

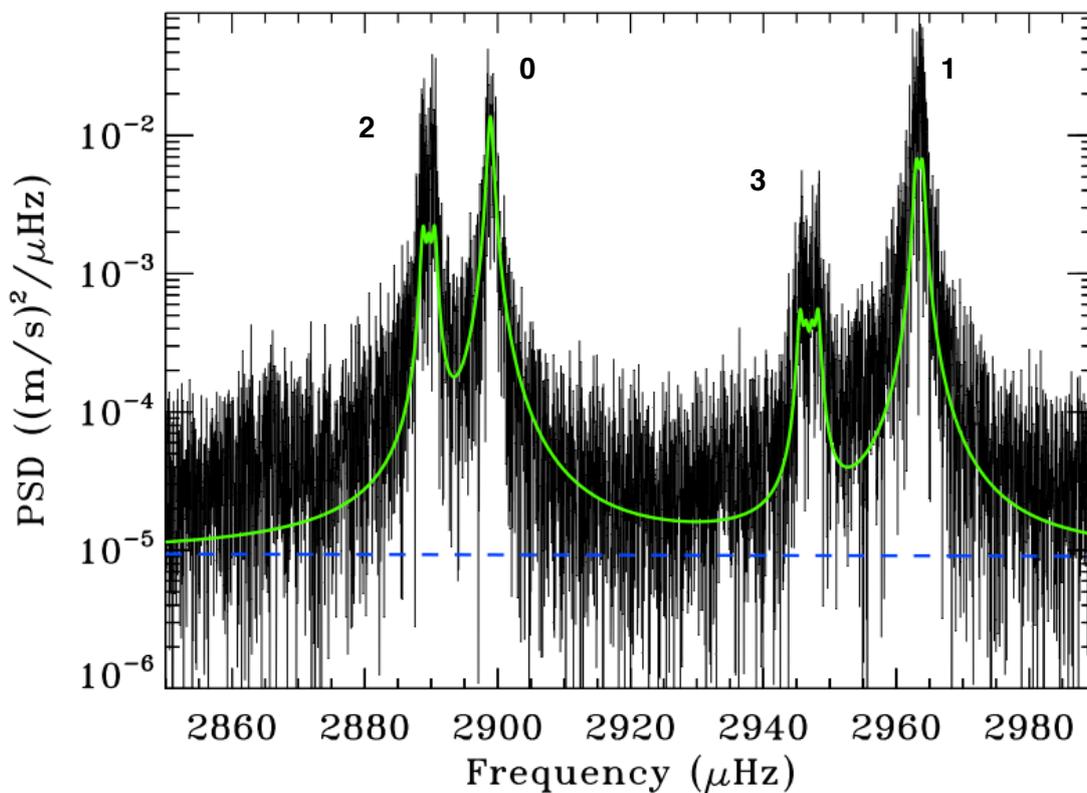


# Fitting solar $p$ modes



Enrico Corsaro  
[enrico.corsaro@cea.fr](mailto:enrico.corsaro@cea.fr)

December 2015



In this documentation I provide a description on how to install, configure, and run the tool based on the Bayesian code DIAMONDS to fit solar  $p$  modes. The first step will consist to install the basic core DIAMONDS, then the fitting tool for the Sun can be easily added and executed.

## 1. Downloading and installing DIAMONDS

Before starting, I would suggest you to take a look at the original paper work of the code, which can be found at <http://adsabs.harvard.edu/abs/2014A%26A...571A..71C> and to download the user guide manual available at [https://fys.kuleuven.be/ster/Software/Diamonds/diamonds\\_userguide/at\\_download/file](https://fys.kuleuven.be/ster/Software/Diamonds/diamonds_userguide/at_download/file). These documents will be essential help to assist you in understanding the different code parts, its working principle, troubleshooting, and for properly using it for future applications.

DIAMONDS is the basic core of the fitting tool presented in this documentation and it is necessary to have it installed in your machine before continuing. To do so please carefully follow the steps explained below. You find also a detailed description in the official website of the code, under the section Installation Guide.

