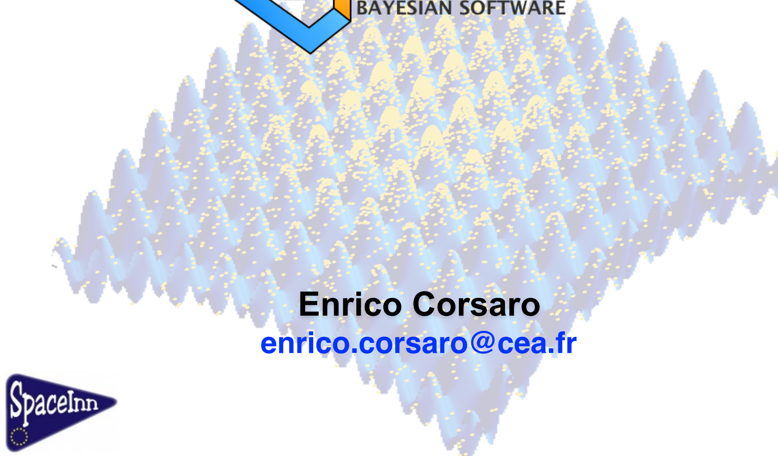
Asteroseismology and Exoplanets:
Listening to the Stars and Searching for New Worlds
IVth Azores International Advanced School in Space Sciences

DE LA RECHERCHE À L'INDUSTRIE

cea

Asteroseismic Data Analysis with DIAMONDS™

BAYESIAN SOFTWARE



Enrico Corsaro
enrico.corsaro@cea.fr

Institute of Research into the Fundamental Laws of the Universe
Astrophysics Division

www.cea.fr

LABORATOIRE AIM PARIS-SACLAY, CEA/DRF

19 JULY 2016

CURRICULUM VITAE

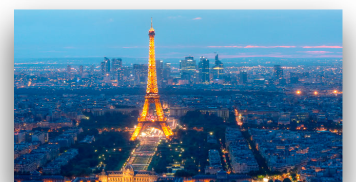
1. PhD 2009 - 2012 in Asteroseismology
   Catania (**Italy**)
   Sydney (**Asutralia**)

   Supervisors:
   Dr. Alfio Bonanno
   Prof. Dennis Stello

2. Postdoc 2012 - 2014
   Leuven (**Belgium**)

Collaborator: **Dr. Joris De Ridder**

3. Postdoc 2014 - 2016
   Saclay Paris (**France**)



Collaborator: **Dr. Rafael A. Garcia**

READING

Bayes in the sky:
Bayesian inference and model selection in cosmology

Roberto Trotta*

Oxford University, Astrophysics Department
Denys Wilkinson Building, Keble Rd, Oxford, OX1 3RH, UK
(*March 28, 2008*)

DIAMONDS: A new Bayesian nested sampling tool*

Application to peak bagging of solar-like oscillations

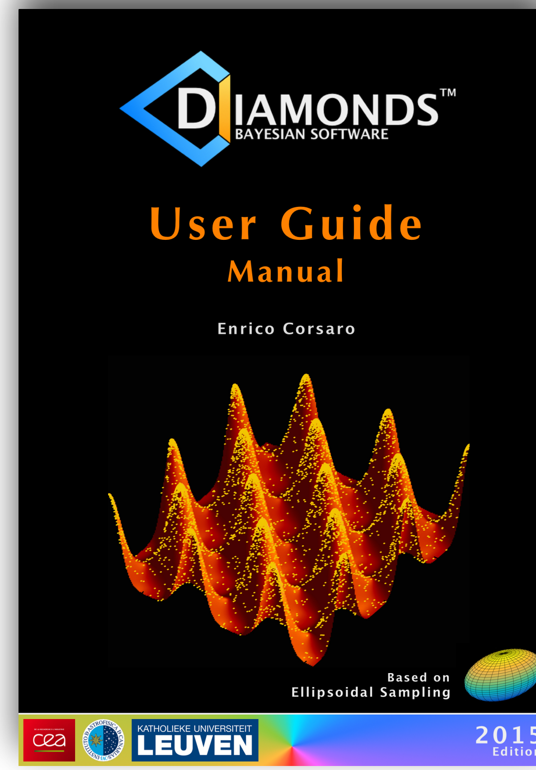
E. Corsaro and J. De Ridder

Instituut voor Sterrenkunde, KU Leuven, Celestijnenlaan 200D, 3001 Leuven, Belgium
e-mail: emncorsaro@gmail.com, joris.deridder@ster.kuleuven.be

Bayesian peak bagging analysis of 19 low-mass low-luminosity red giants observed with *Kepler*

E. Corsaro[1,2,3,4], J. De Ridder[1], and R. A. García[2]

[1] Instituut voor Sterrenkunde, KU Leuven, Celestijnenlaan 200D, 3001 Leuven, Belgium
[2] Laboratoire AIM, CEA/DSM – CNRS – Univ. Paris Diderot – IRFU/SAp, Centre de Saclay, 91191 Gif-sur-Yvette Cedex, France
e-mail: enrico.corsaro@cea.fr
[3] Instituto de Astrofísica de Canarias, 38205 La Laguna, Tenerife, Spain
[4] Universidad de La Laguna, Departamento de Astrofísica, 38206 La Laguna, Tenerife, Spain



SYMBOLS

OUTLINE

Lecture

1. **Introduction to Bayesian statistics** (comparison with frequentist approach)
2. **Numerical techniques for Bayesian Inference** (nested sampling MC)
3. **The DIAMONDS code** (overview, working principle, features, efficiency)

Hands-on tutorial

4. **The fitting of the background signal in a red giant star** (e.g. granulation)
5. **The fitting of the oscillation modes in a red giant star** (p and mixed modes)
6. **The Bayesian peak significance test** (using Bayesian model comparison)

LECTURE

1. - BAYESIAN STATISTICS

2. - NESTED SAMPLING MONTE CARLO

3. - THE DIAMONDS CODE

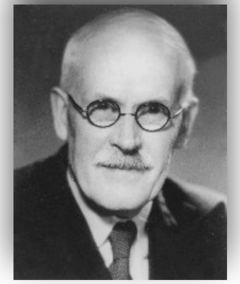
4. - FITTING THE BACKGROUND SIGNAL

5. - FITTING THE OSCILLATION MODES

6. - PEAK SIGNIFICANCE TEST

Bayesian Probability

- First introduced by Thomas Bayes 1763
- Rediscovered by Harold Jeffreys 1939
- More recently expounded by Edwin T. Jaynes (1983)



Frequentist Probability

- "*The probability of an event is defined as the number of times the event occurs over the total number of trials, in the limit of an infinite series of equiprobable repetitions.*"

Jerzy Neyman, Egon Pearson, Pierre-Simon de Laplace, Adrien-Marie Legendre, Ronald Fisher, Harold Jeffreys, Edwin T. Jaynes, Larry Brethorst

Bayesian Probability

- Opposed to the "frequentist" mindset, probability is **conditional** to our knowledge
- ***"Probability is a measure of the degree of belief about a proposition"***
- It recovers the frequentist results and is applicable to any proposition in general

A good review of Bayesian statistics can be found in

1) Roberto Trotta, Bayes in the sky: Bayesian inference and model selection in cosmology, Contemp.Phys.49: 71-104, 2008
2) Corsaro, E. PhDT, 2013

Frequentist Probability: problems

- Definition is circular
- Requires that the event can be repeated
- The exact definition holds only for an infinite number of repetitions

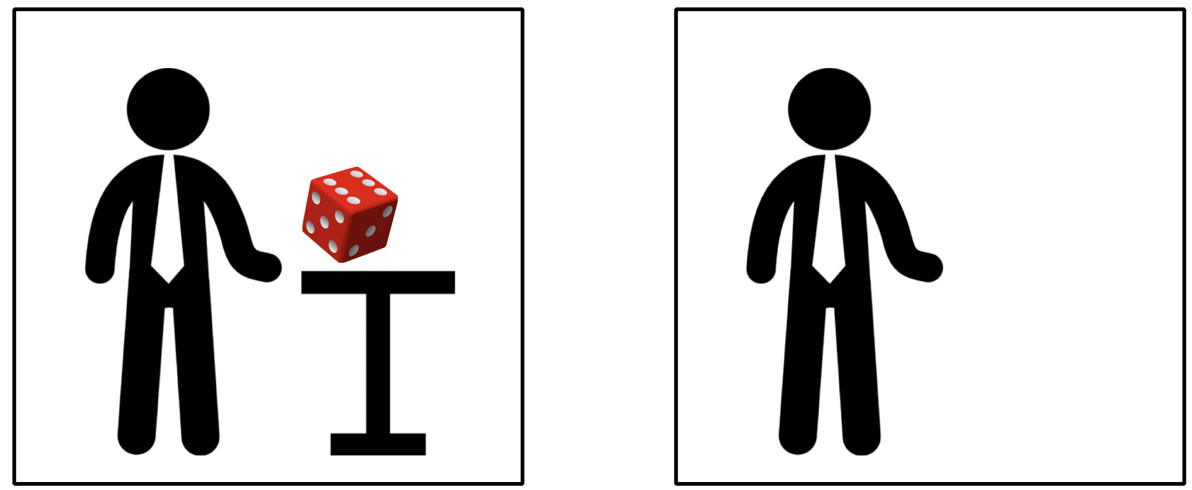
"The probability of an event is defined as the number of times the event occurs over the total number of trials, in the limit of an infinite series of equiprobable repetitions."

The definition of frequentist probability is not adequate.

1) Circular definition
2) Requires that the event is repeatable: many times this is not the case: e.g. an historical event that happens like the beginning of a war
3) The definition implies an infinite number of repetitions, which never happens in reality: asymptotic extrapolations

Example of the coin toss. Suppose we want to see if the coin is properly balanced (e.g. tensor of inertia symmetric about the plane of the coin). Fairness of coin toss
p_Heads = p_Tails = 0.5
but this has nothing to do with the proper balance of the coin. A skilled coin-tosser can change the output of the toss by imposing the required velocity to flip the coin differently. We therefore have a state of knowledge before the toss (e.g. angular momentum, velocity imposed by the coin-tosser), that prevents us from defining a fair toss as that for which p_Heads = 0.5.

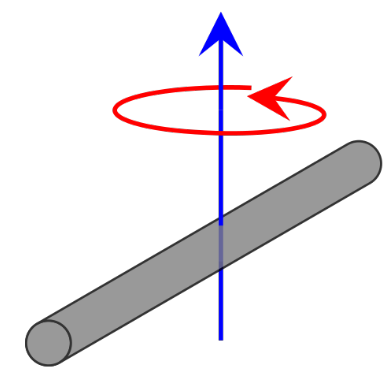
Bayesian mindset: the dice example



Taking into account prior information is essential. Suppose that someone is throwing a dice in a room different than the one you are in. Suppose you are then asked the odds that player rolled a 5. Your answer will be 1/6, given that there are 6 possible outcomes.

But now suppose that after the player has rolled the dice, someone comes out of his room and says that the outcome is an odd number, but he doesn’t known which one. There are only 3 odd numbers in a dice (1,3,5), therefore when you are asked the odds that the player rolled a 5 your answer will be 1/3.

Bayesian mindset: the coin toss example



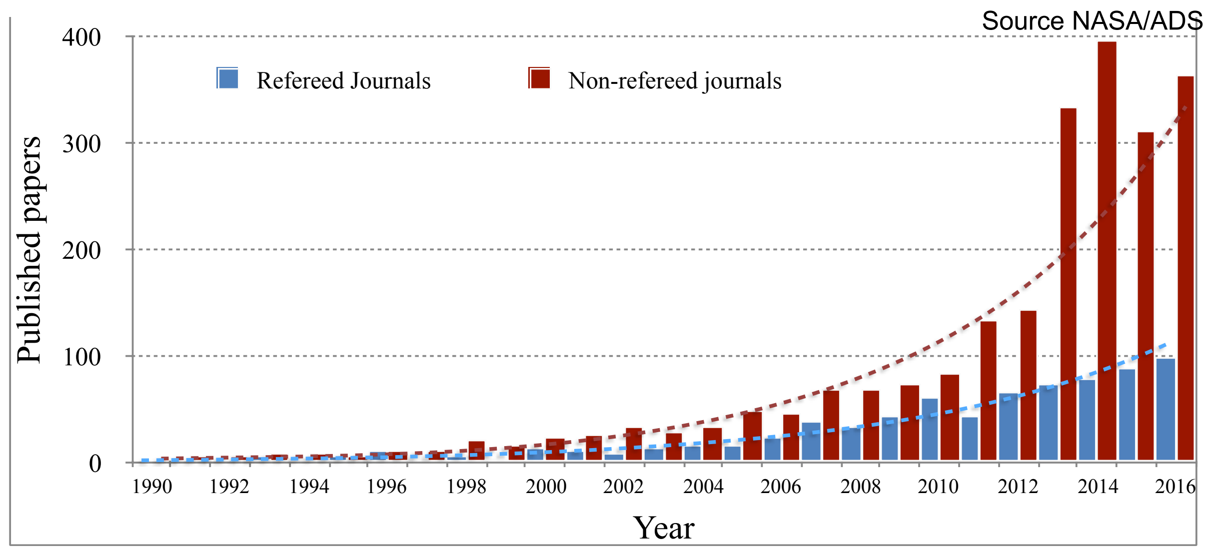
Example of the coin toss. Suppose we want to see if the coin is properly balanced (e.g. tensor of inertia symmetric about the plane of the coin). Fairness of coin toss

p_Heads = p_Tails = 0.5

but this has nothing to do with the proper balance of the coin. A skilled coin-tosser can change the output of the toss by imposing the required velocity to flip the coin differently. We therefore have a state of knowledge before the toss (e.g. angular momentum, velocity imposed by the coin-tosser), that prevents us from defining a fair toss as that for which p_Heads = 0.5.

Lawrence Stone, Colleen Keller et al. applied Bayesian statistics for the search of the Air France Flight 447, disappeared in the Atlantic Ocean in 2009. She combined prior information from last point of contact, speed of currents, weather conditions, and other statistics from previous cases of crashes. The posterior probability map obtained by combining different scenarios together allowed to find the airplane, after 2 years of search. The searching area was increasingly reduced by the unsuccessful results and the continuous updates on the currents.

Bayes in Astrophysics



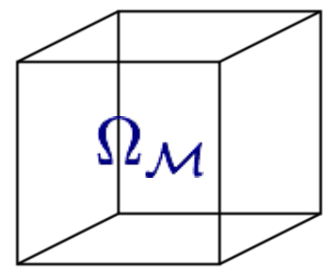
The number of papers with the word “Bayesian” in the title from 1990 until 2016.

Bayes' Theorem

$$\mathbf{D} = \{d_1, d_2, \ldots, d_m\}$$ Dataset (observations)

$$\mathcal{M} = \mathcal{M}(\boldsymbol{\theta})$$ Model to be tested

$$\boldsymbol{\theta} = \{\theta_1, \theta_2, \ldots, \theta_k\}$$ k free parameters (parameter vector)



k-dimensional **parameter space** defined by the free parameters

We can visualize the parameter space containing the solution of a given inference process as a k-dimensional box.

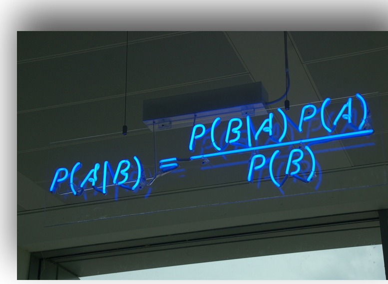
Bayes' Theorem

$$\mathcal{L}(\boldsymbol{\theta}) = \mathcal{L}(\boldsymbol{\theta} \mid \mathbf{D}, \mathcal{M})$$ Likelihood function

$$\pi(\boldsymbol{\theta}) = \pi(\boldsymbol{\theta} \mid \mathcal{M}) \qquad \int \pi(\boldsymbol{\theta}) d\boldsymbol{\theta} = 1$$ Prior PDF

$$p(\boldsymbol{\theta}) = p(\boldsymbol{\theta} \mid \mathbf{D}, \mathcal{M}) \qquad \int p(\boldsymbol{\theta}) d\boldsymbol{\theta} = 1$$ Posterior PDF

$$p(\boldsymbol{\theta}) = \frac{\mathcal{L}(\boldsymbol{\theta})\, \pi(\boldsymbol{\theta})}{\mathcal{E}}$$

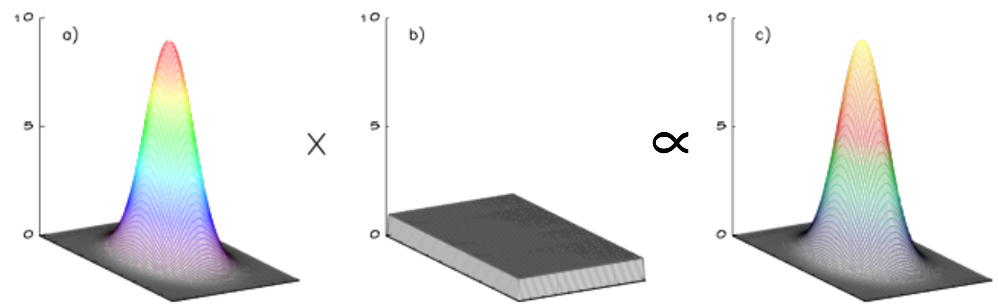


Cox Axioms
Sum rule & Product rule

The Bayes theorem arises from a simple combination of logic reasoning axioms, known as Cox axioms. These axioms are the sum rule and the product rule.

Bayes' Theorem

Likelihood **Prior** **Posterior**

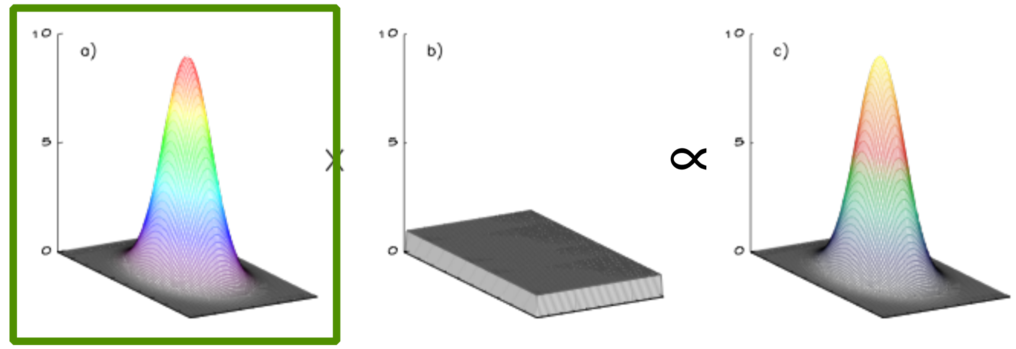


$$p\left(\boldsymbol{\theta}\right) = \frac{\mathcal{L}\left(\boldsymbol{\theta}\right)\pi\left(\boldsymbol{\theta}\right)}{\mathcal{E}}$$

A simple way to visualize the Bayes theorem is to consider the multiplication of a Gaussian likelihood by a flat (uniform) distribution. This will lead to a Gaussian posterior distribution, whose normalization volume factor is the Bayesian evidence.

Bayes' Theorem

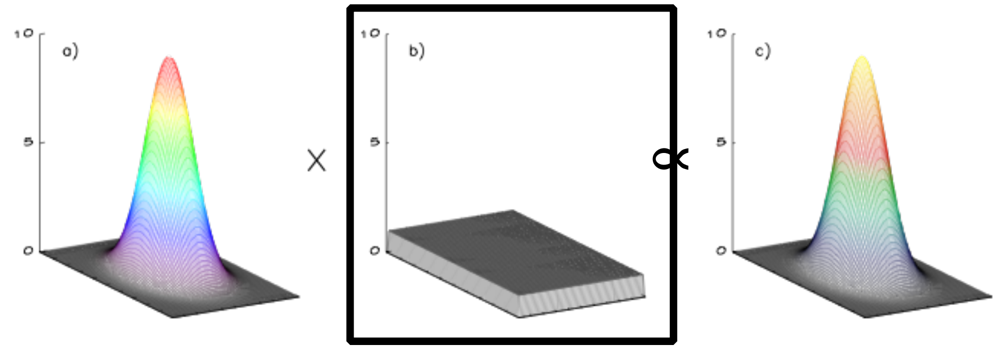
Likelihood **Prior** **Posterior**



$$p(\boldsymbol{\theta}) = \frac{\boxed{\mathcal{L}(\boldsymbol{\theta})}\,\pi(\boldsymbol{\theta})}{\mathcal{E}}$$

Bayes' Theorem

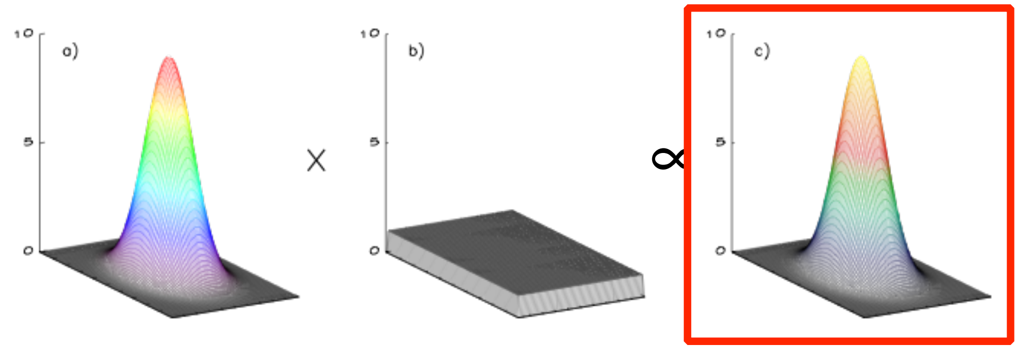
Likelihood **Prior** **Posterior**



$$p\left(\boldsymbol{\theta}\right) = \frac{\mathcal{L}\left(\boldsymbol{\theta}\right)\boxed{\pi\left(\boldsymbol{\theta}\right)}}{\mathcal{E}}$$

Bayes' Theorem

Likelihood **Prior** **Posterior**



$$p(\boldsymbol{\theta}) = \boxed{\frac{\mathcal{L}(\boldsymbol{\theta})\,\pi(\boldsymbol{\theta})}{\mathcal{E}}}$$

Likelihood functions

$$\mathcal{L}(\boldsymbol{\theta}) \propto \prod_{i}^{N} \exp\left[-\left(\frac{O_i - E_i(\boldsymbol{\theta})}{\sigma_{O_i}}\right)^2\right]$$

- For a set of observations distributed as a chi-square with two d.o.f. we use the **Exponential Likelihood** (e.g Fourier spectra, Duvall & Harvey 1986)

$$\ln \mathcal{L}(\boldsymbol{\theta}) \propto \sum_{i}^{N} \left[\ln E_i(\boldsymbol{\theta}) + \frac{O_i}{E_i(\boldsymbol{\theta})}\right]$$

The Gaussian likelihood is the most commonly adopted. It corresponds to a chi-square variable, which is widely adopted in minimization-fitting approaches. The exponential likelihood, is instead used for the fitting of power spectra. For computational reasons, likelihood are often used in logarithmic form. This is because they often lead to very large (or very small) numbers, since they are defined as a productoria. Switching to the logarithmic scales avoids computational overflow or underflow errors.

Priors PDFs

- **Uninformative** priors
  - Ignorance priors
  - Reference priors
  - Jeffreys' priors $\qquad \pi(\theta) \propto \theta^{-1}$
- **Informative** priors
  - Uniform priors $\qquad \pi(\theta) \propto \mathrm{const}$
  - Gaussian priors $\qquad \pi(\theta) \propto \mathcal{N}(\mu, \sigma)$
  - Super-Gaussian priors $\qquad \pi(\theta) \propto \mathcal{S}(\mu, \sigma, w)$

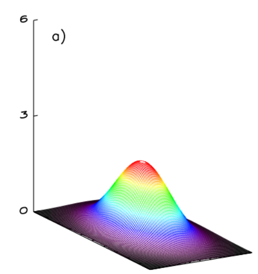
Reference priors maximize divergence from posterior to prior to produce least informative priors.

Eg. Temperature at noon tomorrow.
Gaussian prior using mean value as the temperature of today at noon and as variance the day-to-day variation measured.

Jeffreys' priors are also very useful. They reflect a state of ignorance about the order of magnitude of a given parameter, which often can be the case when performing inferences where no clear a priori information is available. The strength of Jeffreys' priors is that one can easily convert a Jeffreys' prior for a parameter into an uniform prior for the logarithm of the parameter (variable change). This allows for fast and handy inclusions of priors in the statistical inference.

The role of priors

- **Weak** data information and informative priors lead to **prior-dominated posterior**

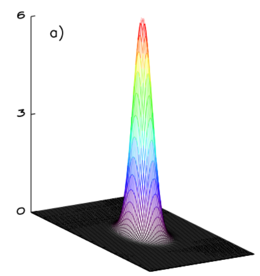


X =

A prior-dominated posterior is essentially depending upon our initial choice on the parameters of the model. This means that the data available are not sufficiently stringent to allow for constraining the free parameters of a model. As a result, by changing priors, and models too, the result will change thus leading to an unreliable condition to draw conclusion.

The role of priors

- **Strong** data information and informative priors lead to **data-dominated posterior**



X

=

When the data available are very informative, as it is often the case with space-based photometric observations (e.g. NASA Kepler), then even a different choice of priors can still lead to the same result. This condition is ideal to fully exploit the potential of the dataset and to allow comparing and testing different proposed models.

Statistical Inference

- The statistical inference of a dataset is divided in two problems:
  - **Parameter Estimation**
    Allows to obtain the estimates of all the free parameters and the corresponding error bars
  - **Model comparison**
    Provides a way to select the best model to represent the observations among different possible ones

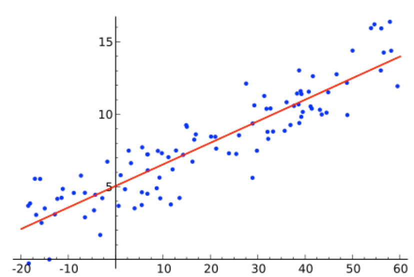
Frequentist Parameter Estimation

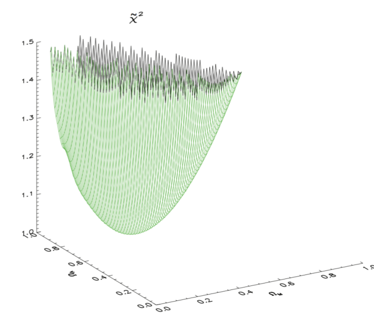
- k-dimensional parameter space

$$\boldsymbol{\theta} = \{\theta_1, \theta_2, \ldots, \theta_k\}$$

k free parameters (parameter vector)

- The likelihood as function of k parameters
- Estimators are obtained from the likelihood only. No prior information is taken into account.





$$y(x; a, b) = mx + b$$

$$\chi^2_{\min}(a, b)$$

In the frequentist approach (e.g. chi-square minimization, or MLE) there is no marginalization. We are interested in the global minimum (if chi-square) or maximum (if MLE) of the distribution. This corresponds to the modal value of the parameter that is estimated. In this way we are not taking into account the global behavior of the distribution, and any degeneracies, asymmetries, or outliers, are not playing any role.

A typical example of minimization problem is that of the linear regression (or linear fitting).

Bayesian Parameter Estimation

- k-dimensional parameter space

$$\boldsymbol{\theta} = \{\theta_1, \theta_2, \ldots, \theta_k\}$$

k free parameters (parameter vector)

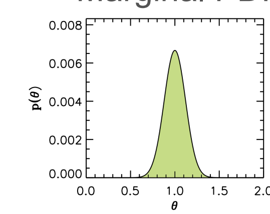
- The posterior PDF is a function of k parameters
- To obtain the PDF of a single parameter we can marginalize the posterior PDF

$$p(\theta_1) = \int p(\boldsymbol{\theta})\, d\theta_2 \ldots d\theta_k$$

θ_1

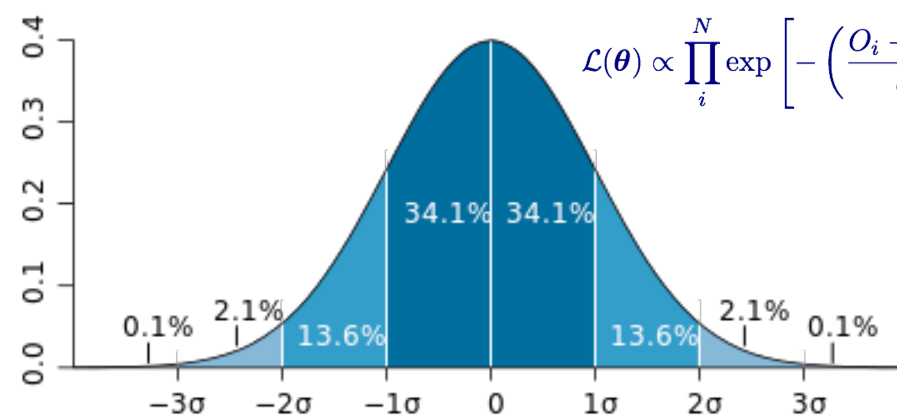
Mean
Mode
Median
Variance
Credible Intervals

Marginal PDF



Marginalization is the key of the Bayesian parameter estimation. Marginalizing means integrating all non-interesting information over, so leaving out only the hypothesis we are interested to test (i.e. the parameter we want to estimate).
Marginalizing can be thought equivalently as averaging all the available information to incorporate it into the final estimate. This allows taking into account the general properties of the distribution (e.g. asymmetry, degeneracies, outliers).

