to download the packages from. To install a packages

Figure 2: CRAN mirrors

use the command *install.packages('package')*. For example, to install the package *coda* type:

```
install.packages('coda')
```

To install some of the packages you might need the R header files. You can get help for a particular command with *help()*. For example to get help for *install.packages()* type:

```
help(install.packages)
```

You may also use the documentation available at the CRAN page at R Installation and Administration. To quit an R session use the command:

```
q()
```

- **Step 2:** If you are running *exofit* and *plotmaker* from the present working directory, load the R script file onto the current R session by typing:

  `source(‘‘path/orbit_plot.R’’)`

  Where *path* refers to the path to *orbit_plot.R*. This scripts loads the data from *extract.dat* and calculates the statistical summaries. You need to have *extract.dat* and the executable `plotmaker` in the present working directory for this R script to work! Otherwise you need to specify the `PATH` to *plotmaker* in the R script *orbit_plot.R* by modifying the line `system('./plotmaker')` to `system('PATH/plotmaker')` where `PATH` referres to the path to *plotmaker*.

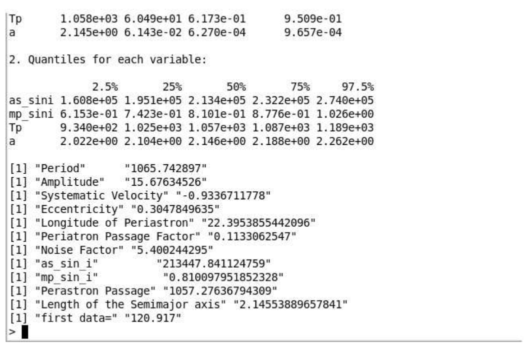
- **Step 3:** Display the estimates of orbital parameters. Type:

  `print.median(‘‘RVdata’’,mass_of_the_star)`

  Where "*RVdata*″ is the radial velocity data and *mass_of_the_star* referes to mass of the parent star in units of mass of the Sun. For example if your data file is `HD187085.dat` and you assume that the mass of HD187085 is 1.16 $M\odot$ then type.

  `print.median(‘‘HD187085.dat’’,1.16)`

  This should produce an out as shown in the Figure[3]. The values displayed are statistical summaries of orbital parameters[3] and the medians of all the samples from *extract.dat* and the medians of other useful quantities such as $m_p sin\, i$.



```
Tp       1.058e+03 6.049e+01 6.173e-01      9.509e-01
a        2.145e+00 6.143e-02 6.270e-04      9.657e-04

2. Quantiles for each variable:

              2.5%       25%       50%       75%     97.5%
as_sini 1.608e+05 1.951e+05 2.134e+05 2.322e+05 2.740e+05
mp_sini 6.153e-01 7.423e-01 8.101e-01 8.776e-01 1.026e+00
Tp      9.340e+02 1.025e+03 1.057e+03 1.087e+03 1.189e+03
a       2.022e+00 2.104e+00 2.146e+00 2.188e+00 2.262e+00

[1] "Period"      "1065.742897"
[1] "Amplitude"   "15.67634526"
[1] "Systematic Velocity" "-0.9336711778"
[1] "Eccentricity" "0.3047849635"
[1] "Longitude of Periastron" "22.3953855442096"
[1] "Periatron Passage Factor" "0.1133062547"
[1] "Noise Factor" "5.400244295"
[1] "as_sin_i"         "213447.841124759"
[1] "mp_sin_i"          "0.810097951852328"
[1] "Perastron Passage" "1057.27636794309"
[1] "Length of the Semimajor axis" "2.14553889657841"
[1] "first data=" "120.917"
>
```

Figure 3: Median value of parameter estimated from samples

- **Step 4:** Display the trace plots of parameters in the radial velocity model:

  `trace.model()`

  This command plots the trace plots of all parameters in the MCMC extract as shown in the Figure[4].