- 1) Go to the official website of the code <https://fys.kuleuven.be/ster/Software/Diamonds/>
- 2) Go to the Download section, fill in the form with your affiliation details and download the latest package available (v. 1.1).
- 3) After having downloaded the **diamonds1.1.tar.gz** file, un-tar and unzip it in a local folder. A folder named **Diamonds** will then appear. For the moment, ignore its content.
- 4) Before installing Diamonds, you have to make sure that you have installed the CMake compiler, which is a free software capable of compiling C, C++ source codes by recognizing the most suited compiler installed in your machine, depending on the platform you have got (normally clang for OS X, and gcc for Linux systems). I recommend you install the version 2.8.12.2 (not older), whose dmg is available for Mac OS X 10.6 or later from the link <http://www.cmake.org/files/v2.8/cmake-2.8.12.2-Darwin64-universal.dmg>. You can however find the other versions at <http://www.cmake.org/download/> in case you are using another platform.  
**Importantly: make sure you install CMake command line tool as well, since you need that to compile Diamonds via terminal. To do so, just open CMake app, then go to Tools/Install for Command Line Use.**
- 5) Once you have installed CMake, we can proceed by compiling Diamonds. To do so, go to your Diamonds directory, then open the ASCII file CMakeLists.txt. Inside you have the first lines commented with what you have to do. Simply follow this guideline:  
from your Diamonds directory, via terminal type:

```
$ mkdir build
$ cd build
$ cmake ..
$ make -j 4
```

The 4 option takes into account the number of CPU cores you have available (e.g. 4 for a quad-core CPU). Change it according to this number in your machine.

Then the compilation and linking is executed and after a while you should display an ending message as

```
Linking CXX shared library libdiamonds.dylib
[100%] Built target diamonds
```

If so, you will have successfully installed Diamonds as a library in your computer.

## 2. Downloading and installing DIAMONDS extension for solar $p$ modes fitting

The second part concerns the fitting of the oscillation modes in the solar power spectrum, using both GOLF and VIRGO data. This requires that you install the extension of DIAMONDS provided under the Spaceln data & software deliverables, <https://www.spaceinn.eu>, which is that for the fitting of solar  $p$  modes.

By unpacking the archive file as you did for DIAMONDS, you will find a folder labeled **SunPeakBagging** and with a structure similar to that of the folder Diamonds. This extension you need to compile separately from Diamonds, but only after you have compiled Diamonds itself (see Section 1). Diamonds is used as a library for this extension.

The compilation commands are the same as those used for Diamonds. Before proceeding with the compilation make sure you put the SunPeakBagging folder at the same path level of that of Diamonds (**BE AWARE: not in the Diamonds folder itself, but in the parent directory where you have Diamonds, since the SunPeakBagging extension has to be in a different path**). Then, using any text editor, open the files

**build/multirun\_golf** (SCRIPT FILE to execute the code for GOLF data)  
**build/multirun\_virgo** (SCRIPT FILE to execute the code for VIRGO data)  
**source/VIRGO/SunVIRGOPeakBagging.cpp** (Main file for peak bagging of VIRGO data)  
**source/GOLF/SunGOLFPeakBagging.cpp** (Main file for peak bagging of GOLF data)

and edit the local paths specified between **\*\*\*YOUR LOCAL PATH HERE\*\*\*** to match those of your own machine.

To install the version to fit GOLF data carefully follow the steps below:

- 1) Move the file **SunGOLFPeakBagging.cpp** from **source/GOLF/** to **source/**
- 2) Go to the build directory **SunPeakBagging/build**, and type via terminal:  
**\$ cmake ..**  
**\$ make -j 4**
- 3) Rename the executable created inside the **SunPeakBagging/build** folder from **sunpeakbagging** to **sunpeakbagging\_golf**

To install the version to fit VIRGO data carefully follow the steps below:

- 1) If you have already installed the version to fit GOLF data, then move the file **SunGOLFPeakBagging.cpp** from **source/** back to **source/GOLF/**
- 2) Move the file **SunGOLFPeakBagging.pro** from **source/GOLF/** to **source/**
- 3) Go to the build directory **SunPeakBagging/build**, and type via terminal:  
**\$ cmake ..**  
**\$ make -j 4**
- 4) Rename the executable created inside **SunPeakBagging/build** folder from **sunpeakbagging** to **sunpeakbagging\_virgo**

## 3. What is provided in the package

In this extension an example of configuring files for the Sun fitting of both GOLF (radial velocity measurements, Gabriel et al. 1995) and VIRGO (photometric variations, Froehlich et al. 1995) data is given. These fittings should work out of the box.