Frequentist Parameter Estimation: Confidence Intervals



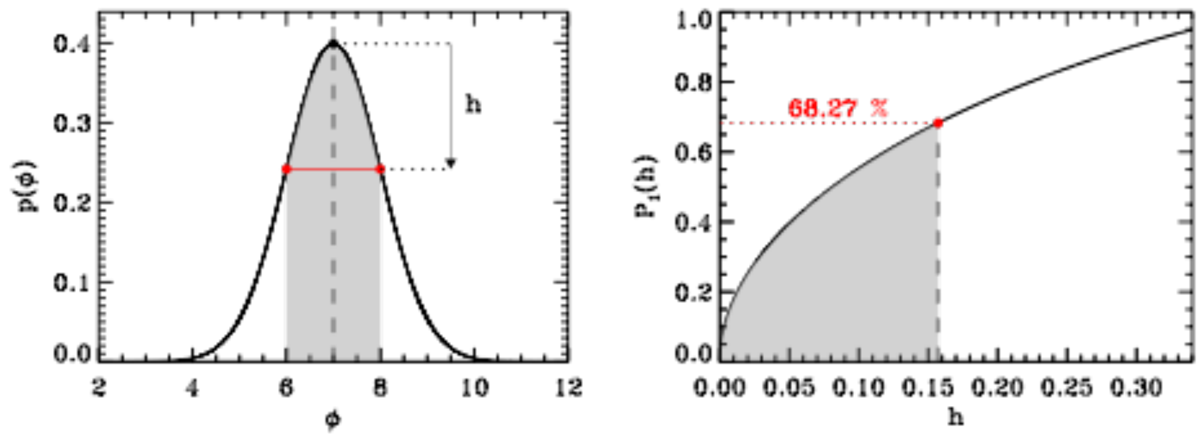
$$\mathcal{L}(\boldsymbol{\theta}) \propto \prod_{i}^{N} \exp\left[-\left(\frac{O_i - E_i(\boldsymbol{\theta})}{\sigma_{O_i}}\right)^2\right]$$

Hessian matrix in general if no marginalization (k-dimensional distribution): lower limits only!

Confidence Intervals (standard deviation in 1D), give the number of times that the free parameter will fall within the range (mode +/- sigma) in an infinite number of repetitions of the same event.

“You have a XXX probability that by repeating the experiment infinite times, the outcomes will fall in the given interval”

Bayesian Parameter Estimation: Credible Intervals

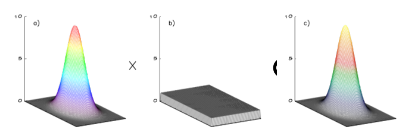


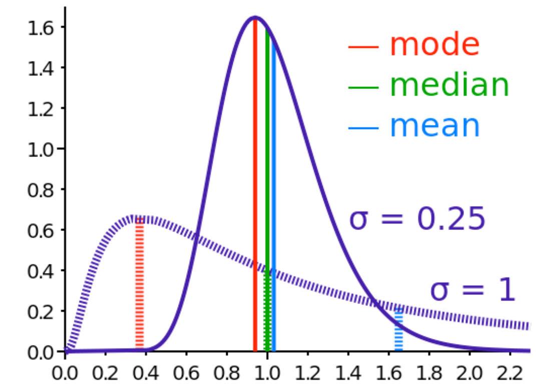
Credible intervals exist only in reference to a marginalization of the k-dimensional distribution. They always refer to a one-dimensional MPD

Credible Intervals give the probability that the free parameter's estimator lies within the given interval, just for that particular event and choice of priors (realization).

Estimators (Dispersion indices)

- **Mode** (or modal value, the most probable: frequentist)
- **Median** (divides the probability distribution in two equal parts)
- **Mean** (the average value of the free parameter from the distribution)
- If distribution is symmetric mode = median = mean
  - If prior PDF is not changing likelihood, then Bayesian and frequentist approach are formally identical





The mode is a frequentist parameter because it represents the most likely value attained, while median and mean are Bayesian because they take into account the global shape of the distribution (asymmetry, degeneracy, outliers, etc.). The median parameter is usually preferred over the mean as it is the most resistant estimator in statistics, meaning that it is the least sensitive to strong outliers that may hamper the result.

Many statisticians do however prefer the mean as it is considered a more realistic estimator (it can be influenced by outliers).

Frequentist Model Comparison

- The best model is the one that gives the best match with the observations. **Only fit quality is taken into account**
- **More complex models favored**! Increasing the number of free parameters always (or almost) improves the fit
- How to decide where to stop in increasing complexity? Usually when no significant improvement in the maximum likelihood is observed, the model is considered too complex

$$\mathcal{L}_{\mathrm{max},1} > \mathcal{L}_{\mathrm{max},2} \qquad \chi^2_{\mathrm{min},1} < \chi^2_{\mathrm{min},2}$$

Increasing the number of parameters without a rigorous statistical control can be dangerous. More complex models tend to be favored, meaning that we are biased against simplistic explanations. A careful assessing of the fitting quality can however still provide a reasonable outcome, but it is suited only for the experienced user.

Comparing two maximum likelihood values, or two minimum chi-squares, does not mean performing a model comparison! In this way you are not testing any hypothesis of the model, and you are not taking into account its complexity. It is only the fit quality that is compared through this process, which is only part of the total information that must be considered.

Frequentist ~~Model Comparison~~ Fit quality comparison

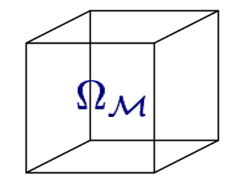
- The best model is the one that gives the best match with the observations. **Only fit quality is taken into account**
- **More complex models favored**! Increasing the number of free parameters always (or almost) improves the fit
- How to decide where to stop in increasing complexity? Usually when no significant improvement in the maximum likelihood is observed, the model is considered too complex

$$\mathcal{L}_{\max,1} > \mathcal{L}_{\max,2} \qquad \chi^2_{\min,1} < \chi^2_{\min,2}$$

Bayesian Model Comparison

- **Bayesian Evidence** is an dimensionless quantity given as a k-dimensional integral over the entire parameter space (does not exist in frequentist approach!)

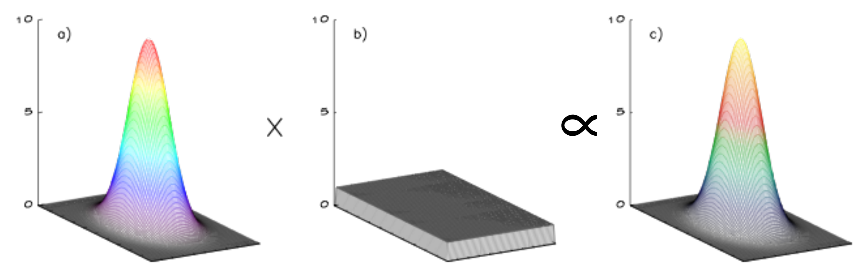
$$\mathcal{E} = \int \mathcal{L}\left(\boldsymbol{\theta}\right) \pi\left(\boldsymbol{\theta}\right) d\boldsymbol{\theta}$$



$$p\left(\boldsymbol{\theta}\right) = \frac{\mathcal{L}\left(\boldsymbol{\theta}\right) \pi\left(\boldsymbol{\theta}\right)}{\mathcal{E}}$$

Bayesian Model Comparison

- **Bayesian Evidence** is key for model comparison problems

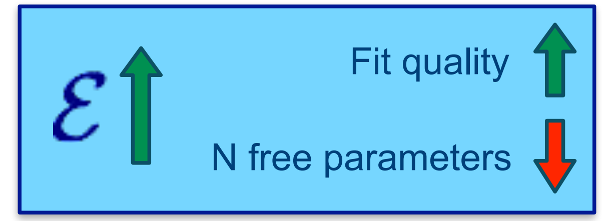


$$p\left(\boldsymbol{\theta}\right) = \frac{\mathcal{L}\left(\boldsymbol{\theta}\right)\pi\left(\boldsymbol{\theta}\right)}{\mathcal{E}}$$

The Bayesian evidence represents the normalization factor of the product of the likelihood distribution by the prior PDF. By default, the likelihood function is not a probability density function, hence to get a posterior PDF as an outcome of the Bayes' theorem we need to normalize the distribution by its total volume.

Bayesian Model Comparison

- **Bayesian Evidence** is weight: simple models are preferred (Occam's razor).
- Direct and effective solution to model comparison problems!



Bayesian model comparison is typically performed following the so-called Jeffreys' scale of strength for the evidence. However, a real Bayesian approach will never rule out a model just because its evidence is below an empirical threshold. All models should me kept during the model comparison, but assigning probabilities to each according to the weight of their Bayesian evidence in the total.

Bayesian Model Comparison

- **Bayesian Evidence** is weight: simple models are preferred (Occam's razor).
- Direct and effective solution to model comparison problems!

$\mathcal{E} \uparrow$ Fit quality $\uparrow$

N free parameters $\downarrow$

Jeffreys' scale

$\lvert \ln B_{01} \rvert$	Odds	Probability	Strength of evidence
< 1.0	$\lesssim 3:1$	< 0.750	Inconclusive
1.0	$\sim 3:1$	0.750	Weak evidence
2.5	$\sim 12:1$	0.923	Moderate evidence
5.0	$\sim 150:1$	0.993	Strong evidence

Odds ratio

$$\mathcal{O}_{i,j} = \frac{\mathcal{E}_i}{\mathcal{E}_j} \cancel{\frac{\pi(\mathcal{M}_i)}{\pi(\mathcal{M}_j)}}$$

$$\mathcal{B}_{i,j} = \mathcal{E}_i / \mathcal{E}_j \geq 150$$

$$\ln \mathcal{B}_{i,j} = \ln \mathcal{E}_i - \ln \mathcal{E}_j \geq 5$$

Bayes factor

Bayesian model comparison is typically performed following the so-called Jeffreys' scale of strength for the evidence. However, a real Bayesian approach will never rule out a model just because its evidence is below an empirical threshold. All models should me kept during the model comparison, but assigning probabilities to each according to the weight of their Bayesian evidence in the total.

1. - BAYESIAN STATISTICS

2. - NESTED SAMPLING MONTE CARLO

3. - THE DIAMONDS CODE

4. - FITTING THE BACKGROUND SIGNAL

5. - FITTING THE OSCILLATION MODES

6. - PEAK SIGNIFICANCE TEST

Bayesian Statistics: everything is perfect?

- **Problems: Type 1 - Integration**

- **Problems: Type 2 - Sampling**

Bayesian Statistics: everything is perfect?

- **Problems: Type 1 (Integration)**
  - For high dimensionality (already from **k > 3**) no more analytical solutions to the marginalization problem (hence also the computation of the Bayesian Evidence integral)

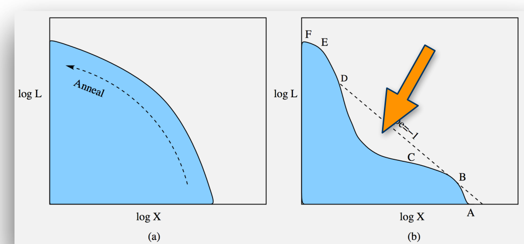
$$p(\theta_1) = \int p(\boldsymbol{\theta})\, d\theta_2 \ldots d\theta_k \qquad \mathcal{E} = \int \mathcal{L}(\boldsymbol{\theta})\, \pi(\boldsymbol{\theta})\, d\boldsymbol{\theta}$$

  - Numerical integration is needed but for higher dimensions (**k ~ 20**) even numerical integration is not enough (too approximated and computationally demanding)

Bayesian Statistics: everything is perfect?

- **Solutions: Type 1 (Integration)**
  - For relatively low k, (up to **k ~ 10**) one can split the k-dimensional integral into k one-dimensional integrals (e.g. Corsaro PhDT 2013) but method requires a grid computation of values (very computationally demanding)
  - For higher dimensions **thermodynamic integration (annealing)** Use *temperature* power of likelihood to gradually move from prior to posterior (problems with “phase changes” in likelihood distribution)

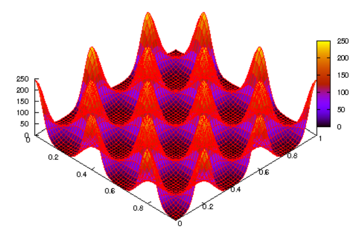
$$\mathcal{L}^{\lambda}(\boldsymbol{\theta})$$
$$0 \leq \lambda \leq 1$$

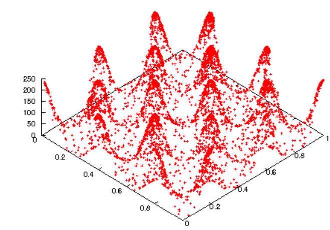


Thermodynamic integration suffers from phase changes in likelihood.

Bayesian Statistics: everything is perfect?

- **Problems: Type 2 (Sampling)**
  - Numerical sampling techniques are approximate by definition
  - Complex distributions are extremely difficult to sample
  - Sampling algorithm can get stuck into a local maximum and never be able to explore all the parameter space (e.g. Eggbox distribution)
  - Computational time and the number of samples required can become a big problem





Bayesian Statistics: everything is perfect?

- **Solutions: Type 2 (Sampling)**
  - Monte Carlo (MC) methods
    - Markov chain Monte Carlo (MCMC) (see Tiago's lecture)
    - Nested sampling Monte Carlo (NSMC)

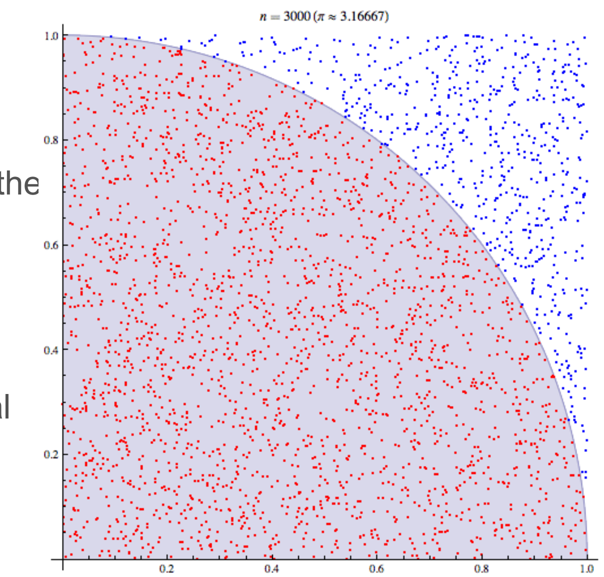
What is a Monte Carlo?

- Enrico Fermi & Stanislaw Ulam in 1940s @ LASL
- No analytical solution available to the experimental problem
- Simulate the experiment: Monte Carlo in homage of the casino
- **Characteristics**: reconstruct a deterministic problem with a probabilistic approach
- **Inverse** approach to that of standard algorithms: search for probabilistic analog to a deterministic problem

Fermi & Ulam were studying neutron collisions at Los Alamos Scientific Laboratory
John von Neumann was in charge of simulating the experiment

What is a Monte Carlo?

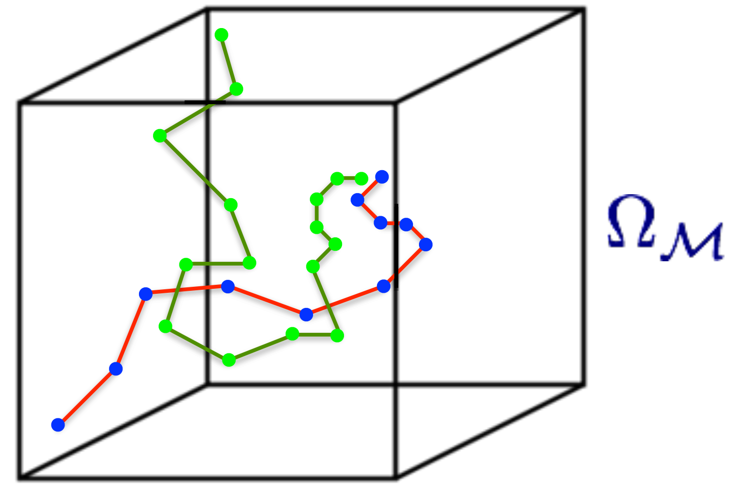
- Define a domain of input values
- Generate random inputs in the domain
- Perform a deterministic computation on the inputs
- Aggregate the results
- **Example**: calculate π from a geometrical figure



From geometry we know that the area of a square over that of a circle inscribed is pi/4. By sampling enough the two portions, we can easily measure the ratio by quantifying the number of sampling points in the two portions.

Markov chain Monte Carlo

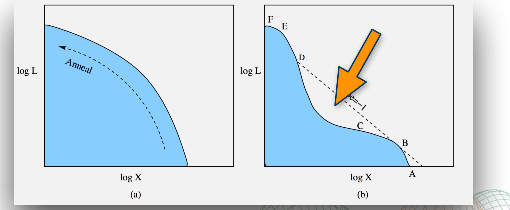
- **Random Walk** (analogy to Brownian motion) in the parameter space to find maximum likelihood value
- Each chain is characterized by a step size (length of each step) and by the initial point
- Accept new point if likelihood is better than previous one



We draw one point per time and proceed by drawing a new point and accepting it only if its likelihood is better than the previous one
We repeat this process with a another chain…and many more, until we sampled enough the parameter space

Markov chain Monte Carlo

- Problems in sampling distributions with multiple modes (maxima)
- Metropolis-Hastings algorithm (1953-1970) to improve this problem
- Can be adopted up to k ~ 100 but very critical configuration of chains
- Requires hundreds of chains running in parallel and typically on the order of millions of sampling points
- Bayesian Evidence is ***not*** computed in this process
- Still sensitive to phase changes in likelihood (multi-modal distributions)



Nested sampling Monte Carlo

$$p(\boldsymbol{\theta}) = \frac{\mathcal{L}(\boldsymbol{\theta})\,\pi(\boldsymbol{\theta})}{\mathcal{E}}$$

- Skilling 2004
- Convert evidence in a one-dimensional integral

$$\mathcal{E} = \int \mathcal{L}(\boldsymbol{\theta})\,\pi(\boldsymbol{\theta})\,d\boldsymbol{\theta}$$

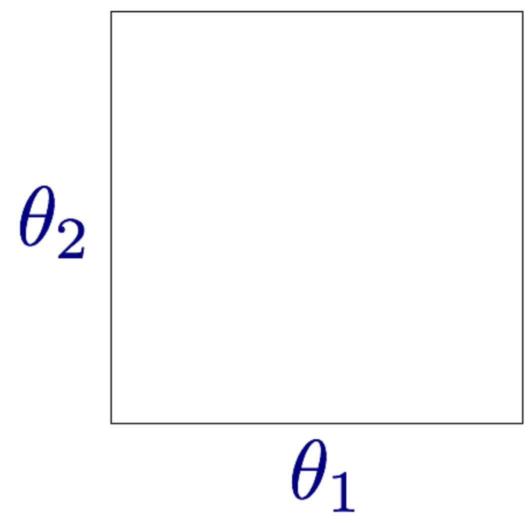
- Define small portion of prior volume (prior mass)

$$dX = \pi(\boldsymbol{\theta})\,d\boldsymbol{\theta}$$

$$\mathcal{E} = \int_0^1 \mathcal{L}(X)\,dX$$

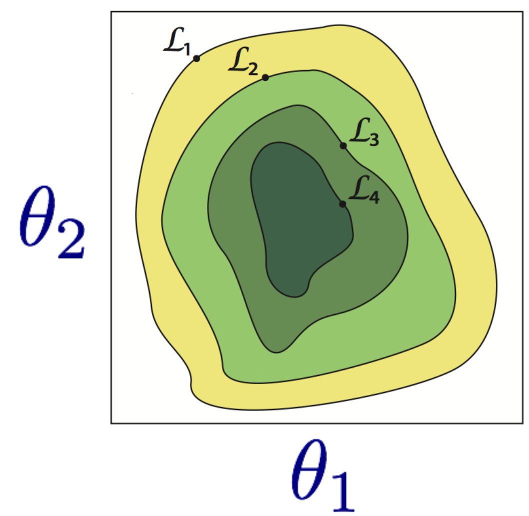
Recommended reading:
Skilling, J. 2004, AIP Conf. Proc., 735, 395

Nested sampling Monte Carlo



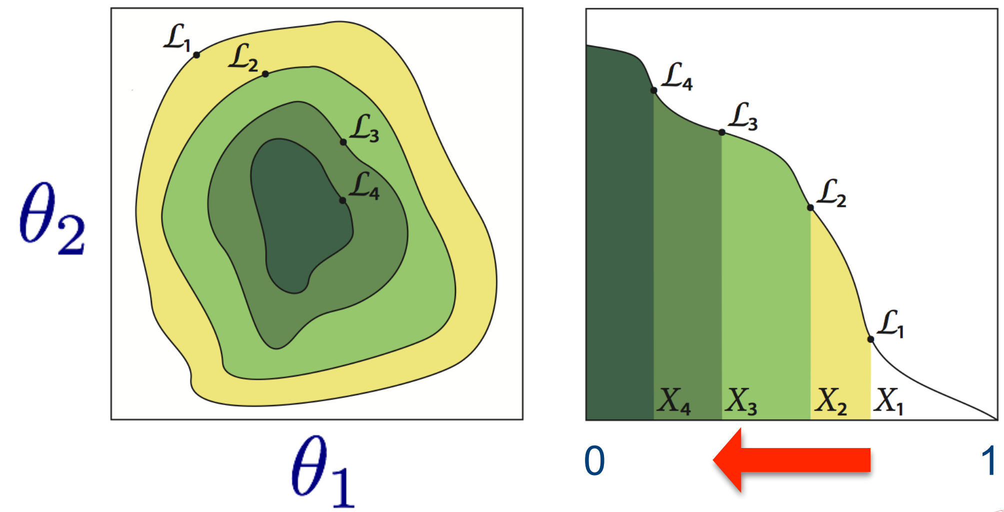
Imagine we have a bi-dimensional parameter space, with a given likelihood distribution. The entire parameter space corresponds to the total prior volume, namely X = 1.

Nested sampling Monte Carlo

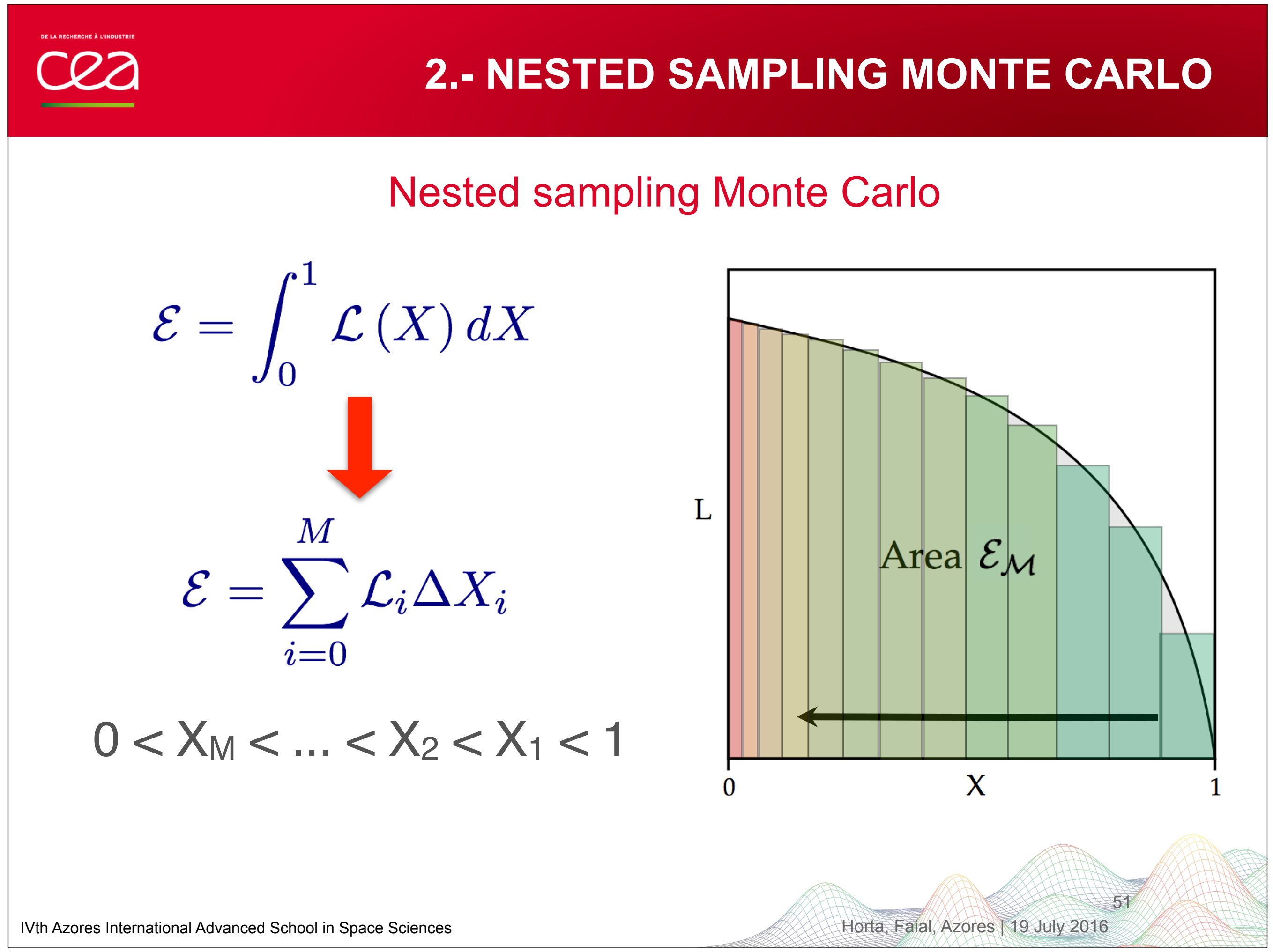


Nested sampling Monte Carlo

$$X(\mathcal{L}^*) = \int_{\mathcal{L}(\boldsymbol{\theta})>\mathcal{L}^*} \pi(\boldsymbol{\theta})d\boldsymbol{\theta} \qquad \int \pi(\boldsymbol{\theta})d\boldsymbol{\theta} = 1$$



As we shrink our constrain to a higher likelihood value, which corresponds to a smaller iso-likelihood contour, we also reduce the amount of prior volume that we are using, hence X is progressively reducing from 1 to 0.



If we can calculate L_i for each X_i then the computation of the Bayesian evidence is straightforward (either rectangular rule, or trapezoidal rule)

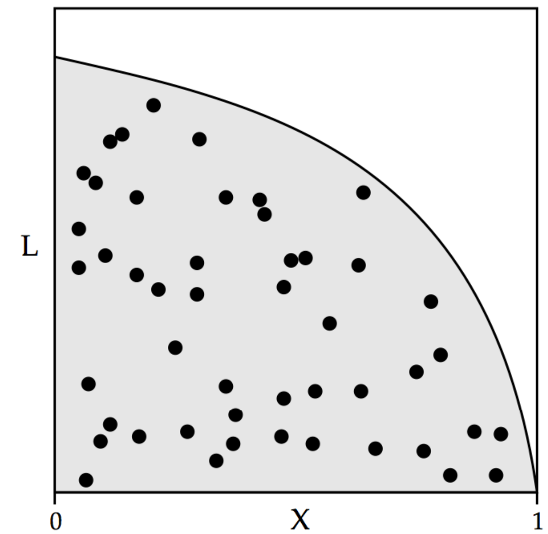
Nested sampling algorithm

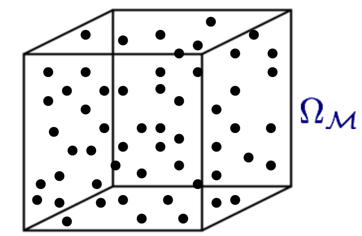
- How to obtain X? Sample randomly (Monte Carlo) from prior PDF
- Prior is known exactly (we set it *a priori*!) and can be sampled with high accuracy, no matter how many dimensions are considered
- This is obtained by sampling each free parameter according to its own prior PDF
- Initially obtain **J** different samples for each free parameter i.e. J points drawn from the total prior distribution
- Each of these points is called **live point**, and corresponds to a set of coordinates in the k-dimensional parameter space

$$\boldsymbol{\theta} = \{\theta_1, \theta_2, \ldots, \theta_k\}$$

Depending on the prior PDF we can draw a new parameter value accordingly. E.g. for a uniform prior the parameter value drawn can randomly lay within the entire range allowed with equal probability. If the prior is a Gaussian, we will have an over density of live points corresponding to the maximum of the Gaussian prior distribution.