- **Step 5:** Plot the histograms and densities of parameters in the model. Type:

[3] You may have to scroll up a bit to see all of them

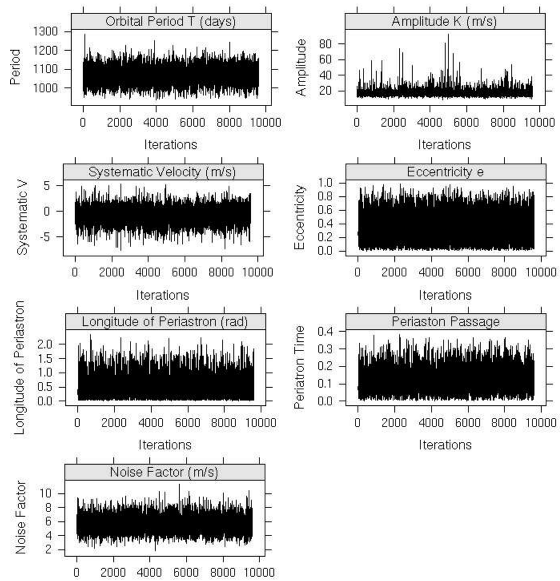


Figure 4: Trace plot for radial velocity data of HD187085

```
hist.dens.model()
```

The output is shown in Figure[5].

- **Step 6** Plot the histograms with densities of other useful parameters as shown in the Figure[6] using *hist.dens.other("RVdata",mass_of_the_star)* . This command needs the name of the radial velocity data as input. For example

```
hist.dens.other(''HD187085'',1.16)
```

- **Step 5** Finally plot the radial velocity curve along with the radial velocity data. Again you need to specify the radial velocity data file

```
orbit.plot(''HD187085.dat'')
```

The output is shown in Figure[7]

6.2 Convergence Tests

Convergence of Markov Chain is an important aspect of any MCMC method. It is very difficult to tell whether a chain has converged to its stationary distribution. On the other hand it might be reasonably easy to tell whether a chain has not converged. There are packages in R which deals with convergence diagnostics of Markov Chains. These packages offers a variety of tests to check the convergence of Markov Chains. We use *coda* for our analysis. It has an interactive menu to navigate between various tests and other statistical tools available in the package.

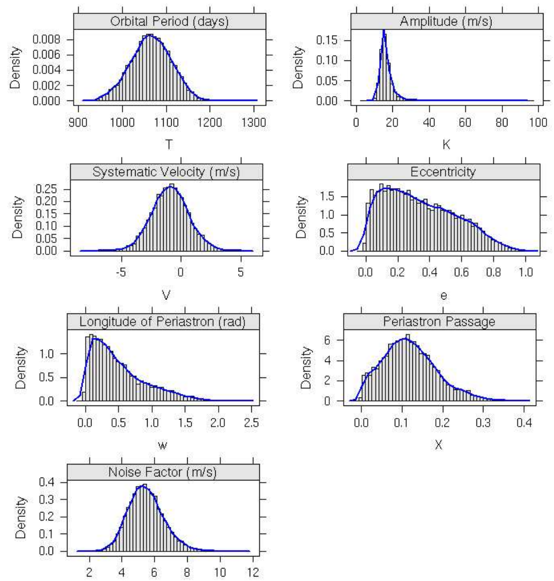


Figure 5: Histogram and density plots for HD187085

In order to analyse the Markove Chain using *coda* one need to create an MCMC object. The R script creates an object called *mc*. This MCMC object corresponds to parameters in the radial velocity model. They are the input to *coda*. To start *coda* in R session type:

```
codamenu()
```

This will start a coda session which looks like Figure[8]. Choose the option 2 in the menu and enter *mc* as the input object. *coda* will check for the *effective step size* first. If it is fine you may proceed further to look at the traceplots and convergence diagnostics.