First of all, in the **data** folder you find two subfolders, GOLF and VIRGO, each one containing the 1-year length PSD from 1996 (Year1996.txt) in an ASCII format. The first column is frequency (in  $\mu\text{Hz}$ ), second column is the power spectral density (PSD), expressed in  $(\text{m/s})^2 / \mu\text{Hz}$  for GOLF and in  $\text{ppm}^2 / \mu\text{Hz}$  for VIRGO.

In the **results** folder you find two subfolders, **GOLF/Year1996/ThreeHarvey** and **VIRGO/Year1996/ThreeHarvey**. Inside these folders the output results from the fitting will be stored. In

each of these folders you can also find the configuring files for the computation and the solution of the background fitting for the corresponding year of observation. The files are explained below:

1. [Xmeans\\_configuringParameters.txt](#) and [NSMC\\_configuringParameters.txt](#), are files containing the configuring parameters of the code. These files should be always present in the folder that we use to store our results. The content of these files should be left unchanged.
2. [backgroundParameters.txt](#) contains the parameters for the background in the PSD of the year 1996 dataset, both for GOLF in the folder **GOLF/Year1996/ThreeHarvey**, and for VIRGO in the folder **VIRGO/Year1996/ThreeHarvey**. These parameters are given as an input reference for the fitting and are kept as constants. They refer to the white noise and to the three Harvey-like components, according to:

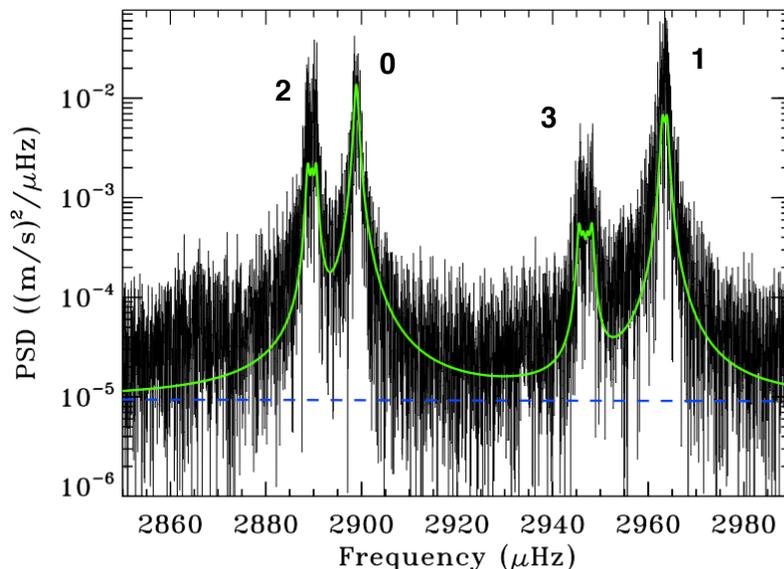
$$B(\nu) = W + \sum_{i=1}^3 \frac{a_i^2/b_i}{(1 + \nu/b_i)^{c_i}}$$

where  $\mathbf{a}_i$  are the amplitudes (not normalized) and  $\mathbf{b}_i$  the characteristic frequencies of each Harvey-like component.  $\mathbf{c}_i$  are the related exponents, which are also provided in the list. A total of 10 parameters is therefore needed to compute the background level in the PSD.