Nested sampling algorithm

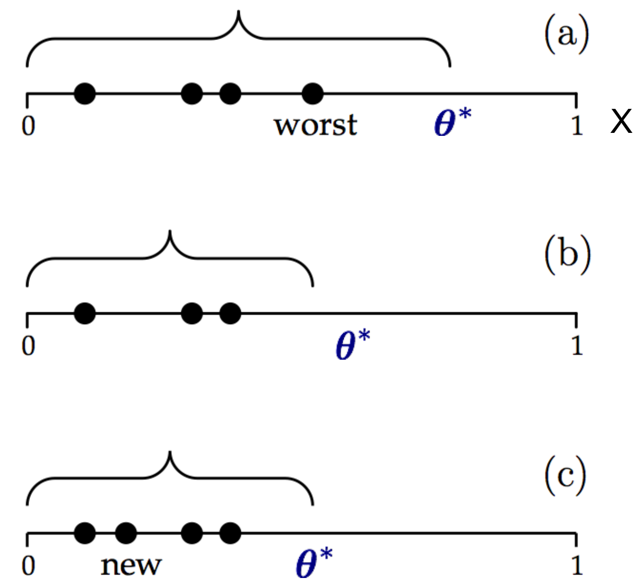




Each set of coordinates, i.e. live point, corresponds to a likelihood value, hence to a specific prior mass X. The initial sampling of live points will resemble the prior PDF in each dimension. If the prior PDF is uniform then the initial sampling will appear uniformly distributed over the entire parameter space, as shown in the box.

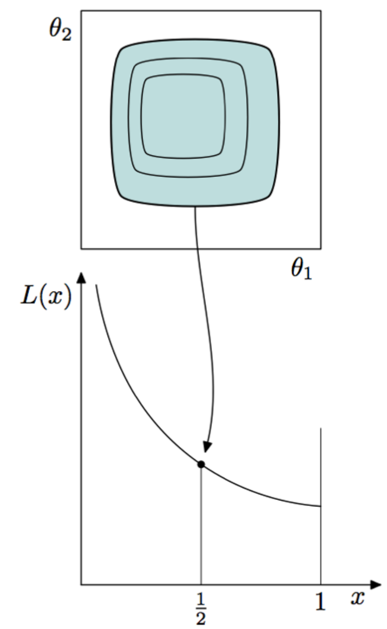
Nested sampling algorithm

- For each live point compute the likelihood value (straightforward), thus obtaining **J** likelihood values
- Take the worst likelihood point L_{worst}, in this sample and remove it from the sample
- Store the likelihood value and the live point removed
- Draw a new live point that satisfies the new constrain $L > L_{worst}$



Depending on the prior PDF we can draw a new parameter value accordingly. E.g. for a uniform prior the drawn parameter value can randomly lay within the entire range allowed, with equal probability.

Nested sampling algorithm

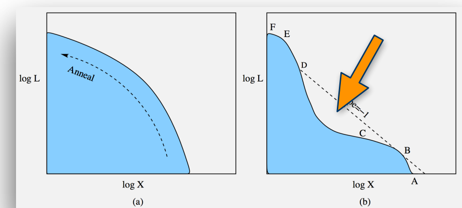


Each set of coordinates, i.e. live point, corresponds to a likelihood value, hence to a prior mass X. The new live point is drawn within the new, smaller region contained in a new iso-likelihood contour at a higher likelihood value. The process is repeated until we reach the maximum of the likelihood distribution.

Nested sampling algorithm: Advantages

- Distributions sampled efficiently and Bayesian evidence computation very accurate (typically requires **100** times less samples than thermodynamic integration to reach same accuracy, + error bar)
- Sampling method insensitive to any phase change in likelihood i.e. can sample very well multi-modal distributions
- Posterior probability values (required for parameter estimation) are a simple by product!

$$p_i = \frac{\mathcal{L}_i \Delta X_i}{\mathcal{E}}$$



Posterior probabilities (not probability densities) come from the definition of the Bayes' theorem and are an easy by-product of the sampling.

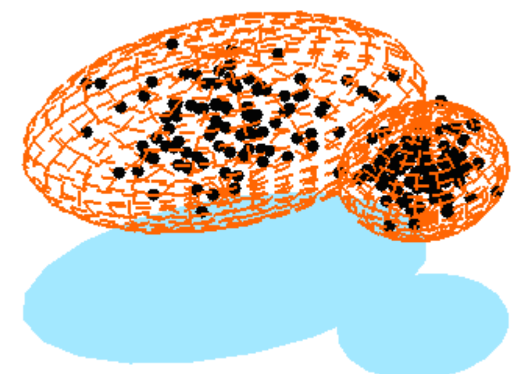
Nested sampling algorithm: Disadvantages

- Drawing a new live point satisfying the likelihood constraint is difficult
- Requires additional algorithms to improve the sampling efficiency
- Example: **ellipsoidal sampling**

Mukherjee P. et al. (2006; ApJ, 638, L51)

Feroz F., Hobson M. P. (2008; MNRAS, 384, 449)

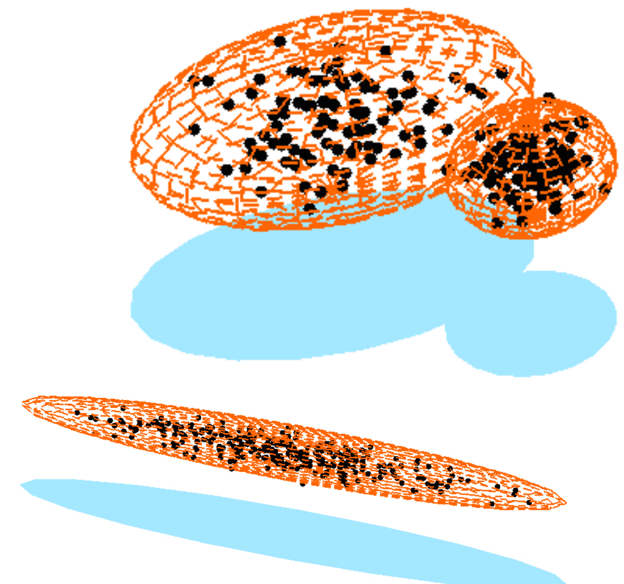
Feroz F., Hobson M. P., Bridges M. (2009; MNRAS, 398, 1601)



The ellipsoidal sampler is a possible way to afford the problem of sampling efficiently from a prior with the hard constraint of the likelihood value.
Other techniques involve the use of Markov chain Monte Carlo within the Nested sampling algorithm, or Galilean Monte Carlo which relies on the evaluation of gradients (derivatives) in the likelihood distribution to be able to sample higher likelihood regions more efficiently. Ellipsoids are computationally more efficient because they allow reducing the volume of prior space from which we sample new points.

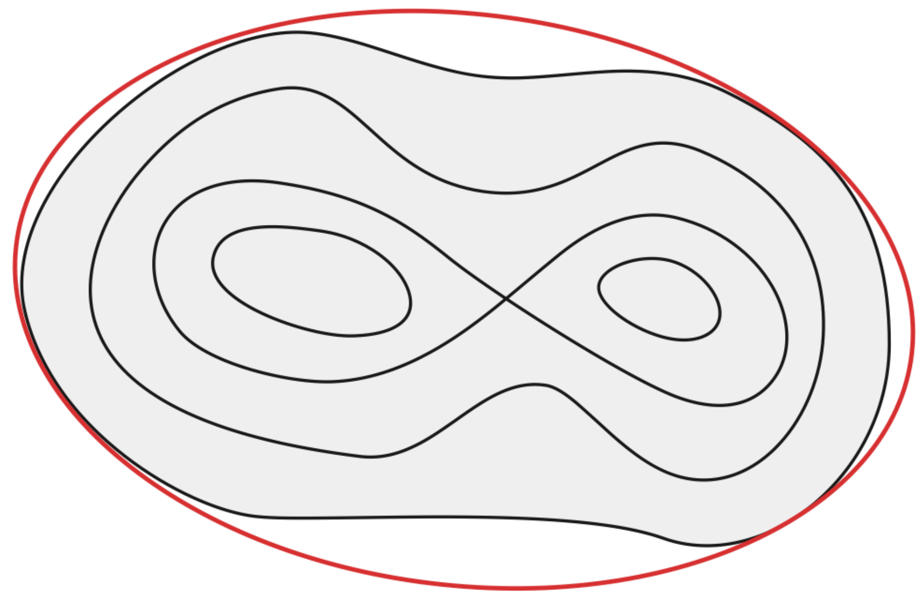
Ellipsoidal Sampling (ES)

- Find clusters of live points (X-means algorithm)
- Construct ellipsoids for each cluster
- Approximate parameter space with sets of ellipsoids to reduce prior volume to draw from
- We can draw from an ellipsoid very rapidly
- Draw a new point until we find one with a likelihood better than the constraint



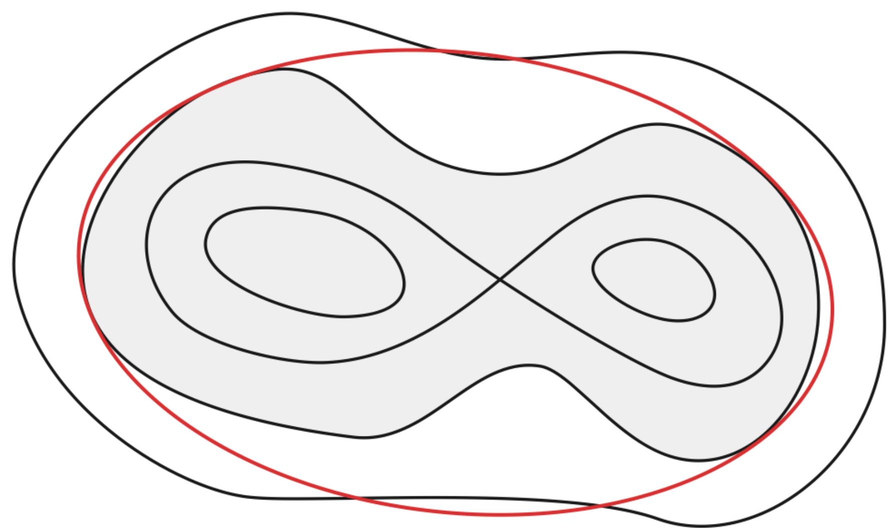
The ellipsoidal sampler is based on a cluster algorithm. The cluster algorithm allows identifying groups of points that are clustering around a common center of mass. During the nested sampling process, the algorithm checks how many clusters can be found and therefore computes a k-dimensional ellipsoid for each of them.

Ellipsoidal Sampling (ES)

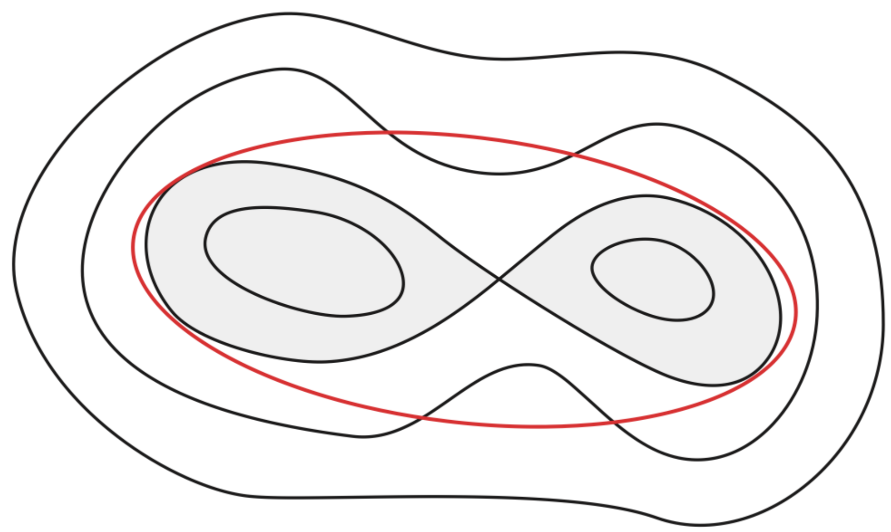


The animation shows an example of how ellipsoids shrink during the progress of the nested iterations to confine the local maxima of the likelihood distribution.

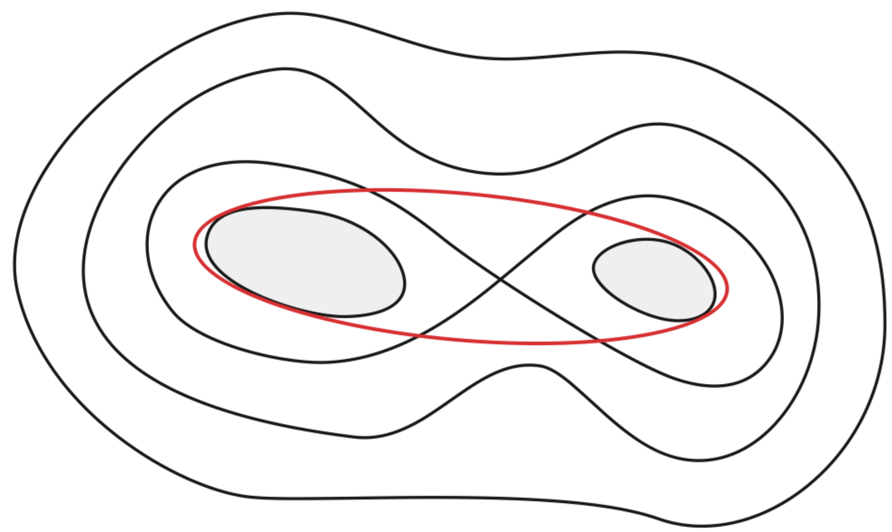
Ellipsoidal Sampling (ES)



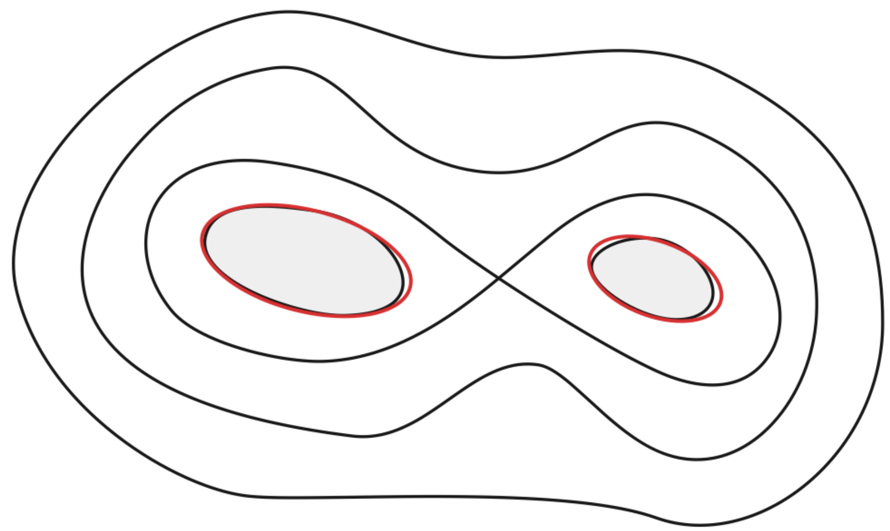
Ellipsoidal Sampling (ES)



Ellipsoidal Sampling (ES)



Ellipsoidal Sampling (ES)



1. - BAYESIAN STATISTICS

2. - NESTED SAMPLING MONTE CARLO

3. - THE DIAMONDS CODE

4. - FITTING THE BACKGROUND SIGNAL

5. - FITTING THE OSCILLATION MODES

6. - PEAK SIGNIFICANCE TEST

What is DIAMONDS?

high-DImensional And multi-MOdal NesteD Sampling

Corsaro & De Ridder 2014 A&A, 571, 71

DIAMONDS™
BAYESIAN SOFTWARE

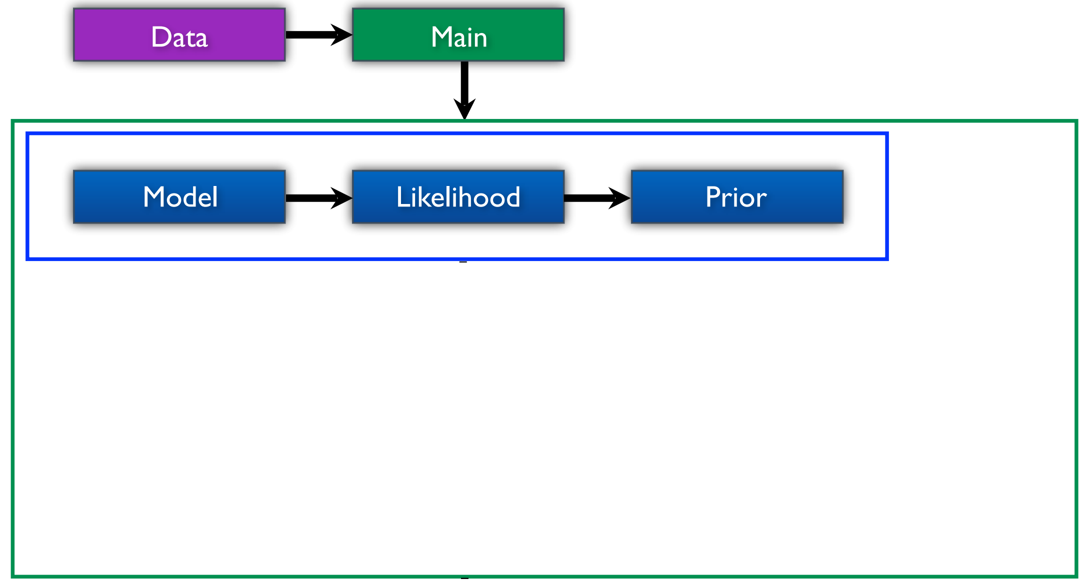
- C++11 code implementing NS with SES and X-means for **Bayesian inference** problems:
  - Dataset to fit
  - Model to test
  - Estimate the free parameters of the model and evaluate the statistical weight of the model

A Bayesian inference problem is happening every time you want to fit a dataset to extract some given information, e.g. constrain (fit) the parameter of a model.

What makes DIAMONDS so appealing?

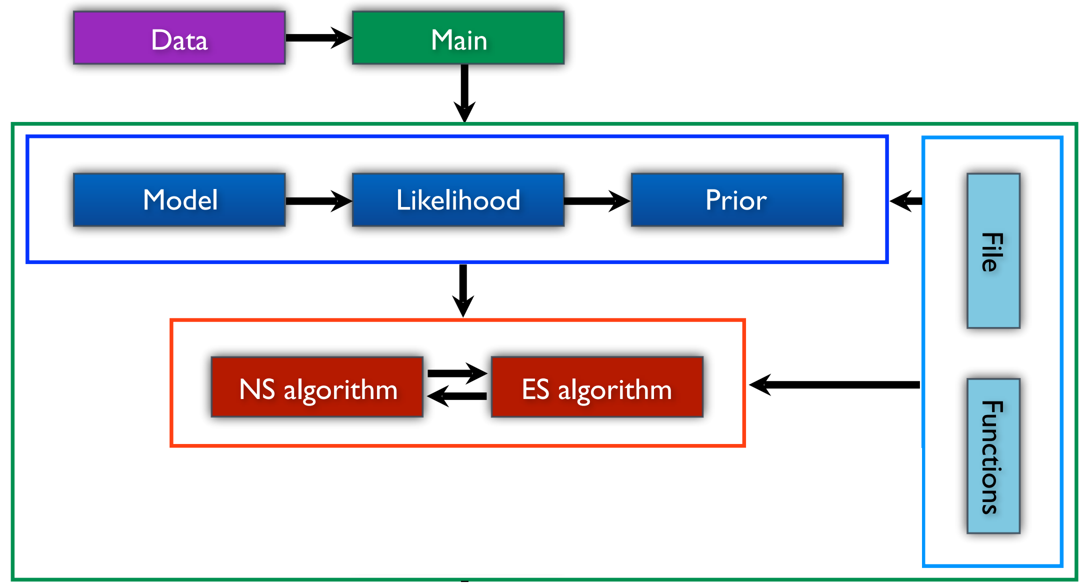
- Basic core **public** available
- General for **any** application involving Bayesian Inference
- **Bayesian evidence** (essential for model comparison problems) is a direct output
- Very powerful in identifying multiple (degenerate) solutions, also in high-dimensions
- Code implementation is **flexible** and easy to upgrade
- Different types of **prior** distributions and **likelihood** functions already provided
- **Overtakes** other existing NSMC codes (e.g. MultiNest, GMC, Polychord)
- Attracted already more than **60 users** from many world's institutions and different fields of physics

DIAMONDS Working Scheme

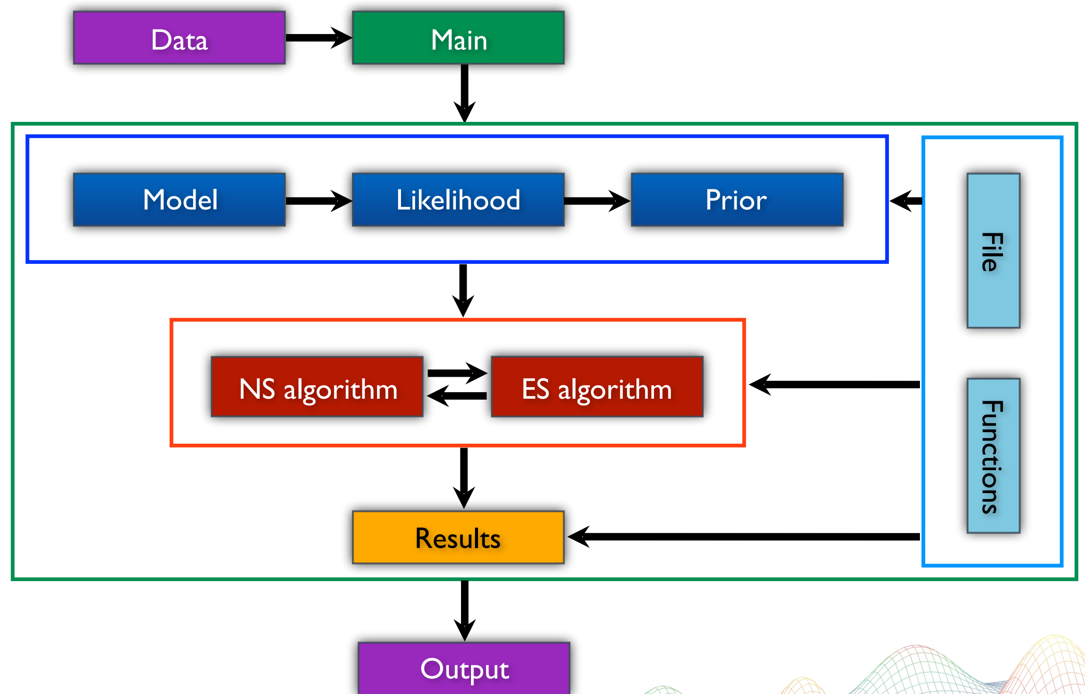


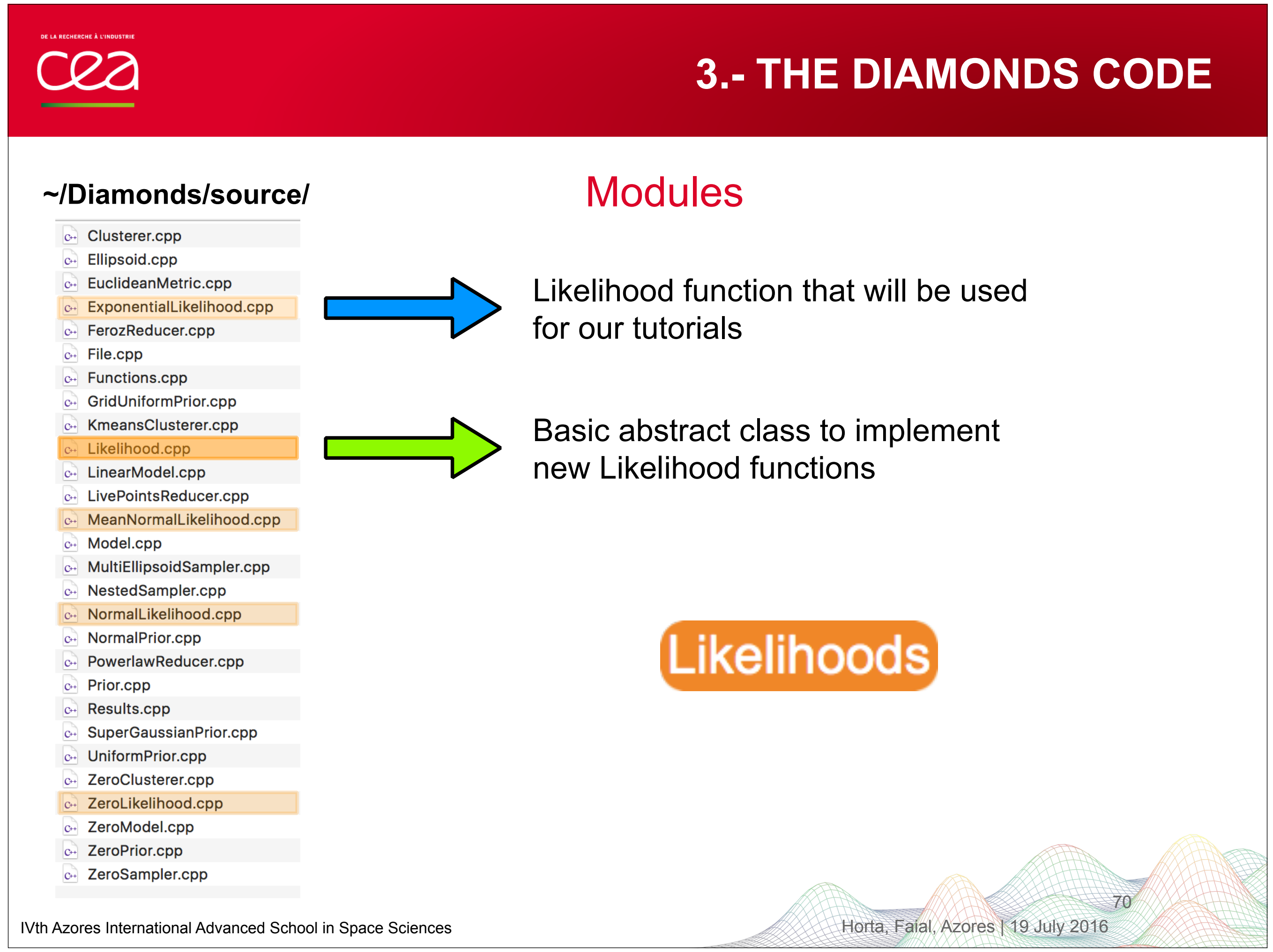
This is the general working flow of the code. You can feed in your own models, likelihoods, and priors. We will come back to that in the hands on computer session.