7 Changing Priors

Changing priors involvs re-compiling *ExoFit*. The prior densities on each parameter are defined in the source file `ExoFit.vX.XX/src/mcmc.cc` as shown below.

```
//add bonds
mcstate.add_bond(period,&jeffreys);
mcstate.add_bond(amplitude,&mod_jeff);
mcstate.add_bond(sys_velocity,&uniform);
mcstate.add_bond(eccentricity,&uniform);
mcstate.add_bond(long_periastron,&uniform);
mcstate.add_bond(periastron_pass,&uniform);
mcstate.add_bond(noise_factor,&mod_jeff);
```

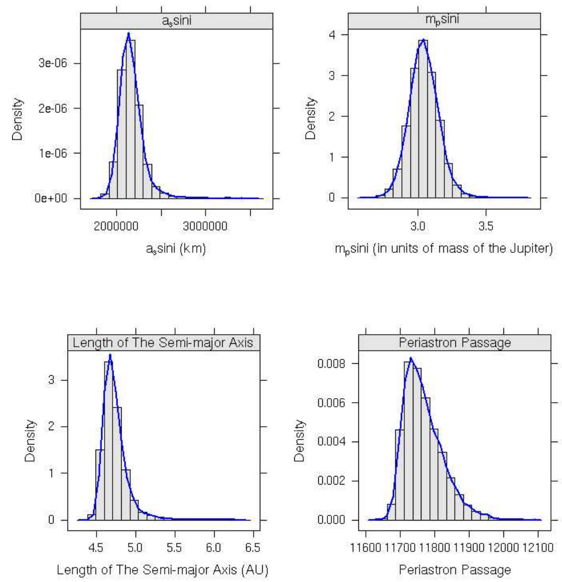


Figure 6: Other useful astronomical quantities for HD187085

Where `jeffreys` represents a Jeffreys prior, `mod_jeff` represents a modified Jeffreys prior with a break at 1.0 and `uniform` represents a uniform prior distribution. For example to change the prior distribution of parameter `period` from `jeffreys` to `uniform`, simply modify the corresponding line to

```
mcstate.add_bond(period,&uniform);
```

Then make the executable binaries usig the command `make` as explained in the Section[2].

References

[1] S. T Balan and O. Lahav. *ExoFit*: Orbital parameters of extra-solar planets from radial velocities. *MNRAS*, 2008.

[2] H. R. A. Jones, R. P. Butler, C. G. Tinney, G. W. Marcy, B. D. Carter, A. J. Penny, C. McCarthy, and J. Bailey. High-eccentricity planets from the anglo-australian planet search. *Monthly Notices of the Royal Astronomical Society*, 369(1):249–256, 2006.

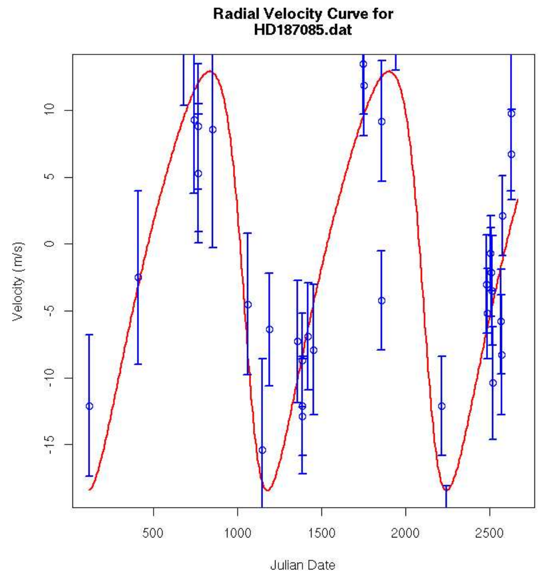


Figure 7: Keplerian orbit fitted onto the radial velocity data of HD187085 using a single planet model

```
>
>
>
>
> codamenu()
CODA startup menu

1: Read BUGS output files
2: Use an mcmc object
3: Quit

Selection: 2

Enter name of saved object (or type "exit" to quit)
1:mc_oth
Checking effective sample size ...OK
CODA Main Menu

1: Output Analysis
2: Diagnostics
3: List/Change Options
4: Quit

Selection: █
```

Figure 8: Coda menu: Choose option 2