3. [modesWithRotation\\_hyperParameters.txt](#) contains the prior values for each free parameter of each oscillation mode to be fitted, assuming that we also want to include the rotational effect on it. The model parameters are given in the equation below for a mode of angular degree  $\ell$

$$L(\nu) = \sum_{m=-\ell}^{m=+\ell} \frac{A^2}{\pi\Gamma} \frac{\mathcal{E}_{\ell,m}(i)}{1 + 4 \left( \frac{\nu - \nu_0 - m\delta\nu_{\text{rot}}}{\Gamma} \right)^2}$$

which means that we fit, in order, frequency  $\nu_0$  (in  $\mu\text{Hz}$ ), amplitude  $A$  (in  $\text{m/s}$  for GOLF and  $\text{ppm}$  for VIRGO), linewidth  $\Gamma$  (in  $\mu\text{Hz}$ ), and rotational splitting  $\delta\nu_{\text{rot}}$  (in  $\mu\text{Hz}$ ). The inclination angle of the rotation axis is in this case fixed to  $i = 90^\circ$ , being it known for the Sun. These four free parameters are repeated in blocks, each block corresponding to a single oscillation mode to be fitted. Since uniform priors are used, the left column in the file defines the lower bounds, while the right column the upper bounds, for each free parameter to be fit. In the example provided, we list three different modes in the order  $\ell=2,3,1$ , corresponding to the figure below for GOLF, where the final fit is indicated in green and the background level with a dashed blue line.



4. [modesWithRotation\\_angularDegrees.txt](#) contains the 1 column list of the angular degrees for all the modes for which we want to fit the rotational effect. Typically for this case, the modes are  $l=1,2,3$ .  $l=4$  modes could also be included if desired. The order of the values in the column has to match the same order in which the modes are listed in the prior parameter file. In the example given, this order is  $l=2,3,1$ .
5. [modesWithoutRotation\\_hyperParameters.txt](#) contains the prior values for each free parameter of each oscillation mode to be fitted, assuming that we do not include the rotational effect on it. The model parameters are given in the equation below for a mode of any angular degree

$$L(\nu) = \frac{A^2}{\pi\Gamma} \frac{1}{1 + 4 \left( \frac{\nu - \nu_0}{\Gamma} \right)^2}$$

so this time we only fit, in order,  $\nu_0$  (in  $\mu\text{Hz}$ ), amplitude  $A$  (in  $\text{m/s}$  for GOLF and  $\text{ppm}$  for VIRGO), linewidth  $\Gamma$  (in  $\mu\text{Hz}$ ). In the example, we have only one radial mode,  $l=0$ , that is always fitted without the rotational effect.

In this way everything is set up, and the code does not need to be recompiled every time you need to change input data and the oscillation peaks you want to fit. All you have to do is to edit the ASCII files for the prior parameters

#### 4. Run the fitting code

The example provided allows you to run the code just out of the box, both for GOLF and VIRGO datasets. To execute the computation follow the steps below:

1. Go to the directory `SunPeakBagging/build/`
2. If you want to execute the code for GOLF data, then use the script `multirun_golf`, previously edited with the correct local path in your machine.
3. Execute the script via terminal by typing:

```
./multirun_golf 1996 0 1
```

In this example you are computing the fit for the GOLF dataset year 1996, and you give a label to your test called "0". The last number indicates how many runs (tests) should be executed at the same time. For your application always use 1.

You can run again the script by changing the label "0" of the run. Values from 0 to 99 are allowed.

4. After you execute the script, you will see that an output file is generated in the same folder, named `output_golf_year1996.txt`. This file contains the output messages printed during the computation. An explanation of the different messages and strings displayed is provided in the user guide manual of the DIAMONDS code. After the computation is completed you will read the last string:

```
Process #00 has been completed.
```

5. You can now check the results. These are stored in the corresponding folder of the year dataset that you used. In this example you will have your results in the folder: `SunPeakBagging/results/GOLF/Year1996/ThreeHarvey/`. Inside this folder you will find a subfolder labeled with the label of your run. For this test the label of the folder will be `00`. In this subfolder you will find many ASCII files, which are all explained in the user guide manual of DIAMONDS. In particular, your final results are stored in the file `peakbagging_parameterSummary.txt`. There you will find the estimates and error bars, for each free parameter that you fit, in the order given by the input priors (first for the modes with rotation to be fitted, and then for the modes without rotational effect). In general the mean estimates, as well as the medians, for each parameter should be considered as the final estimate from the computation.

6. A perfectly analogous procedure can be done for the case of VIRGO by using the script `multirun_virgo`. In this case you will find the results in the folder `SunPeakBagging/results/VIRGO/Year1996/ThreeHarvey/`.