DIAMONDS Working Scheme

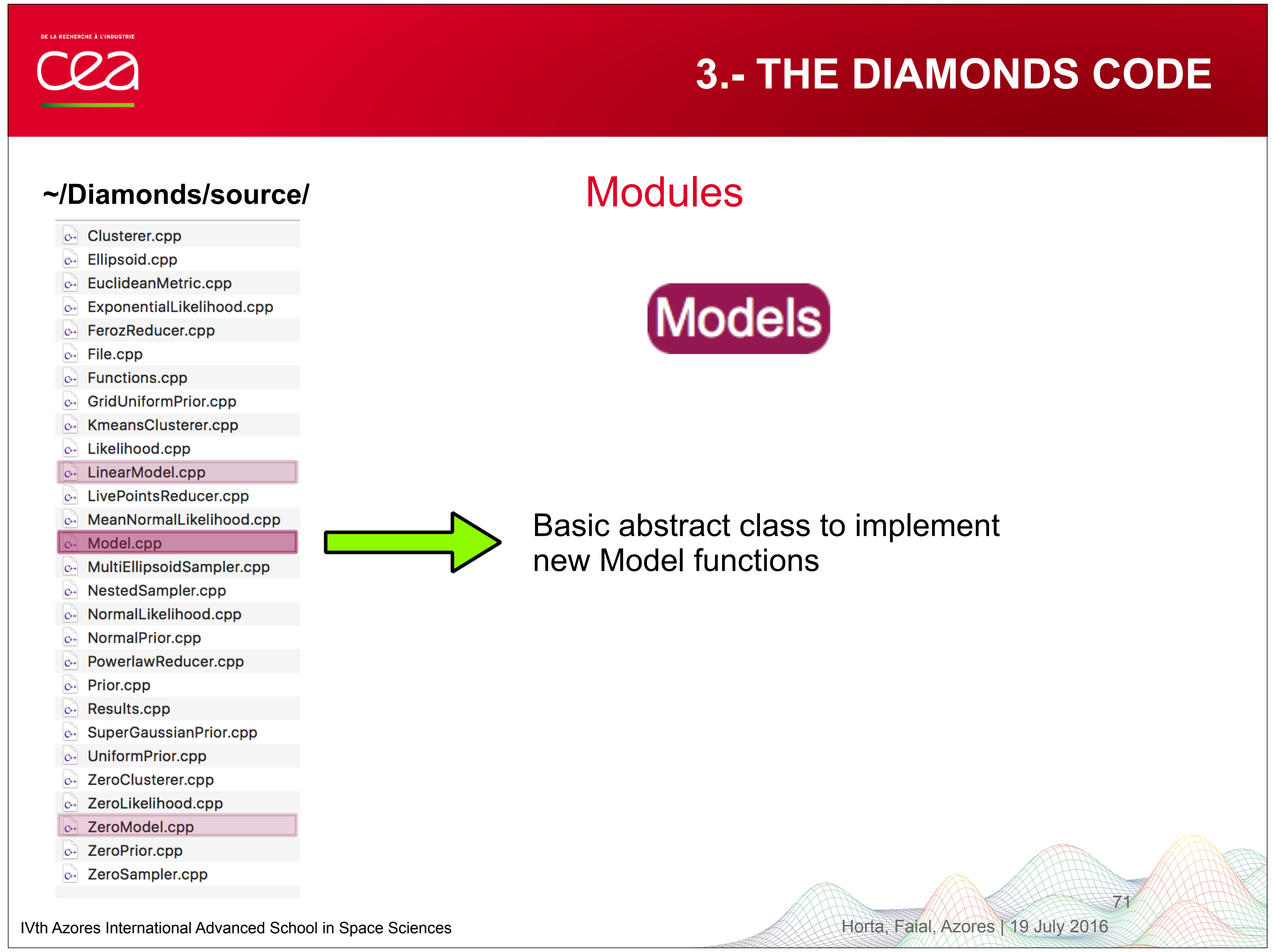


DIAMONDS Working Scheme

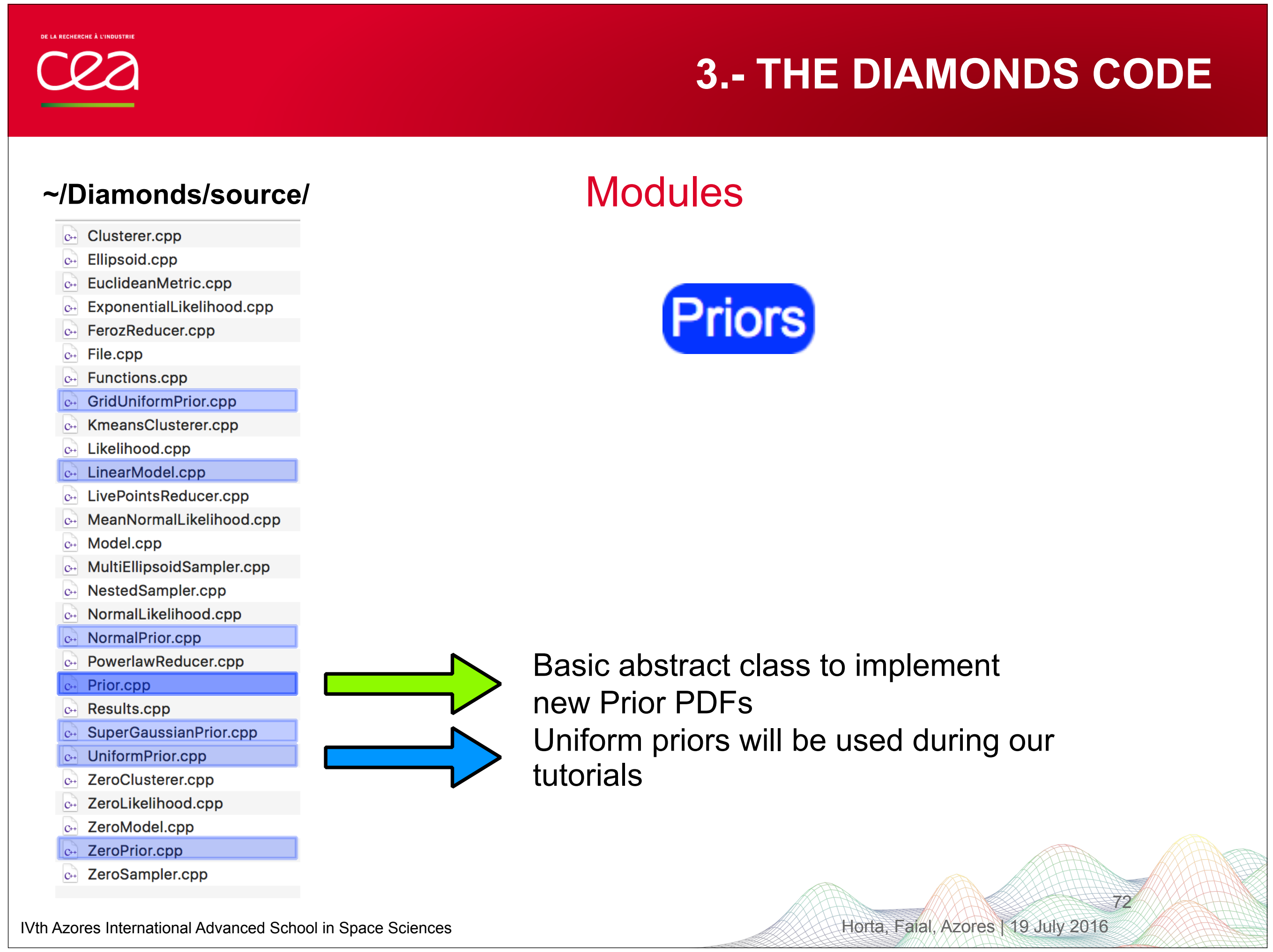




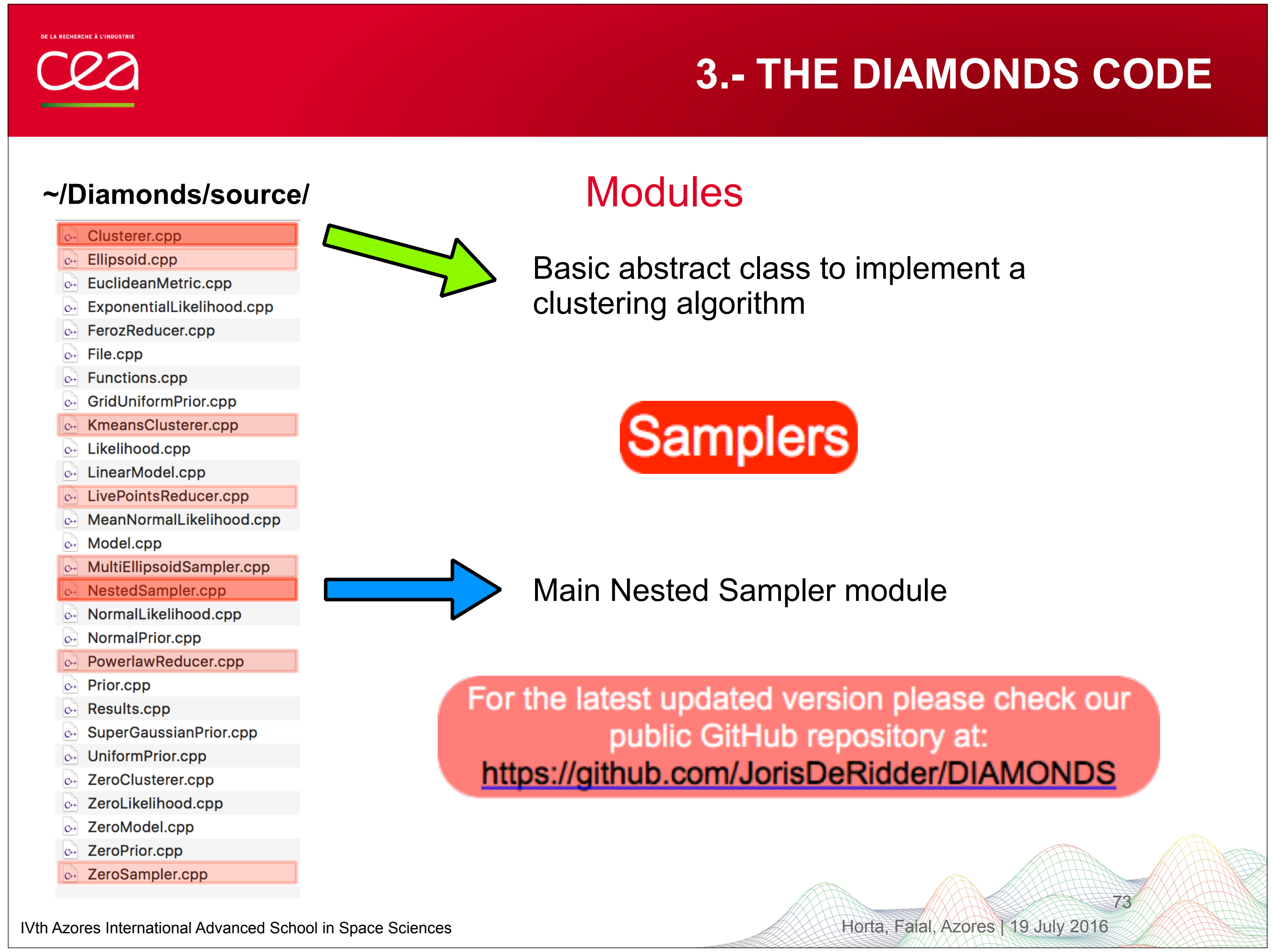
You have different likelihood functions already implemented in the basic package of the code. You can however implement your new ones by considering Likelihood.cpp as a source code template. All the header functions prototypes can be found in the folder ~/Diamonds/include/



You can implement your fitting model by considering Model.cpp as a source code template. The All the header functions prototypes can be found in the folder ~/Diamonds/include/



You have different prior PDFs already implemented in the basic package of the code. You can however implement your new ones by considering Prior.cpp as a source code template. All the header functions prototypes can be found in the folder ~/Diamonds/include/.
Implementing a new prior PDF is generally not an easy task. The prior classes provided will allow a wide applicability to astrophysical problems. Jeffreys' priors can be converted into uniform ones by considering the natural logarithm of the free parameter as the new free parameter.

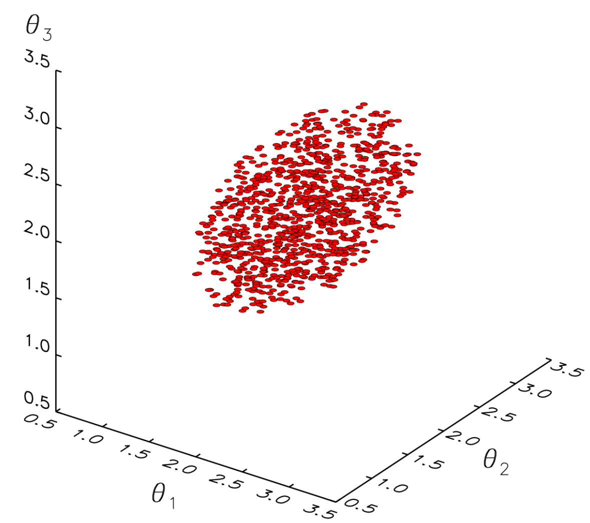


The modules related to the sampler are deeply linked to the working principle of the nested sampler. It is not straightforward to replace them with a different algorithm. I recommend to keep these modules as they are, and check for general updates in the public GitHub repository of the code. Replacing one of these modules will significantly affect the efficiency of the entire code. If you have downloaded the code from the website and you have provided a correct e-mail address then you will be notified whenever new releases will be available.

Prior PDFs

Corsaro & De Ridder 2014 A&A, 571, 71

3D Uniform

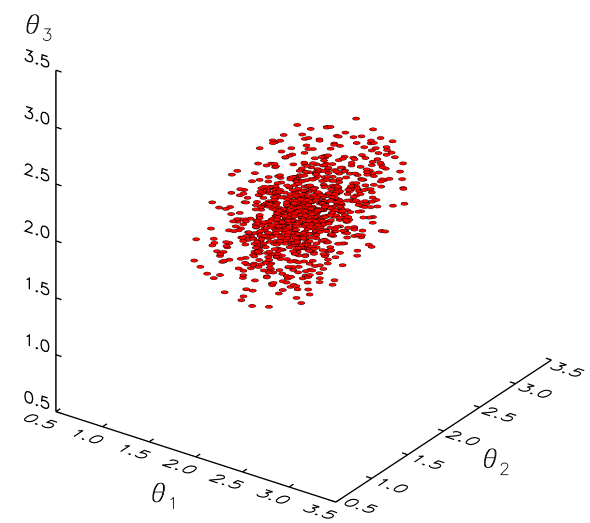


Each set of coordinates, i.e. live point, corresponds to a likelihood value, hence to a prior mass X.

Prior PDFs

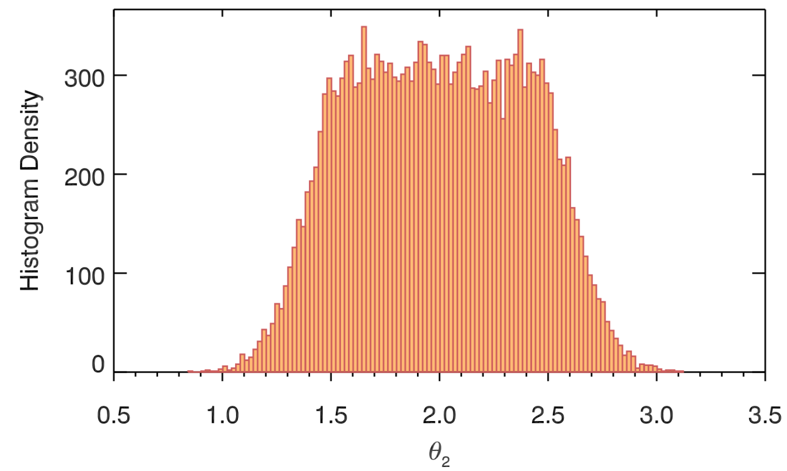
Corsaro & De Ridder 2014 A&A, 571, 71

3D Gaussian



Prior PDFs

Corsaro & De Ridder 2014 A&A, 571, 71



DIAMONDS BAYESIAN SOFTWARE

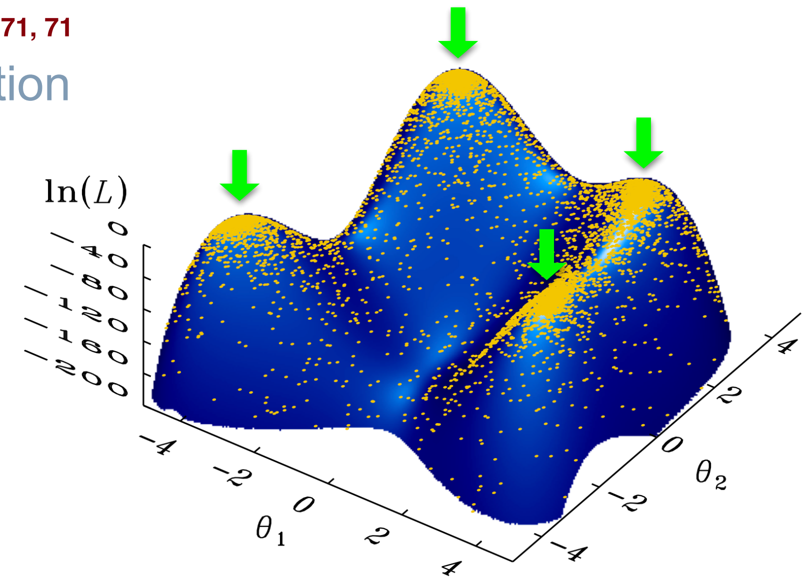
1D Super Gaussian

Sampling efficiency: Analytical likelihood distributions

Corsaro & De Ridder 2014 A&A, 571, 71

Himmelblau's Function

N = 8485 Samples



DIAMONDS™ BAYESIAN SOFTWARE

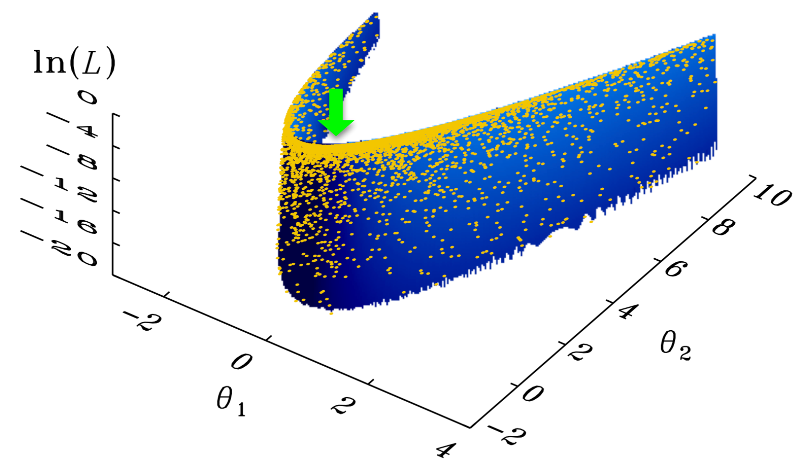
This is an example of a likelihood distribution (with known analytical equation) with four different maxima.

Sampling efficiency: Analytical likelihood distributions

Corsaro & De Ridder 2014 A&A, 571, 71

Rosenbrock's Function

N = 8558 Samples



DIAMONDS™ BAYESIAN SOFTWARE

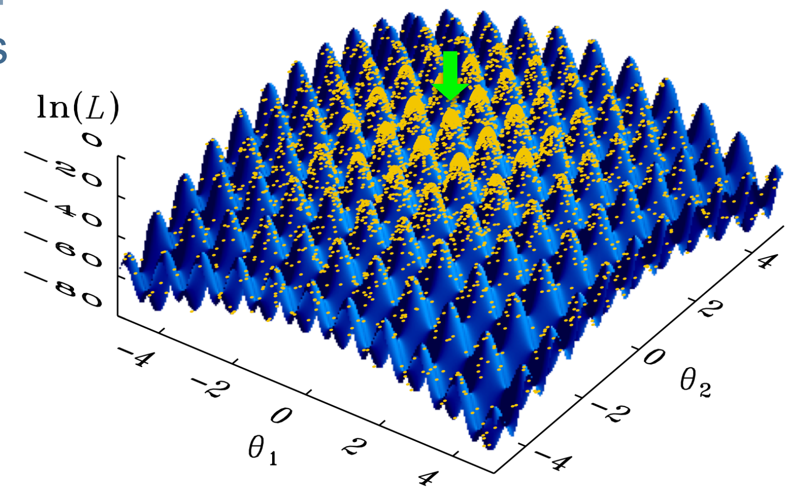
The Rosenbrock function has an hidden global maximum inside a pronounced curving degeneracy.

Sampling efficiency: Analytical likelihood distributions

Corsaro & De Ridder 2014 A&A, 571, 71

Rastrigin's Function

N = 10648 Samples



DIAMONDS™ BAYESIAN SOFTWARE

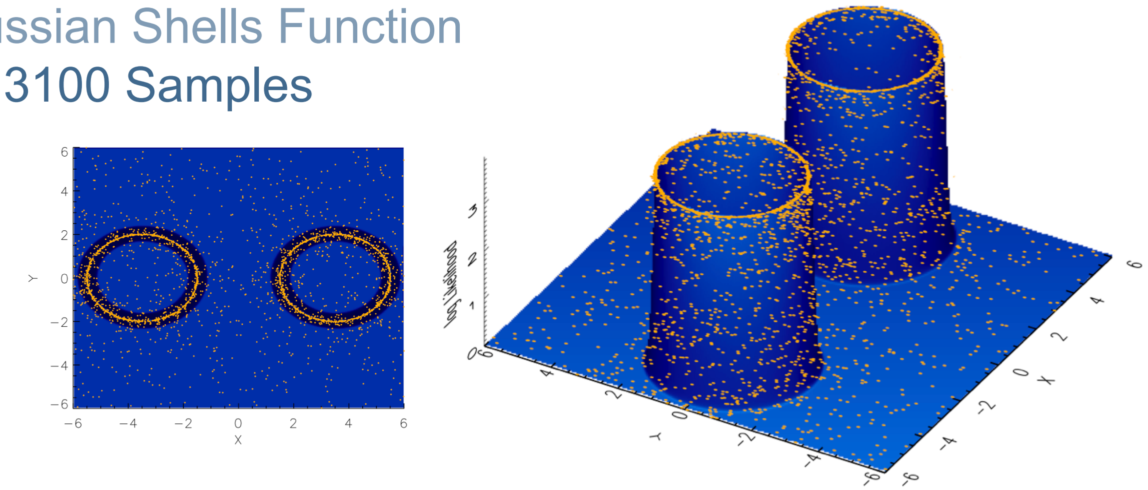
The Rastrigin's function is a nice example of a highly multimodal distribution with a global maximum. The DIAMONDS code proves to be able to quickly recover the position of the global maximum of the distribution.

Sampling efficiency: Analytical likelihood distributions

Corsaro & De Ridder 2014 A&A, 571, 71

Gaussian Shells Function

N = 3100 Samples

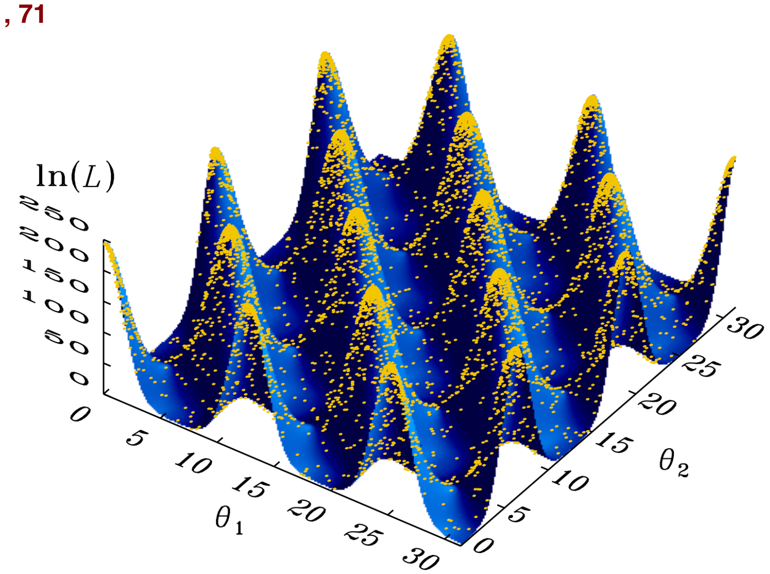


Gaussian Shells cylinders are very complex distributions to sample because of the very pronounced curving degeneracy, which is also difficult to access due to the cylindrical shape of these functions.

Sampling efficiency: Analytical likelihood distributions

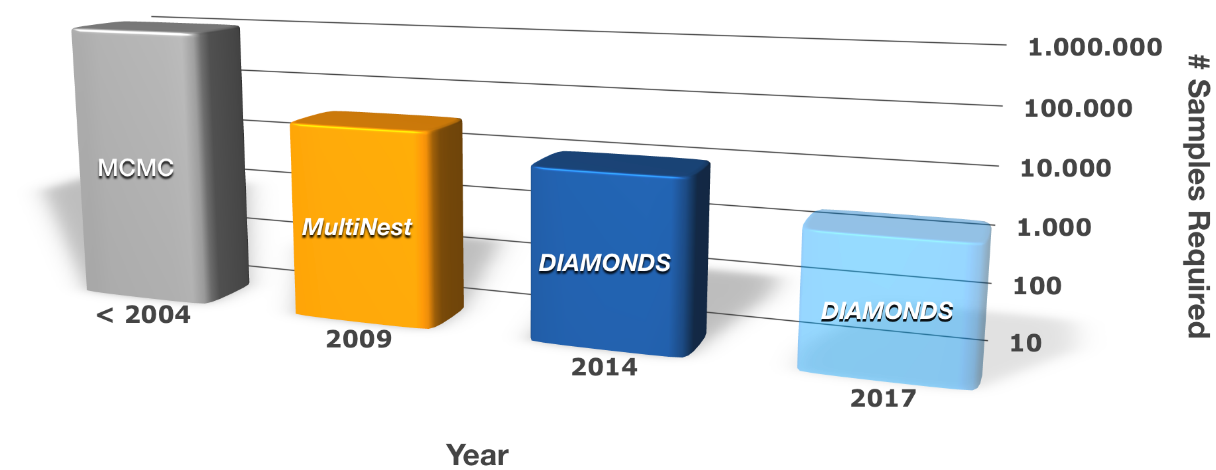
Corsaro & De Ridder 2014 A&A, 571, 71

Eggbox Function
N = 8207 Samples



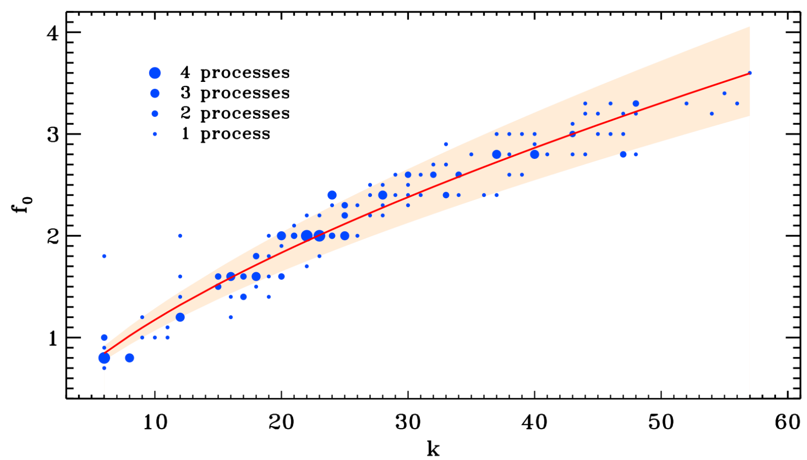
The Eggbox function is a highly complex distribution to be sampled efficiently because it contains a large number of local maxima. Standard MCMC algorithms do fail completely to sample it. NSMC can instead be able to provide a reliable sampling with a relatively low number of likelihood evaluations.

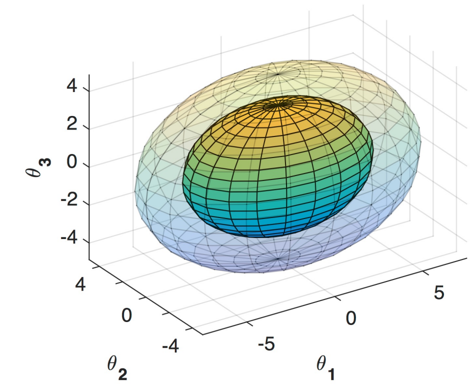
Sampling efficiency



Arbitrary scale of counts to show what can be gained in terms of number of samples with different algorithms.

SES Enlargement fraction

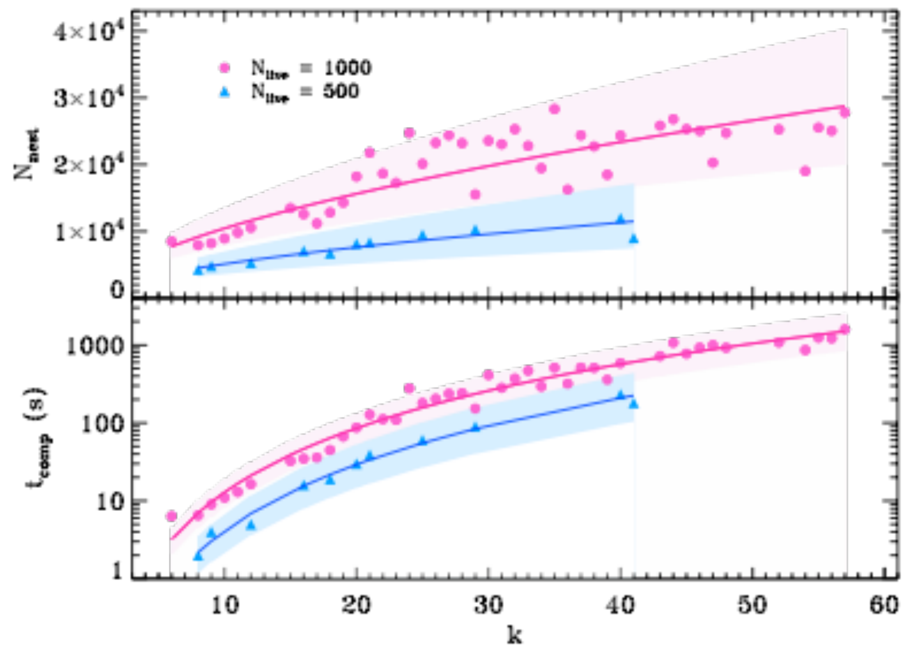




$$f_0 = (0.267 \pm 0.014)k^{0.643 \pm 0.017}$$

The enlargement fraction of the ellipsoids is a critical parameter in the nested sampling based on SES. It depends on the number of free parameters of the fitting model and has to be tuned accordingly in order to be able performing reliable samplings of the posterior probability distribution. The relation show here was calibrated using a set of 150 independent computations for the peak bagging analysis discussed later.

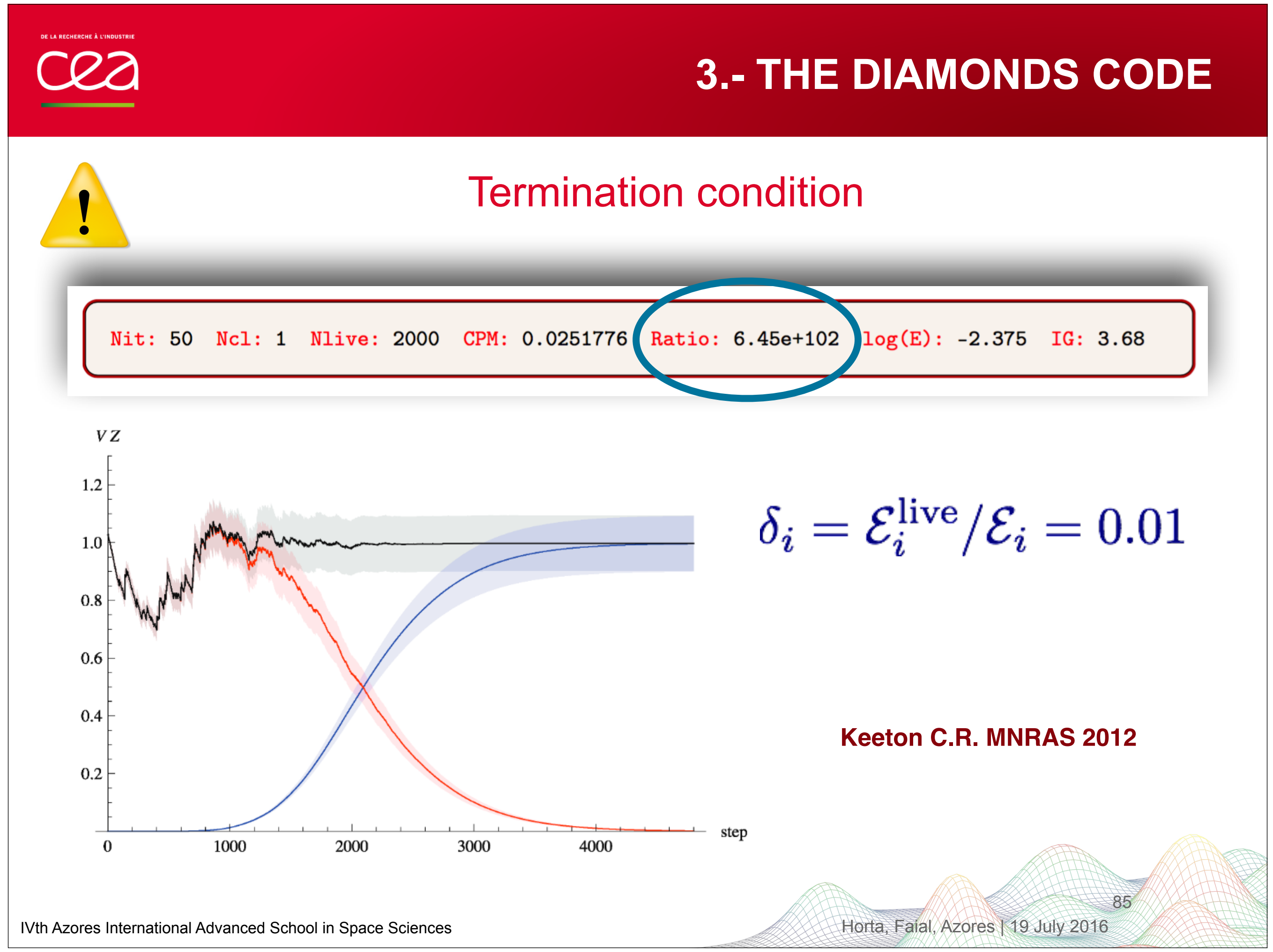
Computational efficiency and live points



$$N_{\rm nest} \sim \mathcal{O}(k^{3/5})$$

$$t_{\rm comp} \sim \mathcal{O}(k^{2.8})$$

The computational efficiency of the DIAMONDS code is shown in terms of the number of nested iterations required to converge to a solution, and the corresponding computational time, in this case referring to a 2.7 GHz single core computation and a dataset accounting for about 2000 data bins.



A detailed explanation of the meaning of the different numbers that are displayed during the computation can be found in the User Guide Manual of the code.
The live evidence E^live is an estimate of the remaining Bayesian evidence from the last set of live points. This number decreases as we proceed with the nested iterations (red curve) and we collect more evidence from the posterior distribution (blue curve). When reaching the termination condition value, the algorithm stops and computes the results.
The ratio has not be considered as a Bayes factor, because it is not the ratio of two difference Bayesian evidences (related to two different models). The ratio is only an estimate of how much evidence remains to be collected with respect to how much evidence has been collected already.

HANDS-ON TUTORIAL

- **~/Background/build/**
- **~/PeakBagging/build/**

Make sure you have edited the script file **multirun** using your local path and the file **localPath.txt** as well

/YOUR_LOCAL_PATH_HERE/

Before running the code make sure that you have edited the local path in the two files specified. You need to include the full path from the root, without any ~.

1. - BAYESIAN STATISTICS

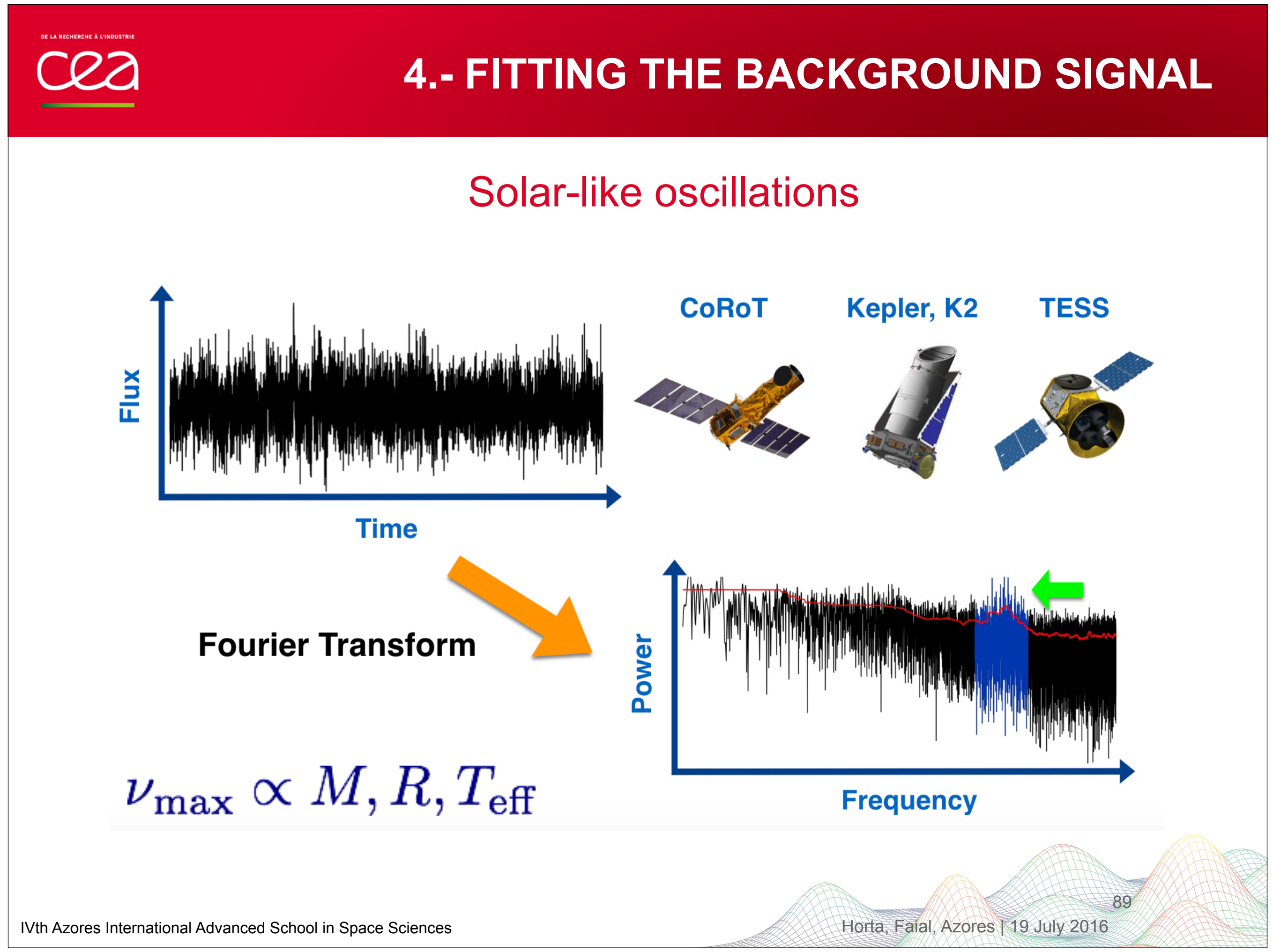
2. - NESTED SAMPLING MONTE CARLO

3. - THE DIAMONDS CODE

4. - FITTING THE BACKGROUND SIGNAL

5. - FITTING THE OSCILLATION MODES

6. - PEAK SIGNIFICANCE TEST

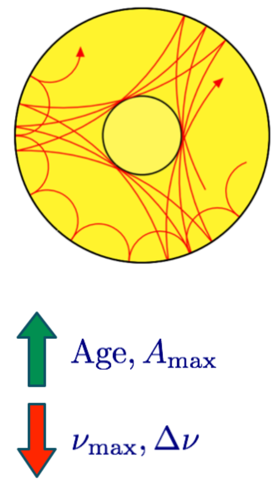


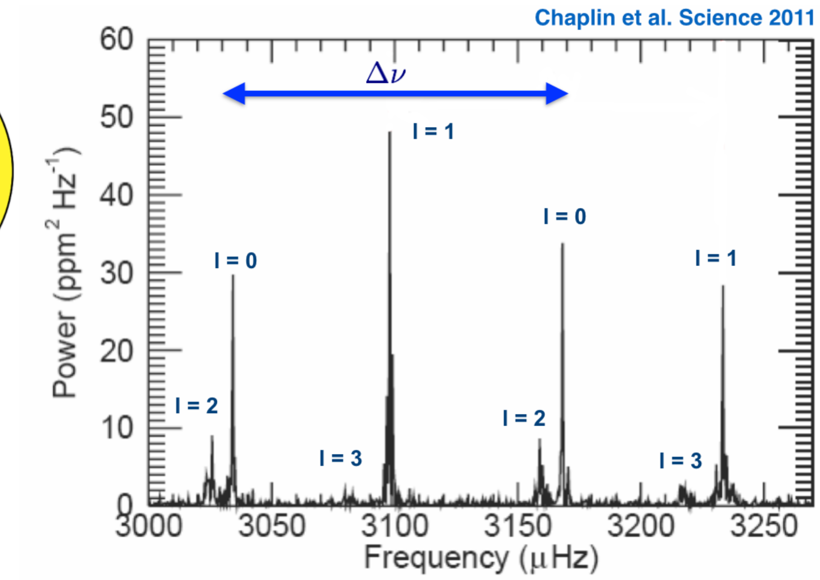
The Fourier analysis of a time-series will lead to a power spectrum, which plots the power of the signal (amplitude squared) versus the frequency. In general, we use a power spectral density (PSD), which is computed as the power divided by the frequency resolution. For photometric observations, a PSD is given in units of ppm^2/microHz.

In the case of solar-like oscillations, the PSD will display a Gaussian bump arising from a continuum background level.

Fine structure of solar-like oscillations

$$\Delta\nu \propto \bar{\rho}$$





The fine-structure of the oscillations shows the presence of a comb-like pattern that has a characteristic main frequency separation given as Deltanu, namely the frequency separation between two modes of the same angular degree l, but consecutive radial order.

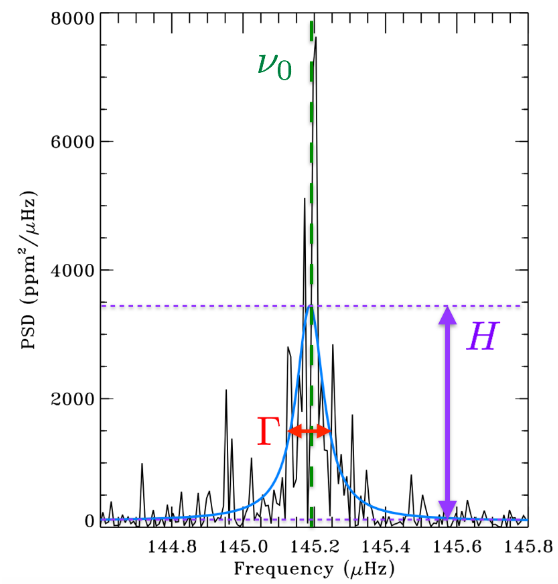
Scaling relations

$$\frac{M}{M_\odot} \simeq \left(\frac{\nu_{\rm max}}{\nu_{\rm max,\odot}}\right)^3 \left(\frac{\Delta\nu}{\Delta\nu_\odot}\right)^{-4} \left(\frac{T_{\rm eff}}{T_{\rm eff,\odot}}\right)^{1.5}$$

$$\frac{R}{R_\odot} \simeq \left(\frac{\nu_{\rm max}}{\nu_{\rm max,\odot}}\right) \left(\frac{\Delta\nu}{\Delta\nu_\odot}\right)^{-2} \left(\frac{T_{\rm eff}}{T_{\rm eff,\odot}}\right)^{0.5}$$

The scaling relations for mass and radius can be derived from the global asteroseismic parameters nu_max and Deltanu. However, they should be used with care because some systematics are present, which can lead to deviations up to 15% in mass, and 5% in radius.

Fine structure of solar-like oscillations



$\nu_{\mathrm{n,l,m}}$

Damped oscillation

Lorentzian profile

$$T_{\mathrm{obs}} \gg \tau$$

$$\Gamma \propto \tau^{-1}$$

$$A^2 = \pi H \Gamma$$

$$\boxed{\nu_0, \Gamma, A}$$

The structure of an individual acoustic oscillation mode is represented by a Lorentzian profile, which arises from the damped oscillation. In our fitting problem, we will fit the amplitude of the Lorentzian (corresponding to the integral over frequency of the profile). This is because the amplitude A is not much correlated to the line width Gamma, thus allowing us to simplify the fitting process with DIAMONDS.

4.- FITTING THE BACKGROUND SIGNAL

To-do list

- Open your terminal and cd **~/Background/build/**
- Make sure you have edited the script file **multirun** using your local path and the file **localPath.txt** as well
- Run the script via terminal for the star KIC 12008916 as
  ./multirun 012008916 0 1

KIC ID with 9 digits

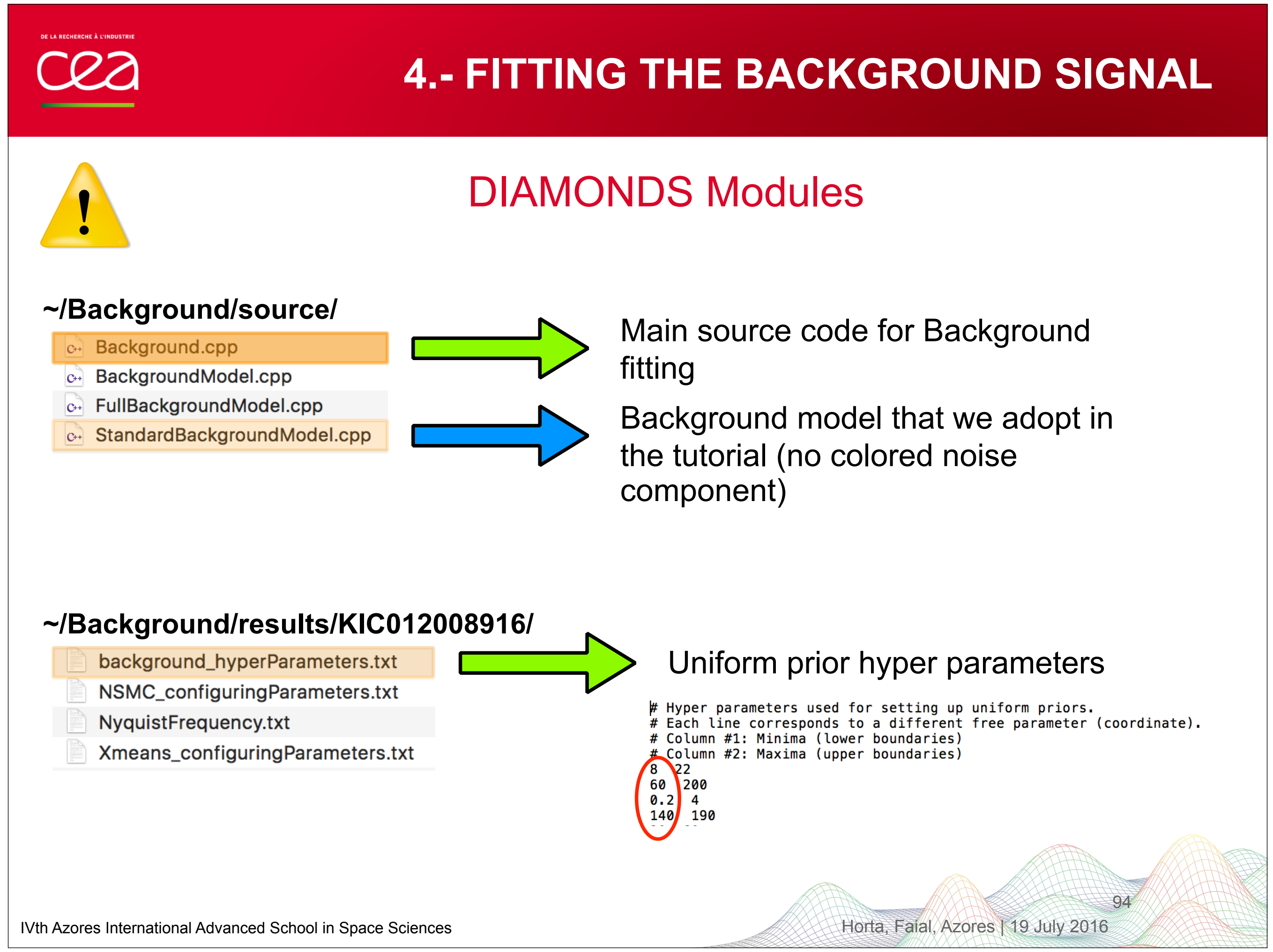
Number of independent runs that can be executed with a single command. We keep this number to **1** for all our tutorials.

Label of the starting run
If 0, it will create a folder labeled 00 inside the folder
Background/results/KIC012008916/

Diamonds is used as a library for the Background fitting code. The run number is important to keep track of your fitting outputs. All the outputs will be stored in the folder of the run, as you labeled it to execute the code.

In case you cannot run the script for a ‘newline’ error, you can still run the session by creating yourself a subfolder labeled 00 under the folder ~/Background/results/KIC012008916/, and then from the directory ~/Background/build/ run the command via terminal

./background 012008916 0



The background_hyperParameters.txt file is the most important one to consider. Its content will depend on the star. A good technique to get reliable priors for the background is that provided by Kallinger et al. 2016. The Nyquist frequency has to be set according to the cadence of the data taken into account.
In general, the NSMC and X-means parameters should not be changed, at least for the applications that we encounter in this tutorial.

The background model equation for a *Kepler* star

$$H_{\text{osc}} \exp\left[-\frac{(\nu-\nu_{\max})^2}{2\sigma_{\text{env}}^2}\right]$$

$$G(\nu)$$

Gaussian envelope oscillations

$$\sum_{i=1}^{3} \frac{\xi a_i^2/b_i}{1+(\nu/b_i)^4}$$

$$B(\nu)$$

super-Lorentzian profiles

W Photon noise (flat)

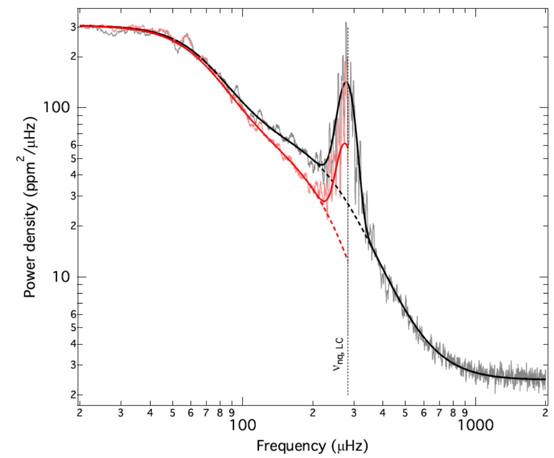
The StandardBackgroundModel consists of a 10 free parameters model. No colored noise is included.

The background model equation

$$P_{\mathrm{bkg}}(\nu) = [G(\nu) + B(\nu)]R(\nu) + W$$

$$R(\nu) = \mathrm{sinc}^2\left(\frac{\pi\nu}{2\nu_{\mathrm{Nyq}}}\right)$$

Apodization - Decrease of the signal for discrete time observations



Kallinger et al. 2014

The apodization is the degradation of the signal due to the finite sampling occurring in the time domain. It has to be taken into account to avoid estimating wrong values of the parameters of the model. This is seen as a convolution in the time domain, which is therefore a product in the frequency domain. The white noise is not affected by this degradation because it is related to the instrumental noise.

The background model in DIAMONDS

```
// Initialize global parameters

double flatNoiseLevel = modelParameters(0);
double amplitudeHarvey1 = modelParameters(1);
double frequencyHarvey1 = modelParameters(2);
double amplitudeHarvey2 = modelParameters(3);
double frequencyHarvey2 = modelParameters(4);
double amplitudeHarvey3 = modelParameters(5);
double frequencyHarvey3 = modelParameters(6);
double heightOscillation = modelParameters(7);
double nuMax = modelParameters(8);
double sigma = modelParameters(9);

// Compute Harvey components and add them to the predictions

double zeta = 2.0*sqrt(2.0)/Functions::PI;
predictions = zeta*amplitudeHarvey1*amplitudeHarvey1/(frequencyHarvey1*(1.0 + (covariates/frequencyHarvey1).pow(4)));
predictions += zeta*amplitudeHarvey2*amplitudeHarvey2/(frequencyHarvey2*(1.0 + (covariates/frequencyHarvey2).pow(4)));
predictions += zeta*amplitudeHarvey3*amplitudeHarvey3/(frequencyHarvey3*(1.0 + (covariates/frequencyHarvey3).pow(4)));


// Compute Gaussian envelope for RGB Oscillations and add it to the predictions

predictions += heightOscillation * exp(-1.0*(nuMax - covariates)*(nuMax - covariates)/(2.0 * sigma * sigma));


// Modulate the model by the response function (apodization)

predictions *= responseFunction;


// Add flat noise level component

predictions += flatNoiseLevel;
```


The implementation of the background model can be found in the file StandardBackgroundModel.cpp under the folder ~/Background/source/. The function that computes the predictions is called predict. The first bit of the code show here represents the initialization of the different free parameters, in the same order as they are defined in the prior distributions.

The background model in DIAMONDS

```
// Initialize global parameters

double flatNoiseLevel = modelParameters(0);
double amplitudeHarvey1 = modelParameters(1);
double frequencyHarvey1 = modelParameters(2);
double amplitudeHarvey2 = modelParameters(3);
double frequencyHarvey2 = modelParameters(4);
double amplitudeHarvey3 = modelParameters(5);
double frequencyHarvey3 = modelParameters(6);
double heightOscillation = modelParameters(7);
double numax = modelParameters(8);
double sigma = modelParameters(9);

// Compute Harvey components and add them to the predictions

double zeta = 2.0*sqrt(2.0)/Functions::PI;
predictions = zeta*amplitudeHarvey1*amplitudeHarvey1/(frequencyHarvey1*(1.0 + (covariates/frequencyHarvey1).pow(4)));
predictions += zeta*amplitudeHarvey2*amplitudeHarvey2/(frequencyHarvey2*(1.0 + (covariates/frequencyHarvey2).pow(4)));
predictions += zeta*amplitudeHarvey3*amplitudeHarvey3/(frequencyHarvey3*(1.0 + (covariates/frequencyHarvey3).pow(4)));


// Compute Gaussian envelope for RGB Oscillations and add it to the predictions

predictions += heightOscillation * exp(-1.0*(nuMax - covariates)*(nuMax - covariates)/(2.0 * sigma * sigma));

// Modulate the model by the response function (apodization)

predictions *= responseFunction;


// Add flat noise level component

predictions += flatNoiseLevel;
```


The second bit shows the implementation of the analytical equations that constitute the super-Lorentzian profiles and the Gaussian envelope of the oscillations.

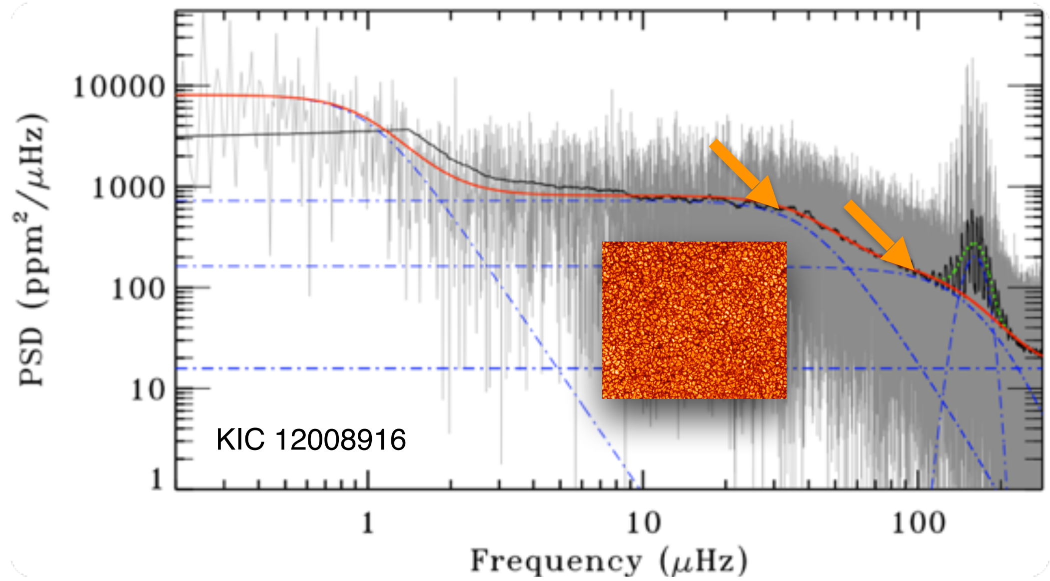
The background model in DIAMONDS

```
// Initialize global parameters

double flatNoiseLevel = modelParameters(0);
double amplitudeHarvey1 = modelParameters(1);
double frequencyHarvey1 = modelParameters(2);
double amplitudeHarvey2 = modelParameters(3);
double frequencyHarvey2 = modelParameters(4);
double amplitudeHarvey3 = modelParameters(5);
double frequencyHarvey3 = modelParameters(6);
double heightOscillation = modelParameters(7);
double nuMax = modelParameters(8);
double sigma = modelParameters(9);


// Compute Harvey components and add them to the predictions

double zeta = 2.0*sqrt(2.0)/Functions::PI;
predictions = zeta*amplitudeHarvey1*amplitudeHarvey1/(frequencyHarvey1*(1.0 + (covariates/frequencyHarvey1).pow(4)));
predictions += zeta*amplitudeHarvey2*amplitudeHarvey2/(frequencyHarvey2*(1.0 + (covariates/frequencyHarvey2).pow(4)));
predictions += zeta*amplitudeHarvey3*amplitudeHarvey3/(frequencyHarvey3*(1.0 + (covariates/frequencyHarvey3).pow(4)));


// Compute Gaussian envelope for RGB Oscillations and add it to the predictions

predictions += heightOscillation * exp(-1.0*(nuMax - covariates)*(nuMax - covariates)/(2.0 * sigma * sigma));

// Modulate the model by the response function (apodization)

predictions *= responseFunction;


// Add flat noise level component

predictions += flatNoiseLevel;
```


The last bit is multiplying all the components by the apodization signal and is adding a white noise level.

The background model fit for a *Kepler* star

Corsaro, De Ridder, García 2015 A&A, 579, 83



Results

- Open a new tab in your terminal and
  cd **~/Background/results/KIC012008916/00/**

```
background_configuringParameters.txt  background_marginalDistribution003.txt background_parameter000.txt  background_parameter007.txt
background_evidenceInformation.txt    background_marginalDistribution004.txt background_parameter001.txt  background_parameter008.txt
background_logLikelihood.txt          background_marginalDistribution005.txt background_parameter002.txt  background_parameter009.txt
background_logWeights.txt             background_marginalDistribution006.txt background_parameter003.txt  background_parameterSummary.txt
background_marginalDistribution000.txt background_marginalDistribution007.txt background_parameter004.txt background_posteriorDistribution.txt
background_marginalDistribution001.txt background_marginalDistribution008.txt background_parameter005.txt
background_marginalDistribution002.txt background_marginalDistribution009.txt background_parameter006.txt
```

ASCII file storing all the configuring parameters used for the run. From the NSMC to the X-means, and ellipsoidal sampler, and more… Useful to keep track of the run configuration.

```
# List of configuring parameters used for the NSMC.
# Row #1: Ndimensions
# Row #2: Initial (Maximum) Nobjects
# Row #3: Minimum Nobjects
# Row #4: NinitialIterationsWithoutClustering
# Row #5: NiterationsWithSameClustering
# Row #6: maxNdrawAttempts
# Row #7: terminationFactor
# Row #8: Niterations
# Row #9: Optimal Niterations
# Row #10: Final Nclusters
# Row #11: Final Nobjects
# Row #12: Computational Time (seconds)
10
500
500
1500
50
```

background_hyperParametersUniform.txt

The additional file background_hyperParametersUniform.txt is a copy of the input prior file background_hyperParameters.txt provided in the folder ~/Background/results/KIC012008916/. It is useful to keep track of which prior distributions you have used for your run session.

4.- FITTING THE BACKGROUND SIGNAL

Results

- Open a new tab in your terminal and
  cd **~/Background/results/KIC012008916/00/**

```
background_configuringParameters.txt   background_marginalDistribution003.txt background_parameter000.txt   background_parameter007.txt
background_evidenceInformation.txt     background_marginalDistribution004.txt background_parameter001.txt   background_parameter008.txt
background_logLikelihood.txt           background_marginalDistribution005.txt background_parameter002.txt   background_parameter009.txt
background_logWeights.txt              background_marginalDistribution006.txt background_parameter003.txt   background_parameterSummary.txt
background_marginalDistribution000.txt background_marginalDistribution007.txt background_parameter004.txt   background_posteriorDistribution.txt
background_marginalDistribution001.txt background_marginalDistribution008.txt background_parameter005.txt
background_marginalDistribution002.txt background_marginalDistribution009.txt background_parameter006.txt
```

The Bayesian evidence of the model that has been fit to the data, its error bar, and the information gain.

```
# Evidence results from nested sampling
# Skilling's log(Evidence)          Skilling's Error log(Evidence)          Skilling's Information Gain
-2.102371509e+05                          1.855036708e-01                        1.720580593e+01
```


The Bayesian evidence is expressed in natural logarithm because it is generally either a very larger or very small value. Remember that a Bayesian evidence alone does not mean anything and it is not useful. To use the Bayesian evidence you need to compare it with another one, coming from a different model that was fit to the same dataset.

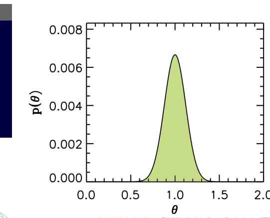
Results

- Open a new tab in your terminal and cd **~/Background/results/KIC012008916/00/**

```
background_configuringParameters.txt     background_marginalDistribution003.txt background_parameter000.txt     background_parameter007.txt
background_evidenceInformation.txt       background_marginalDistribution004.txt background_parameter001.txt     background_parameter008.txt
background_logLikelihood.txt             background_marginalDistribution005.txt background_parameter002.txt     background_parameter009.txt
background_logWeights.txt                background_marginalDistribution006.txt background_parameter003.txt     background_parameterSummary.txt
background_marginalDistribution000.txt   background_marginalDistribution007.txt background_parameter004.txt     background_posteriorDistribution.txt
background_marginalDistribution001.txt   background_marginalDistribution008.txt background_parameter005.txt
background_marginalDistribution002.txt   background_marginalDistribution009.txt background_parameter006.txt
```

Marginal probability distributions for each free parameter in the same order as they are defined in the priors. The values are sorted by increasing parameter value.

```
# Marginal distribution of cubic-spline interpolated points from nested sampling.
# Column #1: Parameter values
# Column #2: Marginal distribution values (probability only)
8.098578398e+00  6.340846004e-124
8.108505950e+00  3.415302968e-28
8.118433503e+00  6.623617878e-28
8.128361055e+00  9.417956670e-28
```



Results

- Open a new tab in your terminal and cd **~/Background/results/KIC012008916/00/**

```
background_configuringParameters.txt  background_marginalDistribution003.txt background_parameter000.txt  background_parameter007.txt
background_evidenceInformation.txt    background_marginalDistribution004.txt background_parameter001.txt  background_parameter008.txt
background_logLikelihood.txt          background_marginalDistribution005.txt background_parameter002.txt  background_parameter009.txt
background_logWeights.txt             background_marginalDistribution006.txt background_parameter003.txt  background_parameterSummary.txt
background_marginalDistribution000.txt background_marginalDistribution007.txt background_parameter004.txt background_posteriorDistribution.txt
background_marginalDistribution001.txt background_marginalDistribution008.txt background_parameter005.txt
background_marginalDistribution002.txt background_marginalDistribution009.txt background_parameter006.txt
```

The likelihood points (values) rejected at each iteration of the NSMC. The points are clearly sorted in increasing order, according to the algorithm working principle.

```
# Posterior sample from nested sampling
# log(Likelihood)
-2.157152544e+05
-2.156454080e+05
-2.153993501e+05
-2.148658921e+05
-2.147671588e+05
```

Results

- Open a new tab in your terminal and cd **~/Background/results/KIC012008916/00/**

```
background_configuringParameters.txt  background_marginalDistribution003.txt background_parameter000.txt  background_parameter007.txt
background_evidenceInformation.txt    background_marginalDistribution004.txt background_parameter001.txt  background_parameter008.txt
background_logLikelihood.txt          background_marginalDistribution005.txt background_parameter002.txt  background_parameter009.txt
background_logWeights.txt             background_marginalDistribution006.txt background_parameter003.txt  background_parameterSummary.txt
background_marginalDistribution000.txt background_marginalDistribution007.txt background_parameter004.txt background_posteriorDistribution.txt
background_marginalDistribution001.txt background_marginalDistribution008.txt background_parameter005.txt
background_marginalDistribution002.txt background_marginalDistribution009.txt background_parameter006.txt
```

```
9.149732536e+00
8.229143469e+00
8.120328126e+00
1.068308871e+01
9.535135315e+00
1.059146273e+01
1.084265904e+01
9.277594058e+00
```

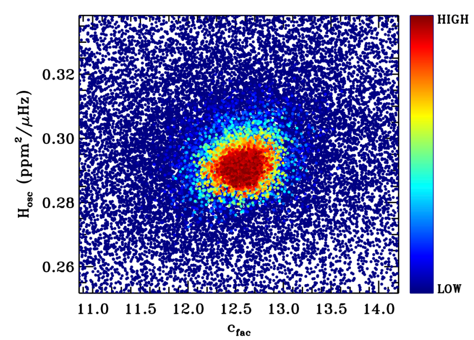
The parameter values for each free parameter, in the same order as the marginal distributions and the priors. The values within each file are sorted according to the likelihood distribution.

4.- FITTING THE BACKGROUND SIGNAL

Results

- Open a new tab in your terminal and cd **~/Background/results/KIC012008916/00/**

```
background_configuringParameters.txt    background_marginalDistribution003.txt background_parameter000.txt    background_parameter007.txt
background_evidenceInformation.txt      background_marginalDistribution004.txt background_parameter001.txt    background_parameter008.txt
background_logLikelihood.txt            background_marginalDistribution005.txt background_parameter002.txt    background_parameter009.txt
background_logWeights.txt               background_marginalDistribution006.txt background_parameter003.txt    background_parameterSummary.txt
background_marginalDistribution000.txt  background_marginalDistribution007.txt background_parameter004.txt    background_posteriorDistribution.txt
background_marginalDistribution001.txt  background_marginalDistribution008.txt background_parameter005.txt
background_marginalDistribution002.txt  background_marginalDistribution009.txt background_parameter006.txt
```



Combining the likelihood sampling and pairs of parameter sampling, allows you to build up a correlation map and therefore extract information of the correlations between parameters.

Results

- Open a new tab in your terminal and
  cd **~/Background/results/KIC012008916/00/**

```
background_configuringParameters.txt  background_marginalDistribution003.txt background_parameter000.txt   background_parameter007.txt
background_evidenceInformation.txt    background_marginalDistribution004.txt background_parameter001.txt   background_parameter008.txt
background_logLikelihood.txt          background_marginalDistribution005.txt background_parameter002.txt   background_parameter009.txt
background_logWeights.txt             background_marginalDistribution006.txt background_parameter003.txt   background_parameterSummary.txt
background_marginalDistribution000.txt background_marginalDistribution007.txt background_parameter004.txt  background_posteriorDistribution.txt
background_marginalDistribution001.txt background_marginalDistribution008.txt background_parameter005.txt
background_marginalDistribution002.txt background_marginalDistribution009.txt background_parameter006.txt
```

```
# Posterior probability distribution from nested sampling
0.000000000e+00
0.000000000e+00
0.000000000e+00
0.000000000e+00
0.000000000e+00
0.000000000e+00
0.000000000e+00
0.000000000e+00
0.000000000e+00
0.000000000e+00
0.000000000e+00
```

k-dimensional posterior probability distribution. You can plot this information against pairs of free parameter values to visualize probability maps in 2D.

Results

- Open a new tab in your terminal and
  cd **~/Background/results/KIC012008916/00/**

```
background_configuringParameters.txt   background_marginalDistribution003.txt background_parameter000.txt   background_parameter007.txt
background_evidenceInformation.txt     background_marginalDistribution004.txt background_parameter001.txt   background_parameter008.txt
background_logLikelihood.txt           background_marginalDistribution005.txt background_parameter002.txt   background_parameter009.txt
background_logWeights.txt              background_marginalDistribution006.txt background_parameter003.txt   background_parameterSummary.txt
background_marginalDistribution000.txt background_marginalDistribution007.txt background_parameter004.txt   background_posteriorDistribution.txt
background_marginalDistribution001.txt background_marginalDistribution008.txt background_parameter005.txt
background_marginalDistribution002.txt background_marginalDistribution009.txt background_parameter006.txt
```

This files contains all the estimators (mode, median, mean, credible intervals, variance, skewness) for each free parameter. The list is ordered top-bottom according to prior definition (it is the same numbering as the MPDs)

```
# Summary of Parameter Estimation from nested sampling
# Credible intervals are the shortest credible intervals
# according to the usual definition
# Credible level: 68.30 %
# Column #1: I Moment (Mean)
# Column #2: Median
# Column #3: Mode
# Column #4: II Moment (Variance if Normal distribution)
# Column #5: Lower Credible Limit
# Column #6: Upper Credible Limit
# Column #7: Skewness (Asymmetry of the distribution, -1 to the left, +1 to the right, 0 if symmetric)
1.580443831e+01  1.581400202e+01  1.594134468e+01  4.513582182e-01  1.515706805e+01  1.646750495e+01  -1.367507252e-02
9.127250886e+01  9.116781429e+01  9.119265731e+01  1.347911336e+01  8.766630643e+01  9.482751130e+01  2.046147954e-01
```


The parameter summary is the most important file for Bayesian parameter estimation. In a Bayesian mindset, the estimator that should be preferred is the median value.

Completing process

- Make sure your run is finished without errors (see the Manual for more details)

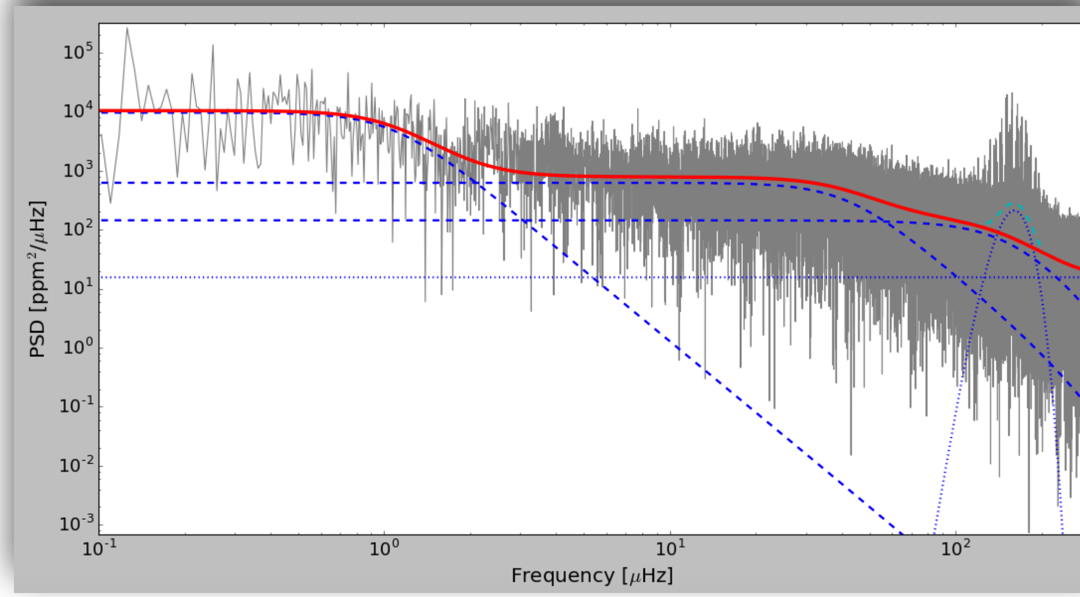
```
Nit: 50  Ncl: 1  Nlive: 2000  CPM: 0.0251776  Ratio: 6.45e+102  log(E): -2.375  IG: 3.68
```

```
------------------------------------------------
 Final log(E): 3.65068e+06 +/- 0.270612
------------------------------------------------
 Total Computational Time: 1.16 hours
------------------------------------------------
```

- Move the file **plot_background.py** to **~/Background/**

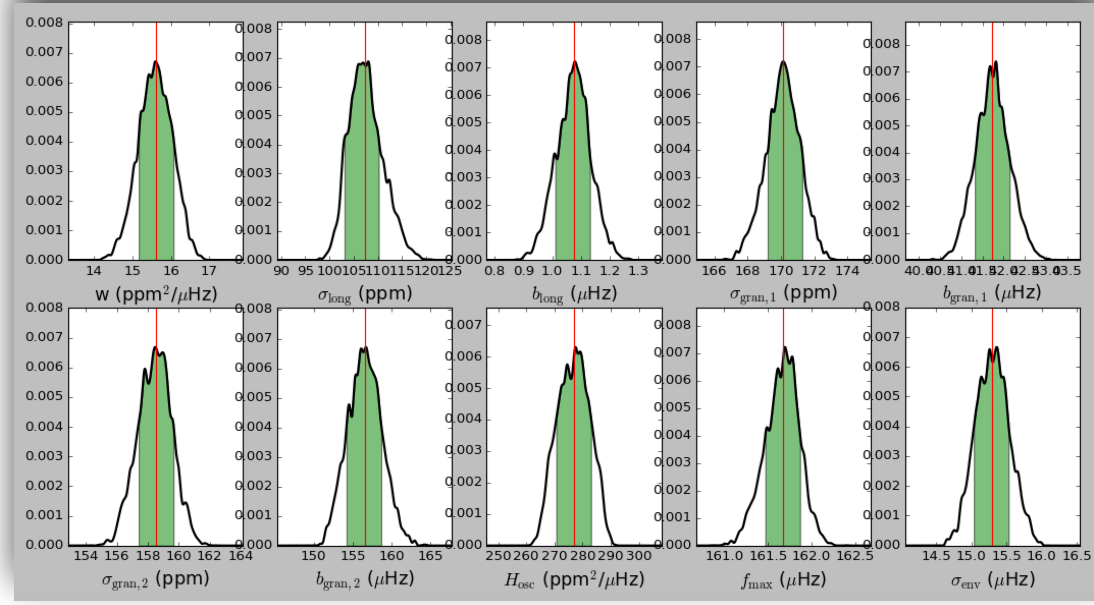
- When your computation is completed:
  cd **~/Background/**
  type on your terminal
  python plot_background.py

Plotting (1)



The background fit that matters for us is represented by the solid red line. The individual components of the background model are marked by blue lines, while the Gaussian hump on top of the background fit is indicated by a dashed cyan line.

Plotting (2)



The marginal probability distributions (MPDs) of all the background fitting parameters are displayed in a 2x5 format window and ordered from top left to bottom right according to their order in the prior parameter file and in the model of the background implemented in DIAMONDS. The colored bands mark the region of the credible levels, corresponding to the 68.3% of the total probability.

4.- FITTING THE BACKGROUND SIGNAL

Questions

- What is the value of $\nu_{\rm max}$ and what estimator (mode, etc.)?
- Can you say what is the evolutionary stage of the star from $\nu_{\rm max}$ (Main Sequence, Subgiant, Red Giant)?
- What is the Radius of the star?

$$\frac{R}{R_\odot} \simeq ?$$

$$\nu_{\rm max,\odot} = 3150\,\mu{\rm Hz} \qquad \Delta\nu_\odot = 134.9\,\mu{\rm Hz}$$

$$T_{\rm eff,\odot} = 5777\,{\rm K} \qquad T_{\rm eff} = 5100\,{\rm K} \qquad \Delta\nu = 12.9\,\mu{\rm Hz}$$

Measure the value of nu_max and calculate the resulting Radius of the star.
R/R_s = (nu_max/nu_max,s) (Dnu/Dnu_s)^(-2) (Teff/Teff_s)^0.5
Use Dnu = 12.9 microHz, and Teff = 5100 K, nu_max,s = 3150 microHz, Dnu_s = 134.9 microHz, Teff_s = 5777 K.
Exp. R ~ 5.2 R_s

1. - BAYESIAN STATISTICS

2. - NESTED SAMPLING MONTE CARLO

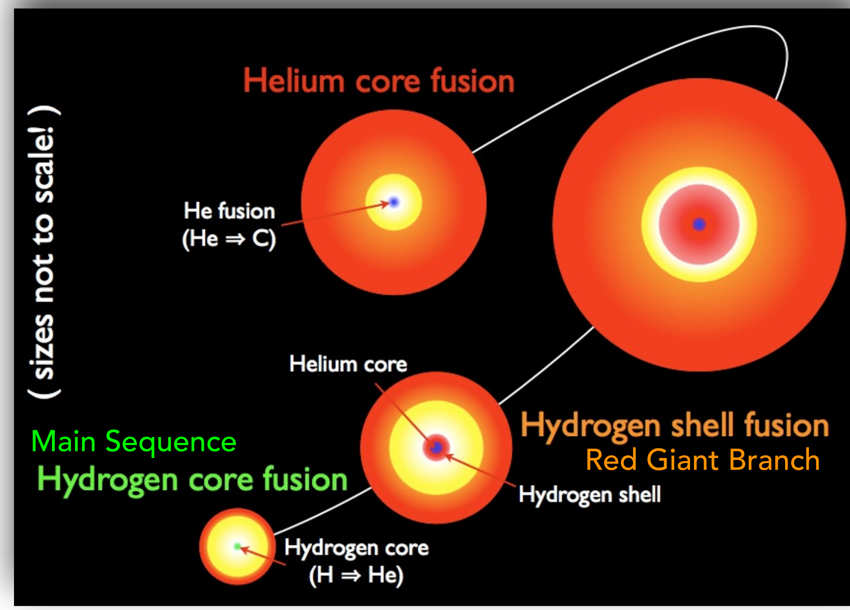
3. - THE DIAMONDS CODE

4. - FITTING THE BACKGROUND SIGNAL

5. - FITTING THE OSCILLATION MODES

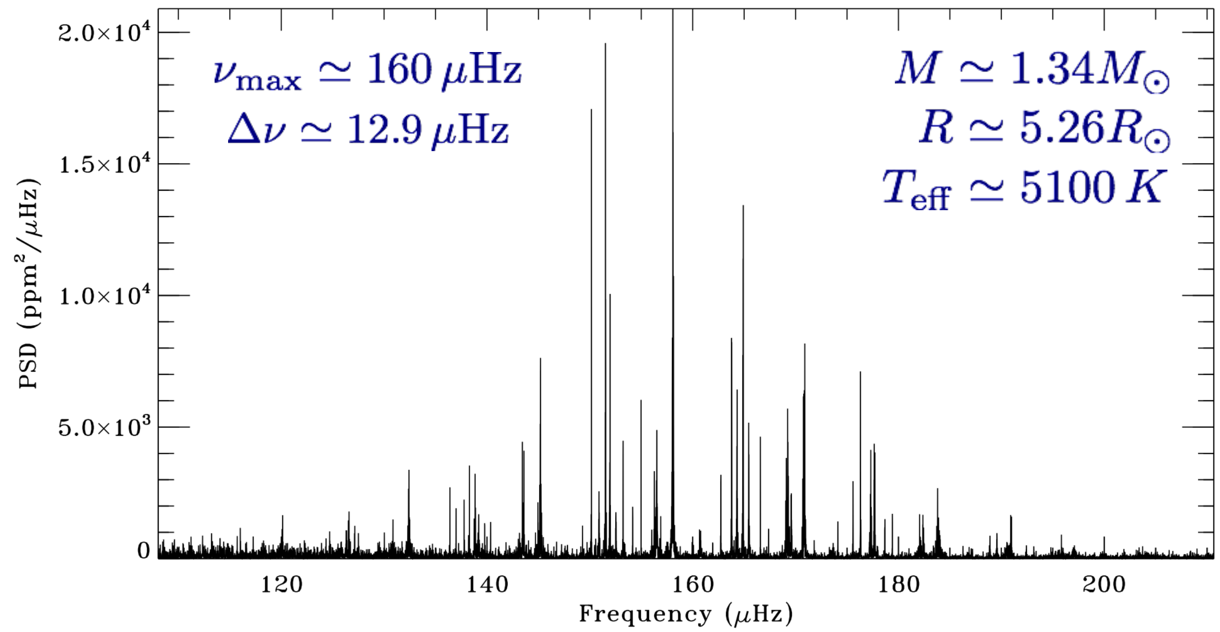
6. - PEAK SIGNIFICANCE TEST

Evolved cool stars



by T. Kallinger

The power spectrum of a low-luminosity RGB



KIC 12008916

Corsaro, De Ridder, García 2015 A&A, 579, 83

To-do list

- Open your terminal and cd **~/PeakBagging/build/**
- Make sure you have edited the script file **multirun** using your local path and the file **localPath.txt** as well
- Run the script via terminal for the star KIC 12008916 and one chunk of its PSD as
  ./multirun 012008916 0 1 run_4

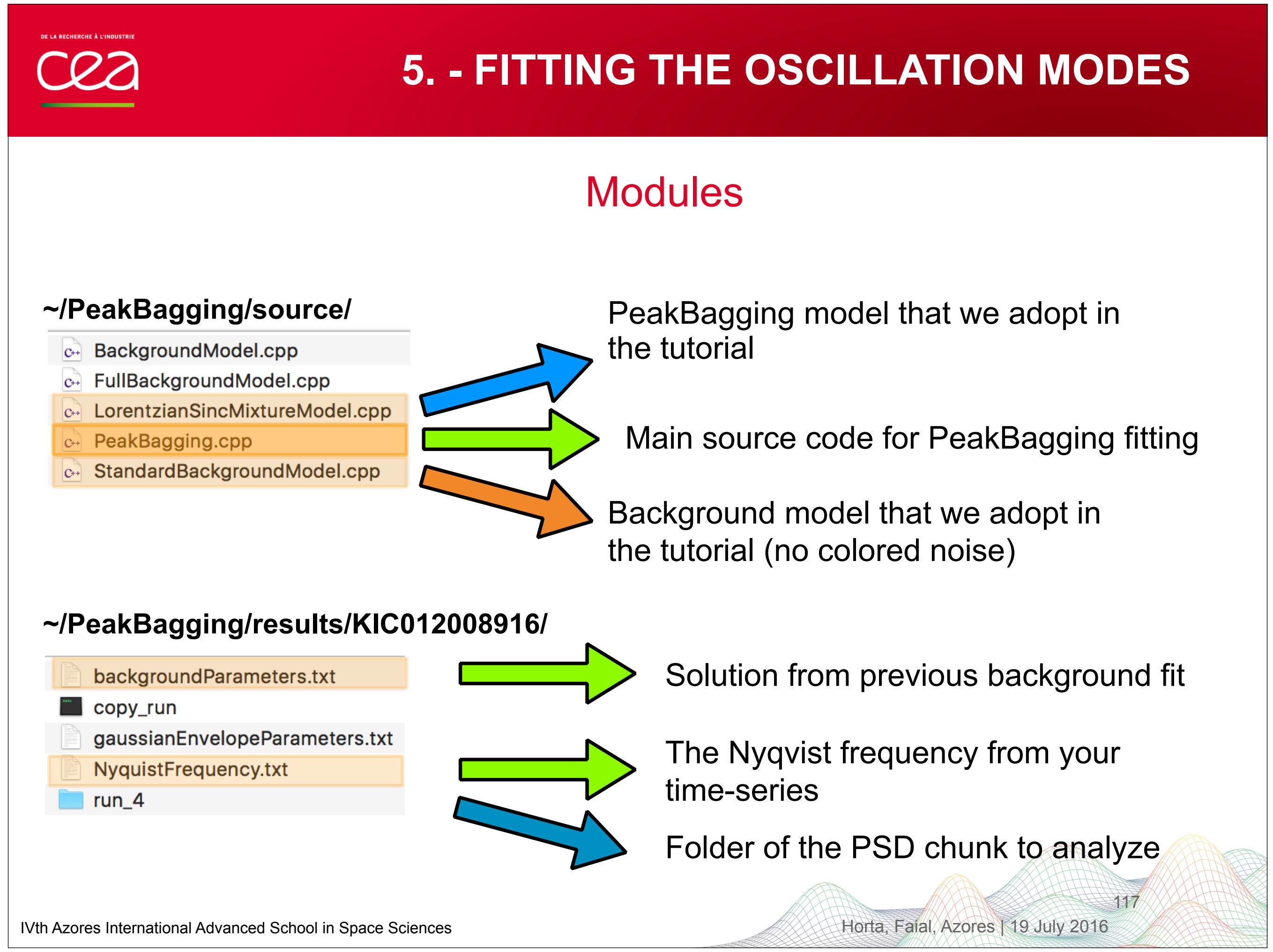
KIC ID with 9 digits

Label of the starting run
PeakBagging/results/
KIC012008916/run_4/00/

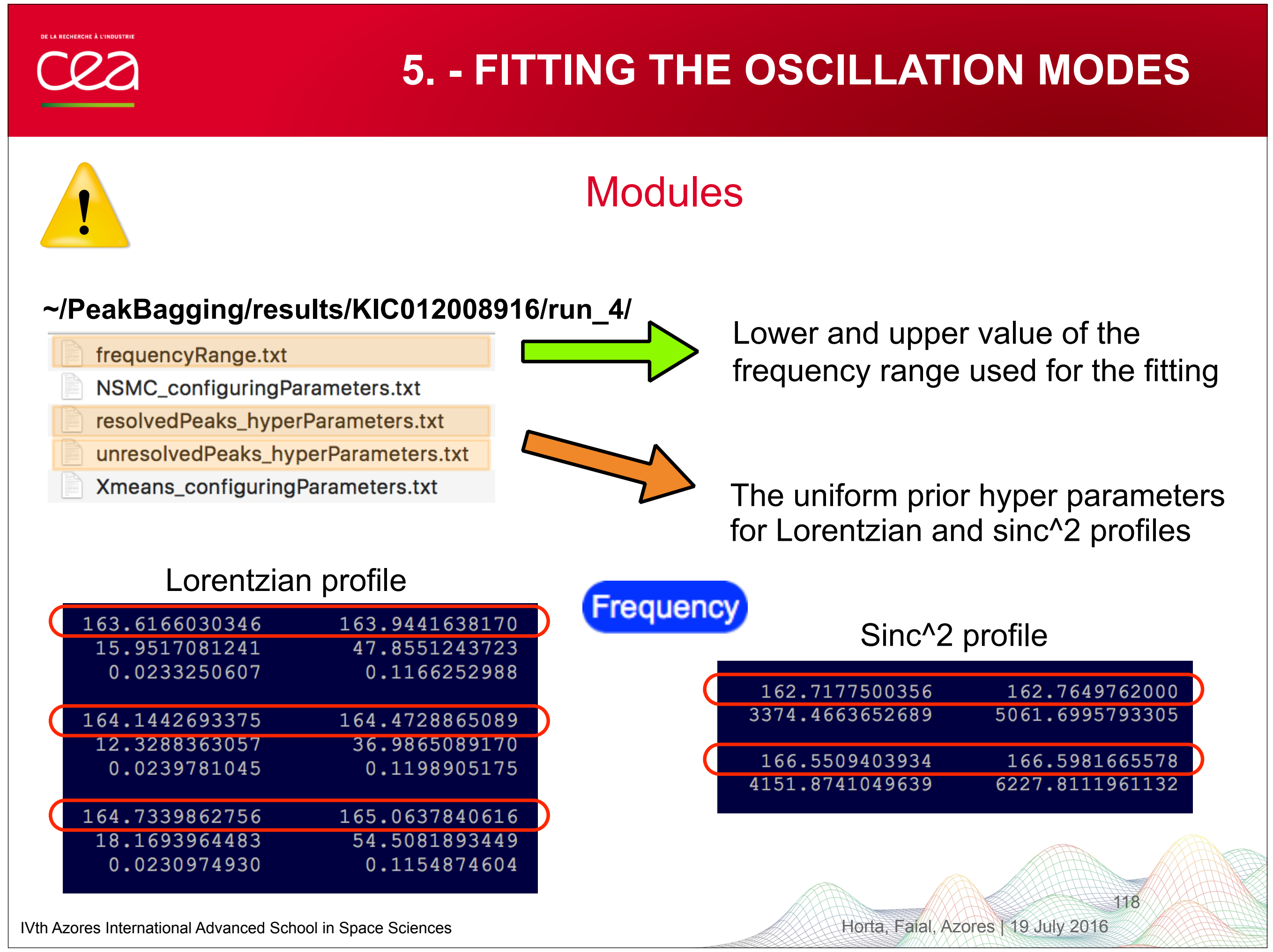
Number of independent runs.

A label (string) with the name of your chunk (can be any name you like)

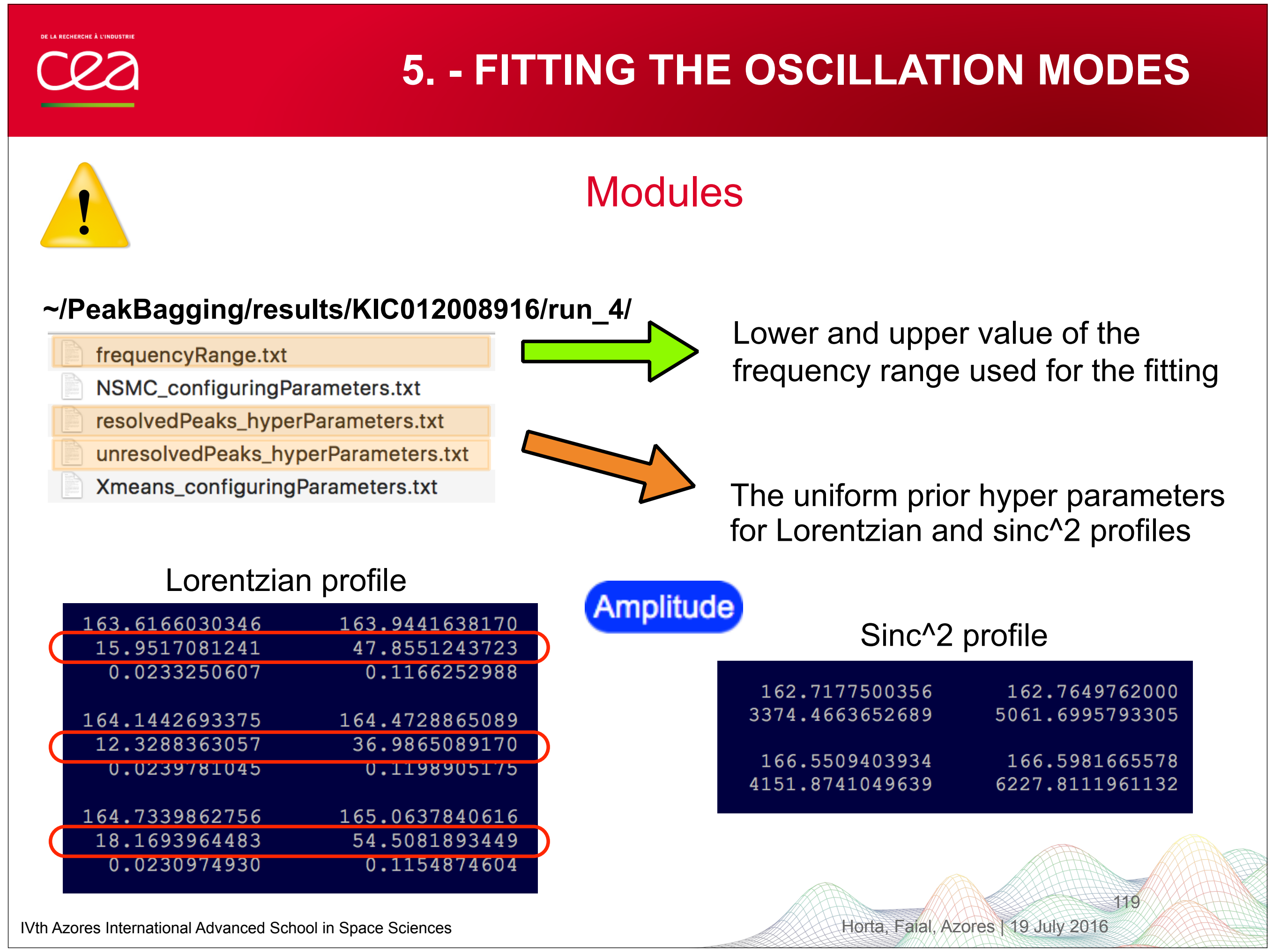
An automated method to measure DP1 (the asymptotic period spacing of gravity modes) has been developed by Vrard M. et al. 2016. Other methods rely on a bi-dimensional grid-search approach (see Buysschaert et al. 2016; Corsaro et al. 2015b).
If you have a ‘newline’ error when executing the script, then you should create a subfolder labeled 00 under the directory ~/PeakBagging/results/KIC012008916/run_4/.
Then, go back to ~/PeakBagging/build/ and execute the command via terminal
./peakbagging 012008916 run_4 00



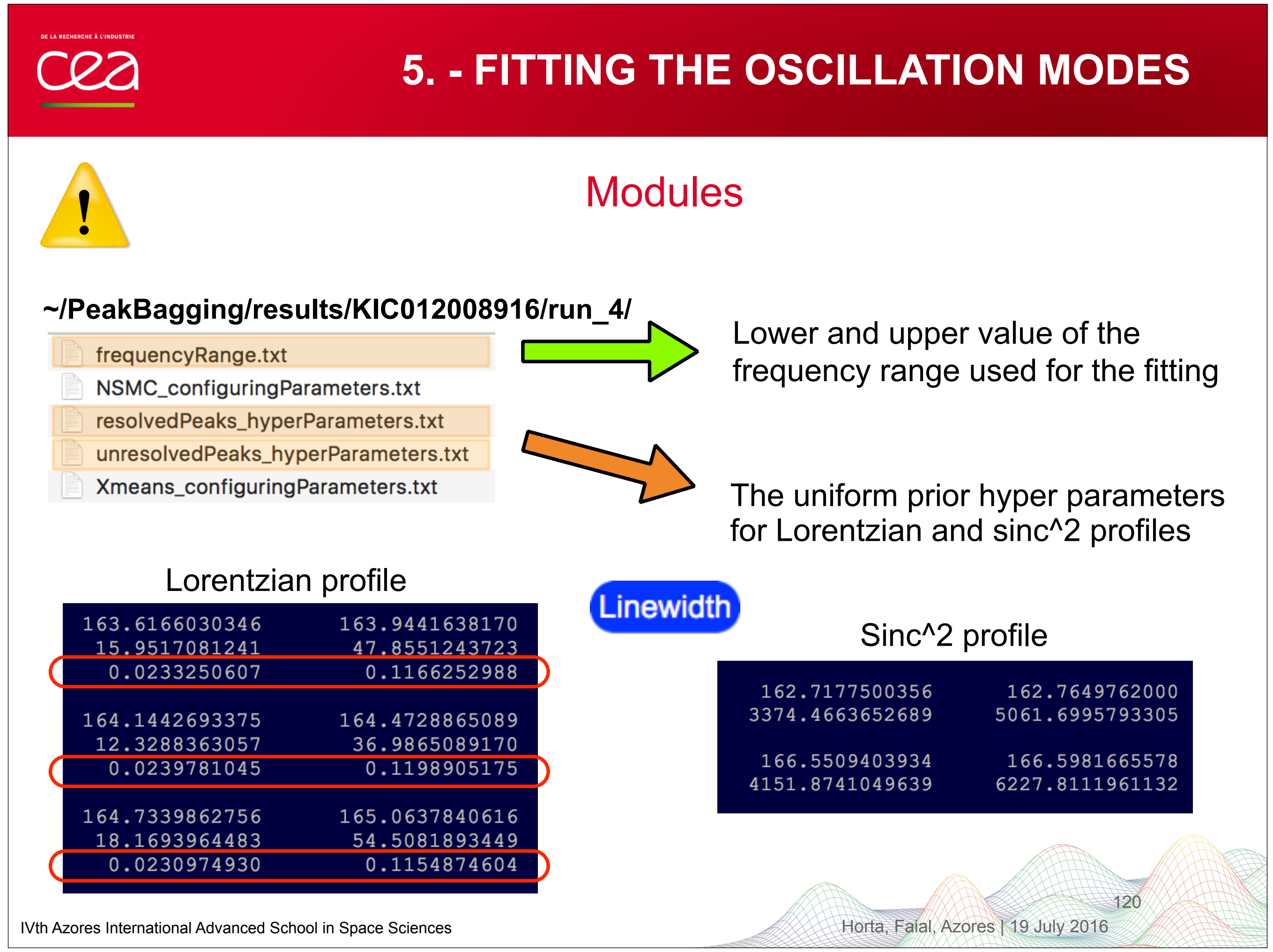
Conversely to the Background code, in this case we have an additional file, backgroundParameters.txt, to contain the solution for the background fitting from the previous analysis, and a folder (here labeled run_4) that contains the configuring parameters of the PSD chunk that we want to fit.



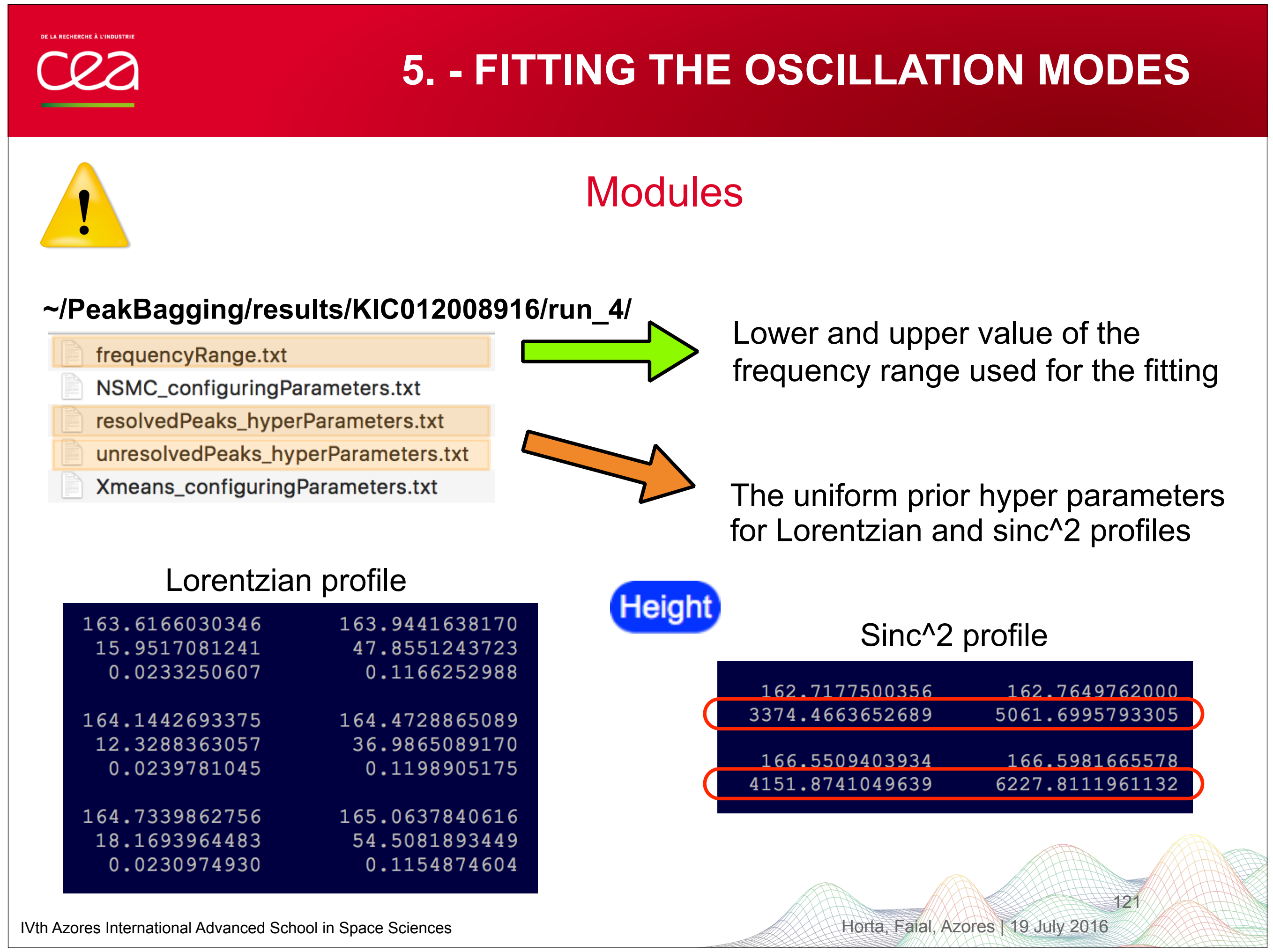
The order of the frequency centroids does not matter in the setting of the priors. However, one has to keep in mind that in the resulting parameter summary, the free parameters will be listed in the same order as the priors are defined, with the block of Lorentzian profiles before that of the sinc^2 profiles.



The order of the frequency centroids does not matter in the setting of the priors. However, one has to keep in mind that in the resulting parameter summary, the free parameters will be listed in the same order as the priors are defined, with the block of Lorentzian profiles before that of the sinc^2 profiles.

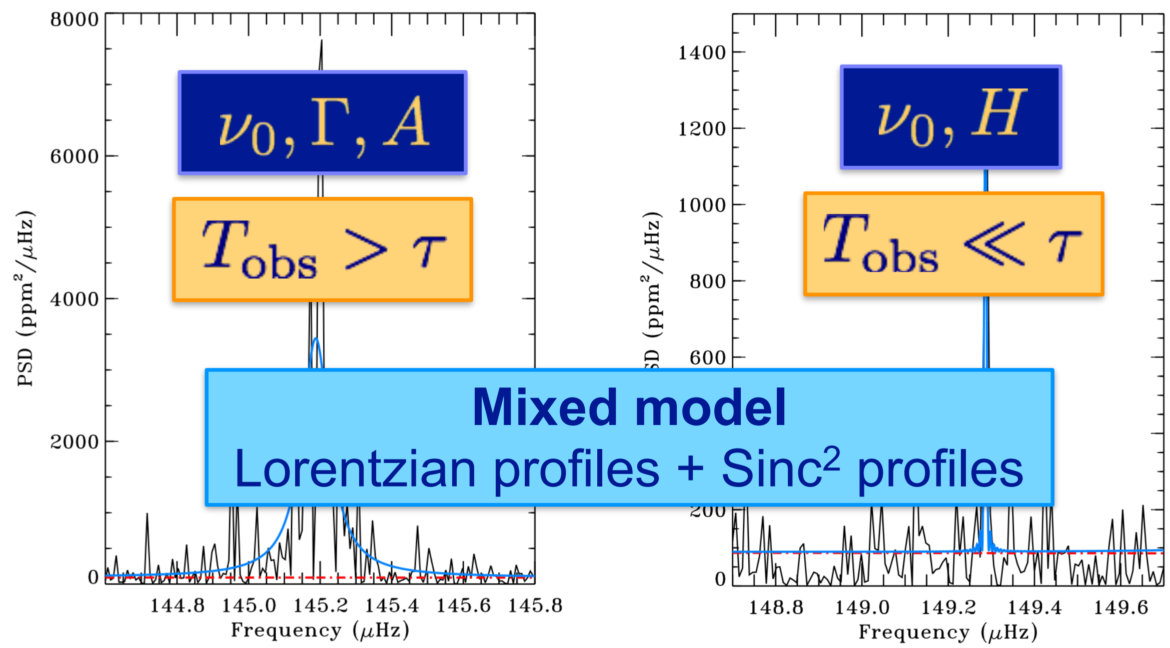


The order of the frequency centroids does not matter in the setting of the priors. However, one has to keep in mind that in the resulting parameter summary, the free parameters will be listed in the same order as the priors are defined, with the block of Lorentzian profiles before that of the sinc^2 profiles.



The order of the frequency centroids does not matter in the setting of the priors. However, one has to keep in mind that in the resulting parameter summary, the free parameters will be listed in the same order as the priors are defined, with the block of Lorentzian profiles before that of the sinc^2 profiles.
It is important to note that these prior parameters will be treated as a unique block during the computation, with the block of the Lorentzian profiles before that of the sinc^2 profiles. We will come back to that later in the presentation.

The Lorentzian and sinc^2 profiles



The mixture model of Lorentzian and sinc^2 profiles arises from the presence of mixed dipole modes in the red giant stars. Mixed dipole modes appear to be often unresolved, meaning that they do not exhibit a line width that can be measured with reliability. In this case, it is pointless to fit an oscillation peak with a Lorentzian profile, because the line width parameter is not physically meaningful. A sinc^2 profile requires only two free parameters, its priors are easy to set up and it improves the overall efficiency of the fitting process. For less evolved stars (essential main sequence stars with solar-like oscillations) there is not need to include sinc^2 profiles because the oscillation peaks do show some degree of resolution, hence they exhibit a width that can be measured.

The Lorentzian and sinc² profiles

$$P_{\mathrm{pkb}}(\nu) = \left[P_{\mathrm{osc}}(\nu) + \overline{B}(\nu)\right] R(\nu) + \overline{W}$$

$$P_{\mathrm{osc}}(\nu) = \sum_i \mathcal{P}_{\mathrm{res,i}}(\nu) + \sum_j \mathcal{P}_{\mathrm{unres,j}}(\nu)$$

Sinc² Profile

$$\mathcal{P}_{\mathrm{unres,0}}(\nu) = H_0 \,\mathrm{sinc}^2\left[\frac{\pi(\nu-\nu_0)}{\delta\nu_{\mathrm{res}}}\right]$$

Lorentzian Profile

$$\mathcal{P}_{\mathrm{res,0}}(\nu) = \frac{A_0^2/(\pi\Gamma_0)}{1+4\left(\frac{\nu-\nu_0}{\Gamma_0}\right)^2}$$

The amplitude in the Lorentzian profile has to be rescaled by a factor sqrt(2) if the power spectrum is computed as a single-sided one. This is because normally the computed power spectra have to satisfy the Plancherel theorem of energy conservation from time domain to frequency domain.

5. - FITTING THE OSCILLATION MODES

The peak bagging model in DIAMONDS

```
// Add a Lorentzian profile for each resolved mode (both p modes and resolved mixed modes)

for (int mode = 0; mode < Nresolved; ++mode)
{
    // Initialize parameters of current mode with proper access to elements of total array of free parameters

    double centralFrequency = modelParameters(3*mode);
    double amplitude = modelParameters(3*mode + 1);
    double linewidth = modelParameters(3*mode + 2);

    Functions::modeProfileWithAmplitude(singleModePrediction, covariates, centralFrequency, amplitude, linewidth);
    predictions += singleModePrediction;
}

// Add sinc-square profile for each unresolved mixed mode

for (int mode = 0; mode < Nunresolved; ++mode)
{
    double centralFrequency = modelParameters(3*Nresolved + 2*mode);
    double height = modelParameters(3*Nresolved + 2*mode + 1);            // mixedModesHeights(mode);

    Functions::modeProfileSinc(singleModePrediction, covariates, centralFrequency, height, frequencyResolution);
    predictions += singleModePrediction;
}

// Adjust by apodization of the signal

predictions *= responseFunction;

// Add background component

predictions += backgroundPrediction;
```


The implementation of the peak bagging model can be found in the file LorentzianSincMixtureModel.cpp, under the directory ~/PeakBagging/source/. The predict function contains a first bit with a loop over all the Lorentzian profiles, with frequency centroid (microHz), amplitude (ppm), and line width (microHz) for each profile, as given with the input prior information.

The peak bagging model in DIAMONDS

```
// Add a Lorentzian profile for each resolved mode (both p modes and resolved mixed modes)

for (int mode = 0; mode < Nresolved; ++mode)
{
    // Initialize parameters of current mode with proper access to elements of total array of free parameters

    double centralFrequency = modelParameters(3*mode);
    double amplitude = modelParameters(3*mode + 1);
    double linewidth = modelParameters(3*mode + 2);

    Functions::modeProfileWithAmplitude(singleModePrediction, covariates, centralFrequency, amplitude, linewidth);
    predictions += singleModePrediction;
}

// Add sinc-square profile for each unresolved mixed mode

for (int mode = 0; mode < Nunresolved; ++mode)
{
    double centralFrequency = modelParameters(3*Nresolved + 2*mode);
    double height = modelParameters(3*Nresolved + 2*mode + 1);            // mixedModesHeights(mode);

    Functions::modeProfileSinc(singleModePrediction, covariates, centralFrequency, height, frequencyResolution);
    predictions += singleModePrediction;
}

// Adjust by apodization of the signal

predictions *= responseFunction;


// Add background component

predictions += backgroundPrediction;
```

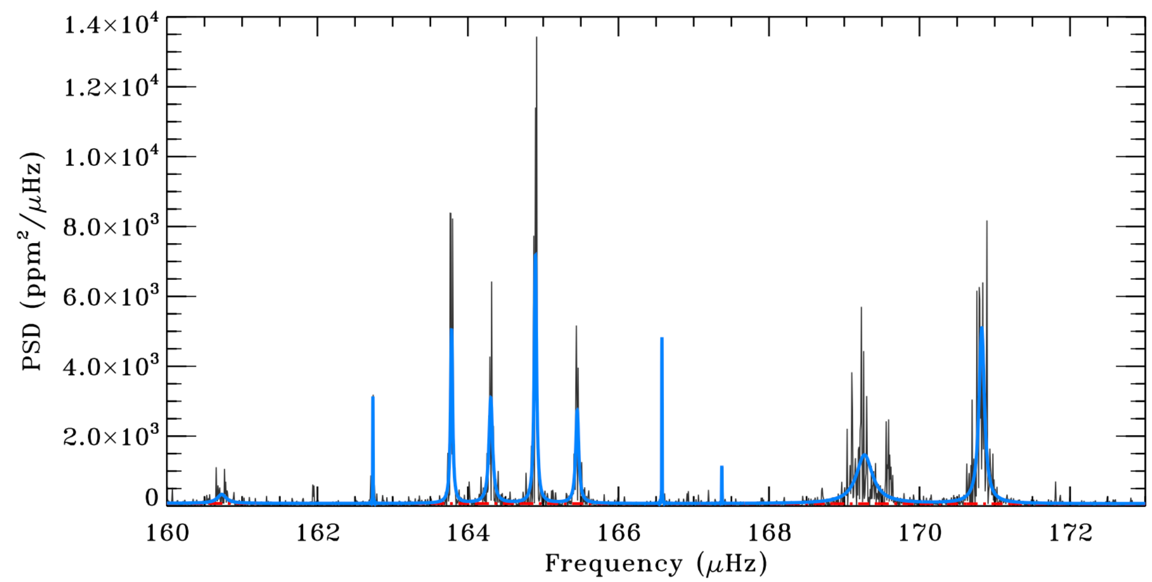

The second bit is a loop over all the sinc^2 profiles, with free parameters of frequency centroid (microHz) and height (ppm^2/microHz).

The peak bagging model in DIAMONDS

```
// Add a Lorentzian profile for each resolved mode (both p modes and resolved mixed modes)

for (int mode = 0; mode < Nresolved; ++mode)
{
    // Initialize parameters of current mode with proper access to elements of total array of free parameters

    double centralFrequency = modelParameters(3*mode);
    double amplitude = modelParameters(3*mode + 1);
    double linewidth = modelParameters(3*mode + 2);

    Functions::modeProfileWithAmplitude(singleModePrediction, covariates, centralFrequency, amplitude, linewidth);
    predictions += singleModePrediction;
}


// Add sinc-square profile for each unresolved mixed mode

for (int mode = 0; mode < Nunresolved; ++mode)
{
    double centralFrequency = modelParameters(3*Nresolved + 2*mode);
    double height = modelParameters(3*Nresolved + 2*mode + 1);            // mixedModesHeights(mode);

    Functions::modeProfileSinc(singleModePrediction, covariates, centralFrequency, height, frequencyResolution);
    predictions += singleModePrediction;
}

// Adjust by apodization of the signal

predictions *= responseFunction;


// Add background component

predictions += backgroundPrediction;
```


The last bit is the product with the apodization signal, and finally the sum of the white noise component and of the other background components (which are in turn modulated by the same apodization signal), and that are kept as a constants in this case.

Fitting an individual radial order



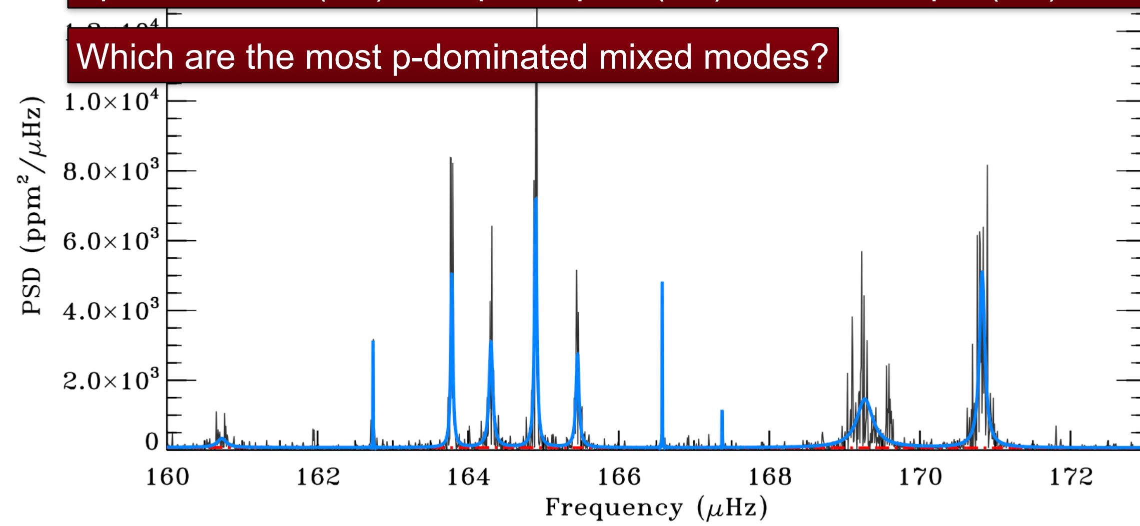
KIC 12008916

This is the chunk of PSD that you are going to fit with a peak bagging model. In blue is the fit that is obtained with DIAMONDS. The frequency range of this chunk corresponds to the value of the large frequency separation Dnu. The radial mode is the high peak on the right side of the range.

Questions

Spot the radial (l=0), the quadrupole (l=2) and the octuple (l=3) mode

Which are the most p-dominated mixed modes?



KIC 12008916

Completing process

- Make sure your run is finished without errors (see the Manual for more details)

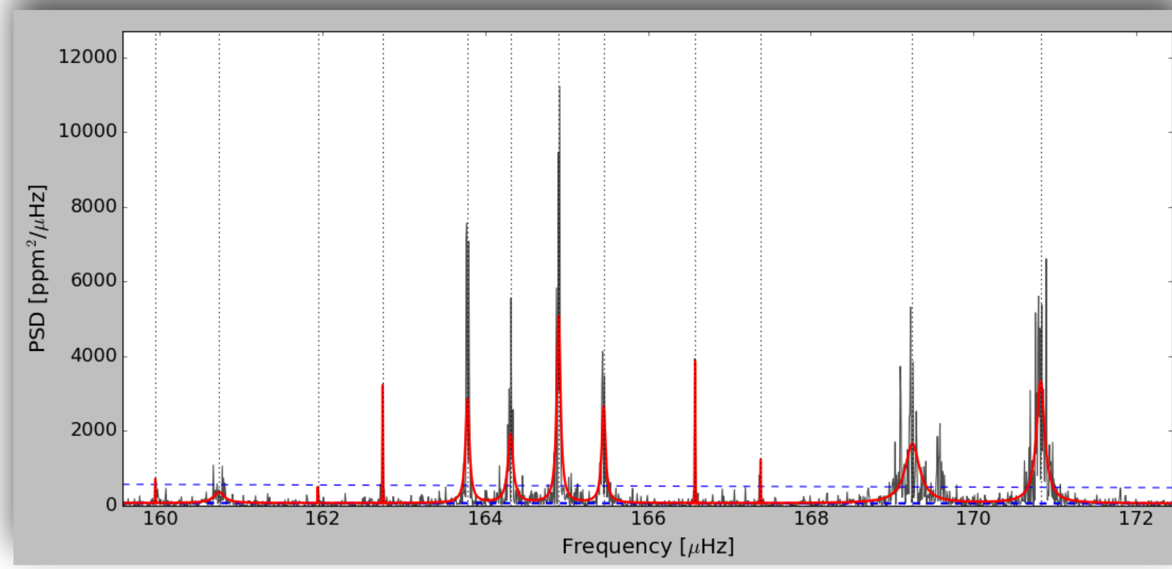
```
Nit: 50  Ncl: 1  Nlive: 2000  CPM: 0.0251776  Ratio: 6.45e+102  log(E): -2.375  IG: 3.68
```

```
------------------------------------------------
 Final log(E): 3.65068e+06 +/- 0.270612
------------------------------------------------
 Total Computational Time: 1.16 hours
------------------------------------------------
```

- Move the file **plot_peakbagging.py** to **~/PeakBagging/**

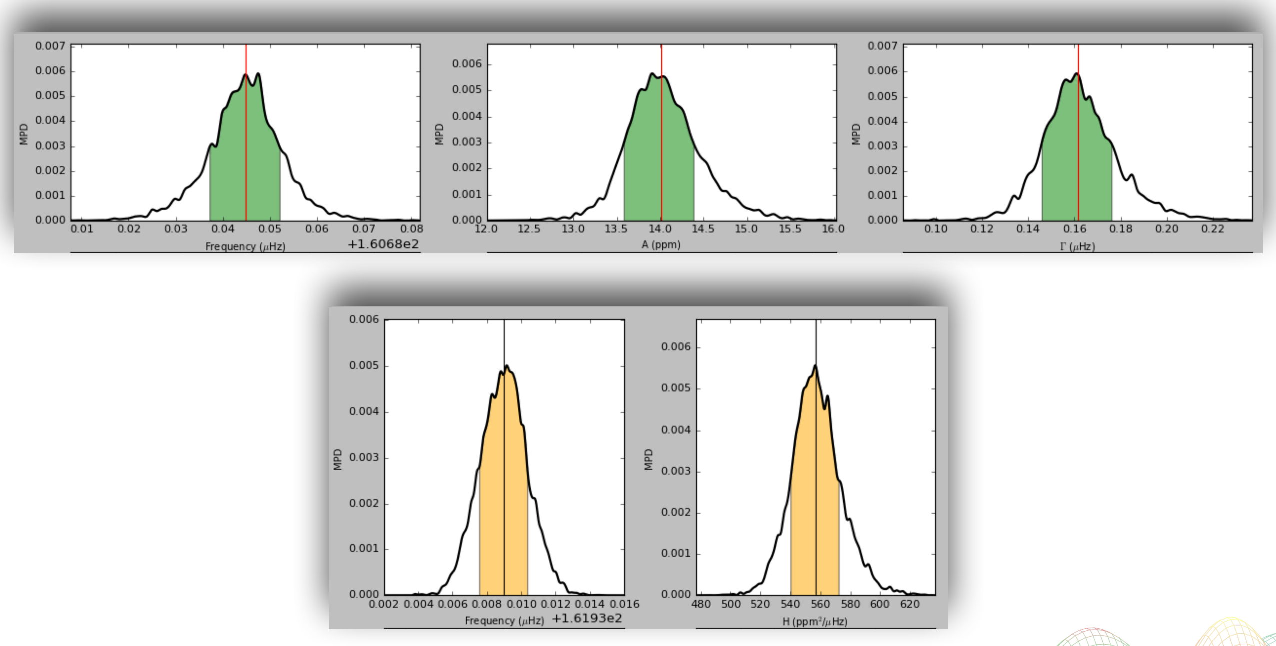
- When your computation is completed:
  cd **~/PeakBagging/**
  type on your terminal
  python plot_peakbagging.py

Plotting (1)



If for example we want to access the frequency of the last peak on the right side, l=0, we need to count how many Lorentzian profiles were fit in our chunk (in this case 7), therefore evaluate the number of the line in the parameterSummary.txt file that corresponds to the frequency of the last Lorentzian profile, namely 7x3 = 21. So the frequency of the radial mode can be found in line 21 of the output file containing all the estimators. If instead we want to pick up the frequency output of the last sinc^2 profile, the one with the highest frequency, then we have to consider that we have 5 sinc^2 profiles, hence the corresponding frequency will be placed in line 7x3 + 5x2 = 31 of the output file containing the estimators of each free parameter.

Plotting (2)



The plotting of the marginal probability distributions (MPDs) shows the Lorentzian profiles (in green) in a format 3x3, where each row corresponds to a single oscillation peak, with parameters Frequency centroid (microHz), Amplitude (ppm), Linewidth (microHz) from left to right. The sinc^2 profile oscillation peaks are instead displayed in yellow in a 2x4 format, where in each row there are two oscillation peaks, with corresponding parameters Frequency centroid (microHz) and Height (ppm^2/microHz) from left to right.

5. - FITTING THE OSCILLATION MODES

Results

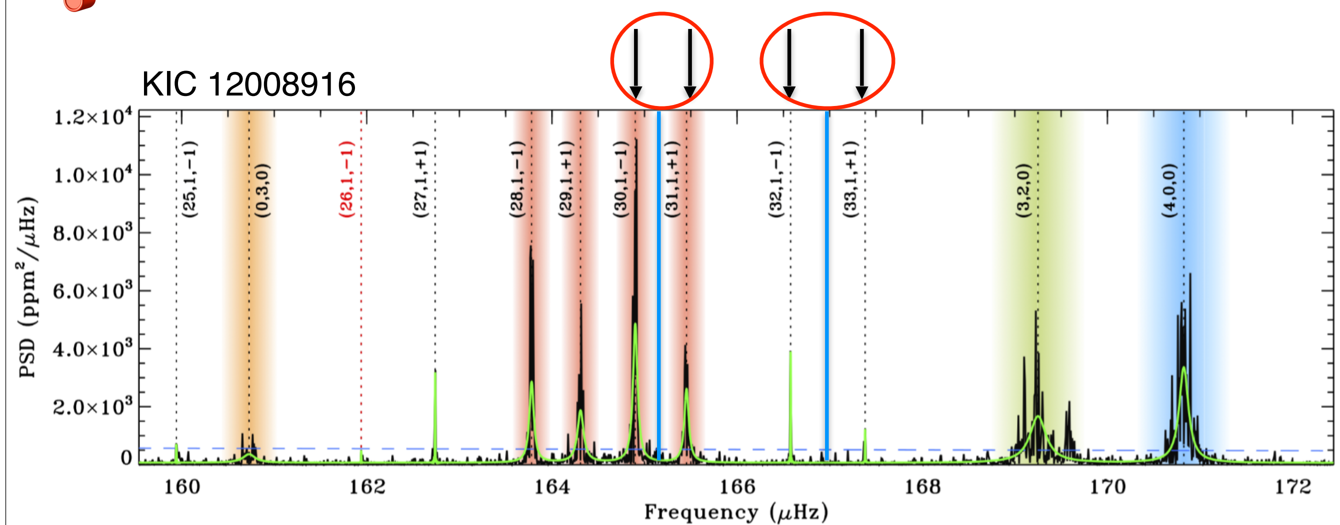
- Open a new tab in your terminal and cd **~/PeakBagging/results/KIC012008916/run_4/00/**

```
peakbagging_configuringParameters.txt   peakbagging_marginalDistribution020.txt peakbagging_parameter012.txt
peakbagging_evidenceInformation.txt     peakbagging_marginalDistribution021.txt peakbagging_parameter013.txt
peakbagging_logLikelihood.txt           peakbagging_marginalDistribution022.txt peakbagging_parameter014.txt
peakbagging_marginalDistribution000.txt peakbagging_marginalDistribution023.txt peakbagging_parameter015.txt
peakbagging_marginalDistribution001.txt peakbagging_marginalDistribution024.txt peakbagging_parameter016.txt
peakbagging_marginalDistribution002.txt peakbagging_marginalDistribution025.txt peakbagging_parameter017.txt
peakbagging_marginalDistribution003.txt peakbagging_marginalDistribution026.txt peakbagging_parameter018.txt
peakbagging_marginalDistribution004.txt peakbagging_marginalDistribution027.txt peakbagging_parameter019.txt
peakbagging_marginalDistribution005.txt peakbagging_marginalDistribution028.txt peakbagging_parameter020.txt
peakbagging_marginalDistribution006.txt peakbagging_marginalDistribution029.txt peakbagging_parameter021.txt
peakbagging_marginalDistribution007.txt peakbagging_marginalDistribution030.txt peakbagging_parameter022.txt
peakbagging_marginalDistribution008.txt peakbagging_parameter000.txt            peakbagging_parameter023.txt
peakbagging_marginalDistribution009.txt peakbagging_parameter001.txt            peakbagging_parameter024.txt
peakbagging_marginalDistribution010.txt peakbagging_parameter002.txt            peakbagging_parameter025.txt
peakbagging_marginalDistribution011.txt peakbagging_parameter003.txt            peakbagging_parameter026.txt
peakbagging_marginalDistribution012.txt peakbagging_parameter004.txt            peakbagging_parameter027.txt
peakbagging_marginalDistribution013.txt peakbagging_parameter005.txt            peakbagging_parameter028.txt
peakbagging_marginalDistribution014.txt peakbagging_parameter006.txt            peakbagging_parameter029.txt
peakbagging_marginalDistribution015.txt peakbagging_parameter007.txt            peakbagging_parameter030.txt
peakbagging_marginalDistribution016.txt peakbagging_parameter008.txt            peakbagging_parameterSummary.txt
peakbagging_marginalDistribution017.txt peakbagging_parameter009.txt            peakbagging_posteriorDistribution.txt
peakbagging_marginalDistribution018.txt peakbagging_parameter010.txt
peakbagging_marginalDistribution019.txt peakbagging_parameter011.txt
```


The output files generated from the peak bagging fitting process. The description of the output files is the same as that of the background fitting.

5. - FITTING THE OSCILLATION MODES

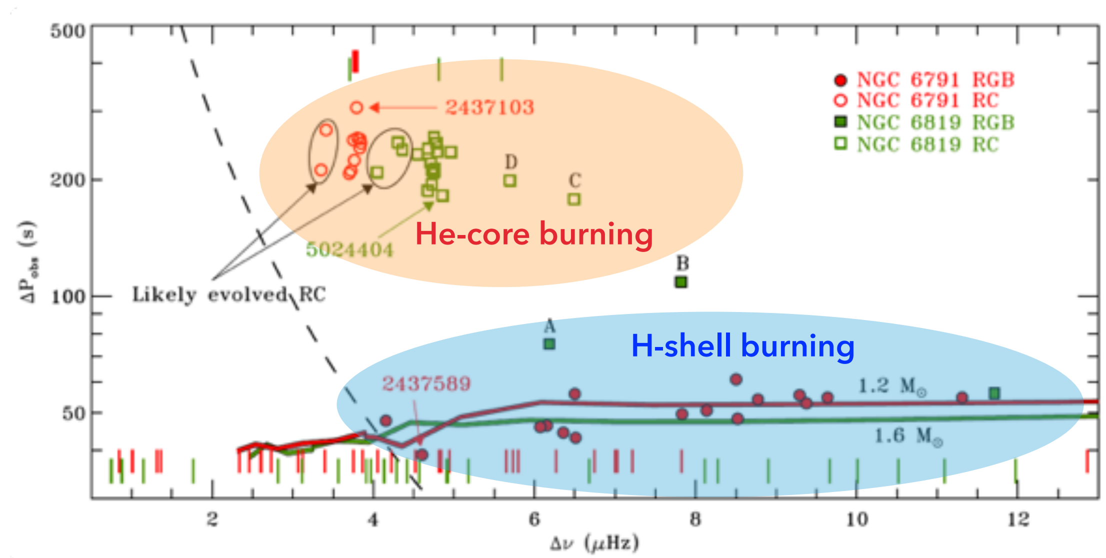
Questions



$$\nu_{\ell=1,m=0} = 165.178\,\mu\text{Hz}$$

Measure the spacing, in period, between the two frequencies marked by the vertical blue lines. The first frequency is given in the slide, while the second one has to be computed from the fitted frequencies of the last two sinc^2 profiles (those marked by the arrows). To do so, we need to compute the average value between the two peaks. This is because the two peaks correspond to the components m=-1 and m=+1 since the inclination angle of the spin axis of this star is close to 90deg.

Questions

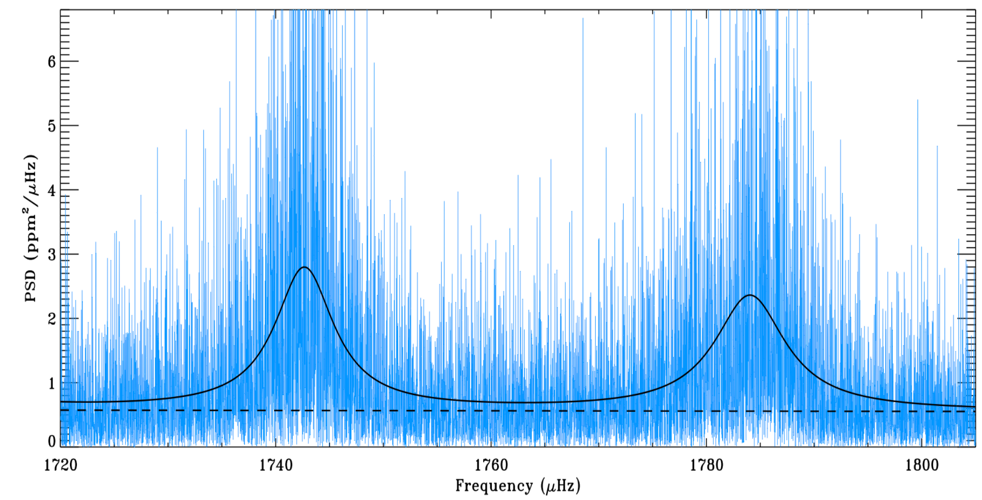


Compare the observed period spacing with this DP - Dnu diagram, and say what is the evolutionary stage of the star (either He-burning-core RG or H-burning-shell RG).

6. - PEAK SIGNIFICANCE TEST

Peak significance on multiple modes

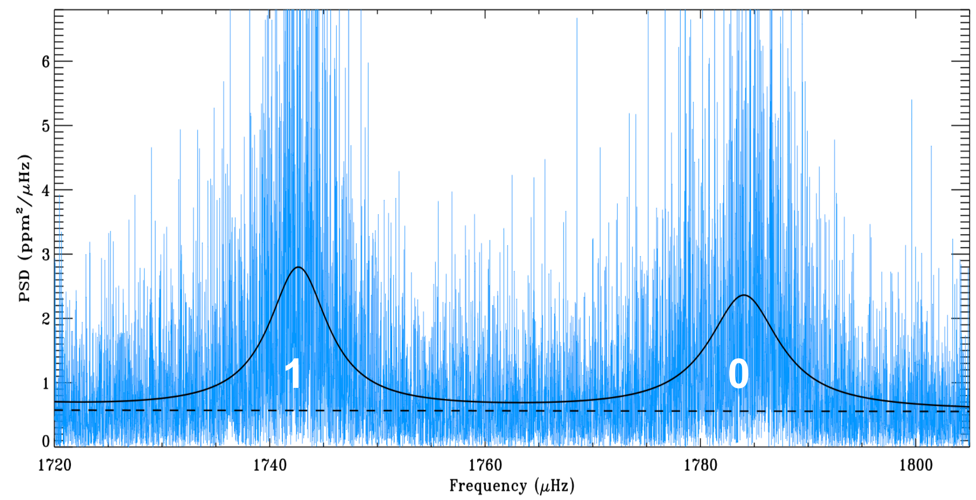
Spot the radial mode (l=0)



In this chunk of an F-type star we cannot clearly distinguish which of the two peaks is an l=1 (or l=0).

Peak significance on multiple modes

How can we test whether this is correct?

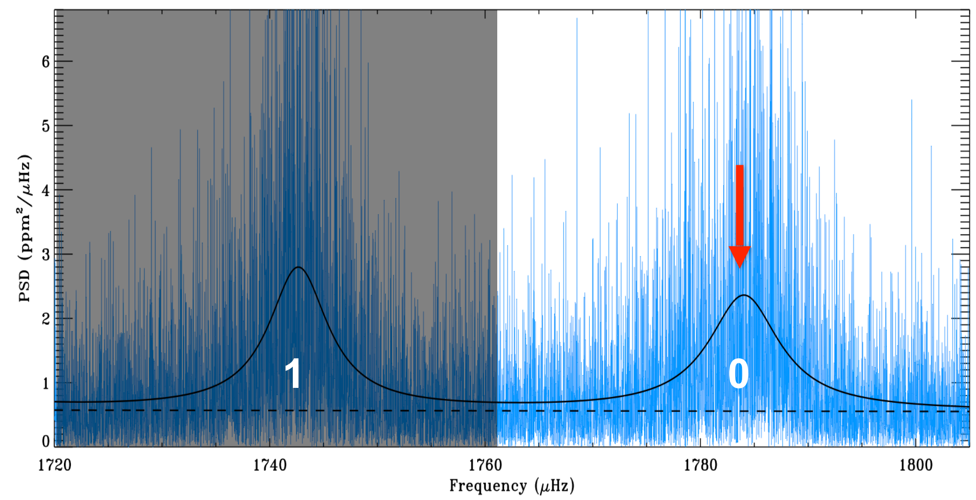


To test our interpretation we perform a Bayesian model comparison.

Peak significance on multiple modes

$\mathcal{M}_{\ell=0}$ **Only ℓ = 0**

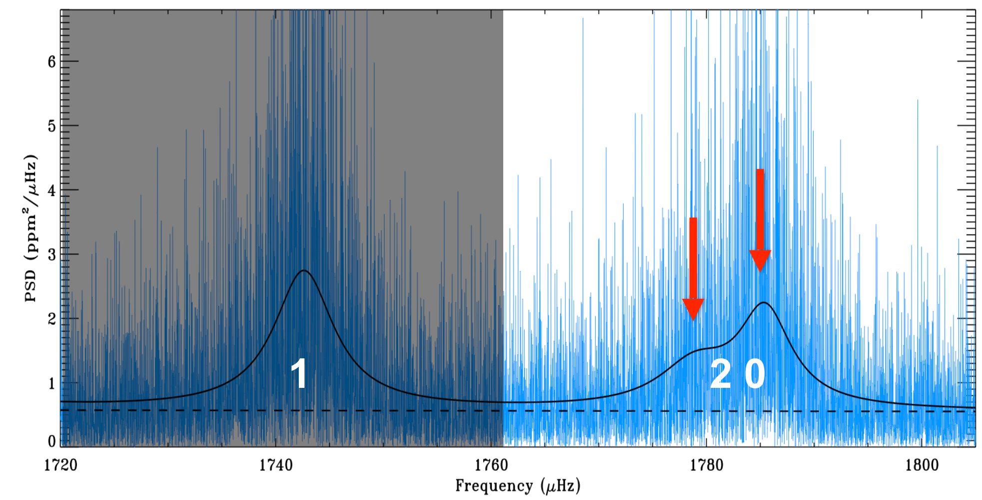
$\mathcal{E}_{\ell=0}$ **Bayesian Evidence**



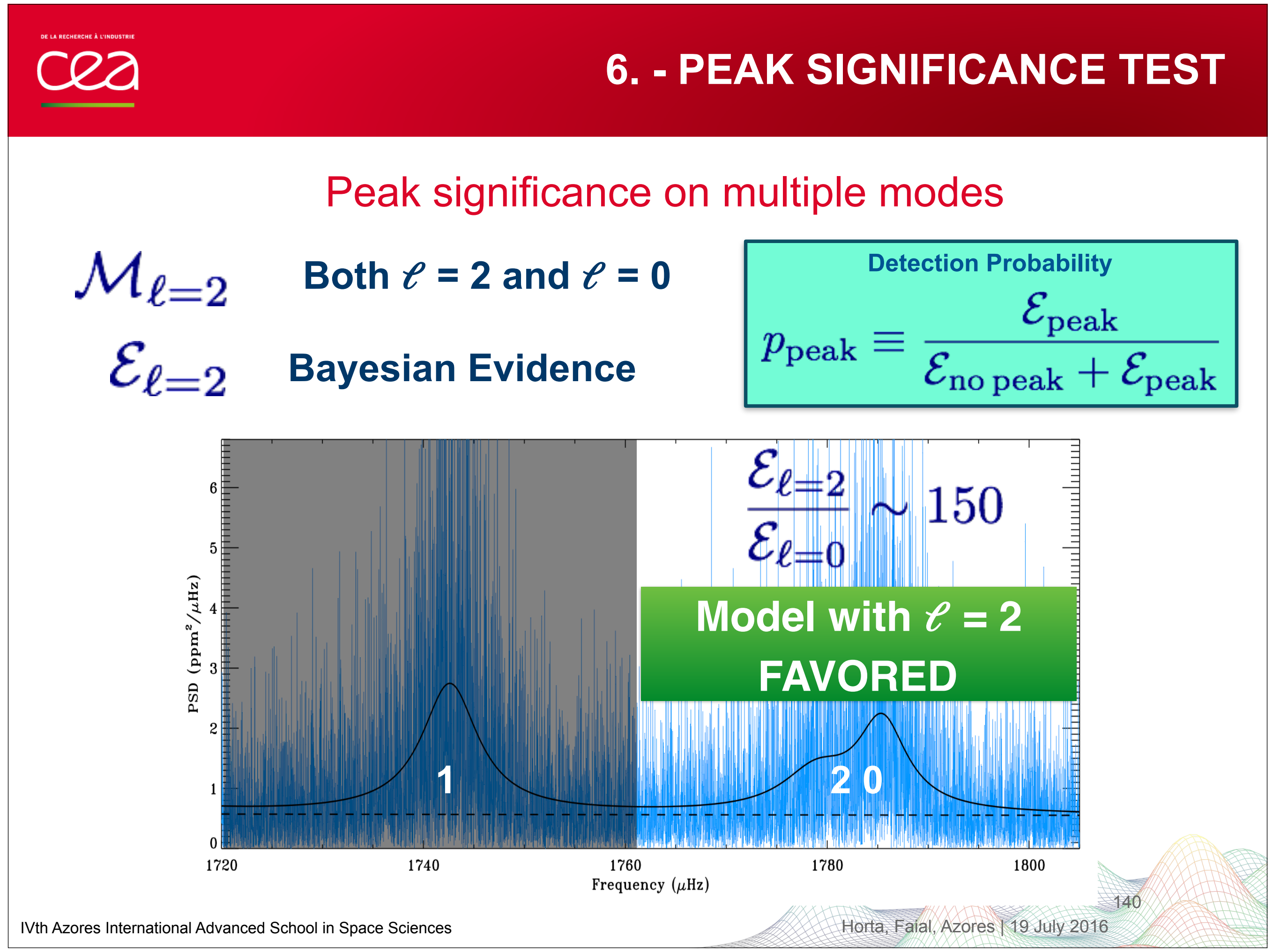
Peak significance on multiple modes

$\mathcal{M}_{\ell=2}$ **Both ℓ = 2 and ℓ = 0**

$\mathcal{E}_{\ell=2}$ **Bayesian Evidence**



To test the case of a double peak (with blending), we need three models (N_peaks + 1). One model will not include any peak, a second model will only include one peak, and a third model will include two peaks.

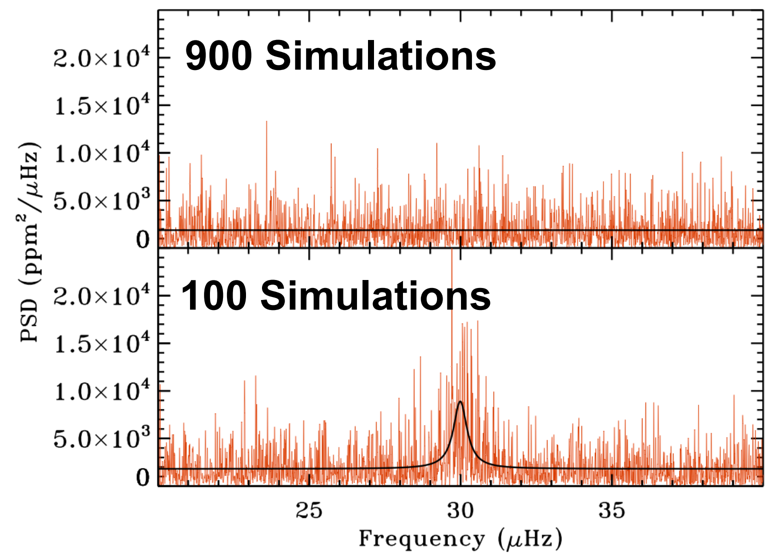


A useful way to note down the significance of peak is to compute its detection probability as defined in the slide. In a Bayesian framework, model should not be excluded even if their detection probability is very close to 0%. An honest and correct way of reporting results arising from a Bayesian model comparison is to quote all the models tested, and their corresponding probabilities.

6. - PEAK SIGNIFICANCE TEST

Simulation calibration

- Simulations test
- **1000** artificial chunks of PSD
- Blind search for those with a peak



Bayes factor

$$B_{yes,no} = \frac{\mathcal{E}_{yes}}{\mathcal{E}_{no}}$$

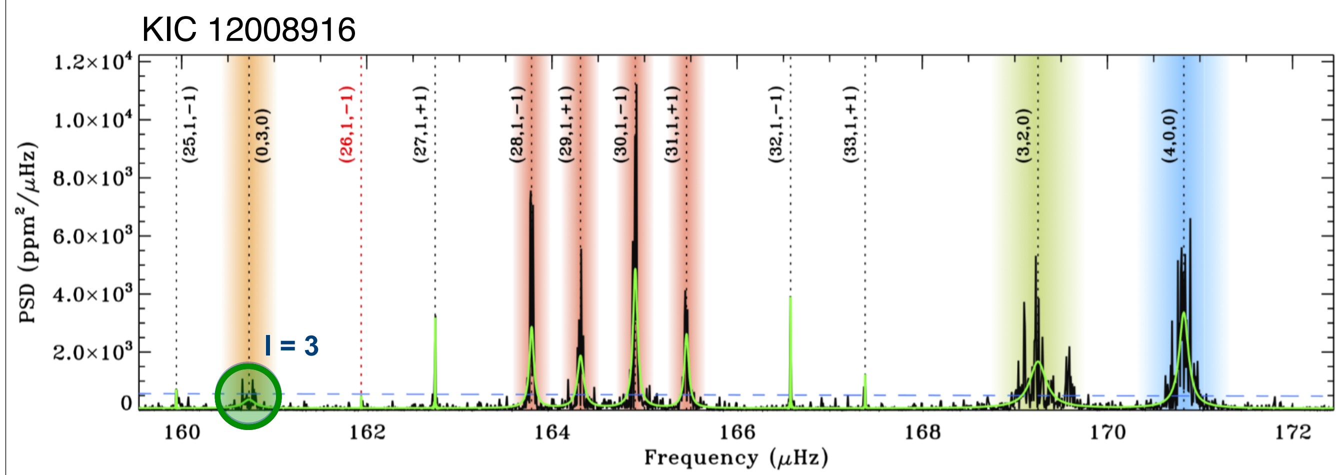
$$B_{yes,no} \sim 150$$

Strong Evidence (Jeffreys' scale)

All peaks found!

In a blind search for 100 oscillation peaks hidden in a set of 1000 simulations, we could find all those simulations containing an oscillation peak by adopting a strong evidence condition on the Bayes factor. A strong evidence is what I recommend to conclude on the significance of a model.

Perform the significance test on a peak



Test the significance of the l=3 mode by means of the Bayesian evidence.
Provide the value of the natural logarithm of the Bayes factor.

To-do list (1)

- Open your terminal and cd **~/PeakBagging/results/KIC012008916/**
- Run the script **copy_run** with the following command **./copy_run run_4 run_4A**
- Open the file **run_4A/resolvedPeaks_hyperParameters.txt** and comment out the block corresponding to the peak you want to test (*l=3*)

```
# 165.2833348340     165.6142324169
#  12.7373605996      38.2120817988
#   0.0257881623       0.1289408062

 169.0510643370     169.3895049224
  16.4600241256      49.3800723769
   0.0591908344       0.2959541604
```

To-do list (2)

- Now cd **~/PeakBagging/build/**
- Run again the multirun script via terminal but this time as **./multirun 012008916 0 1 run_4A**

What are you doing exactly?

What is the difference between run_4 and run_4A?

Why do you need two different runs?

Results

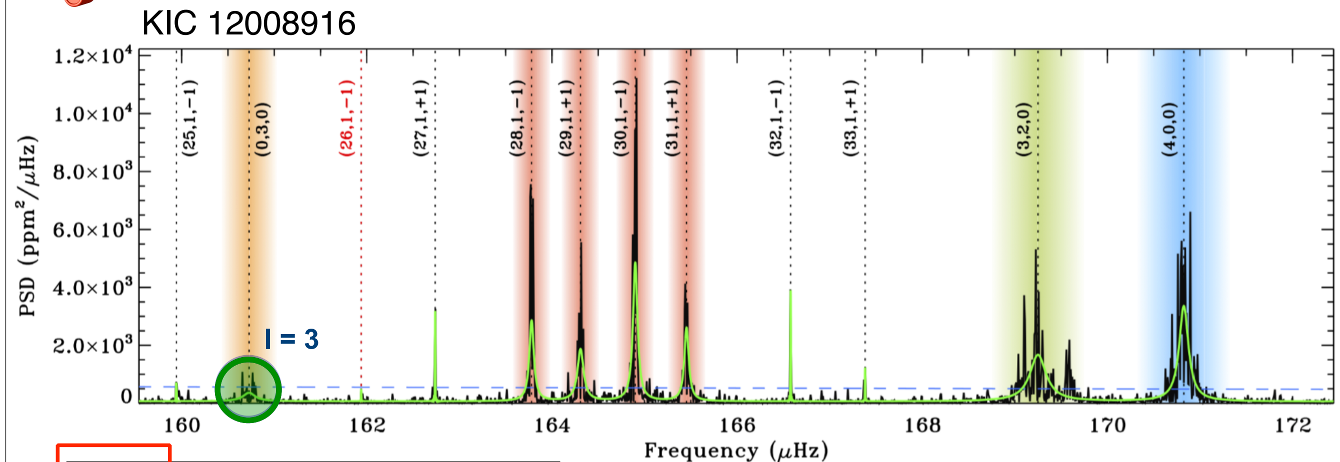
- You need to compare results that are stored in the two folders

~/PeakBagging/results/KIC012008916/run_4/00/

~/PeakBagging/results/KIC012008916/run_4A/00/

6. - PEAK SIGNIFICANCE TEST

Perform the significance test on a peak



$\vert\ln B_{01}\vert$	Odds	Probability	Strength of evidence
< 1.0	$\lesssim 3:1$	< 0.750	Inconclusive
1.0	$\sim 3:1$	0.750	Weak evidence
2.5	$\sim 12:1$	0.923	Moderate evidence
5.0	$\sim 150:1$	0.993	Strong evidence

~/PeakBagging/results/KIC012008916/results/run_4/
~/PeakBagging/results/KIC012008916/results/run_4A/

$$\ln \mathcal{B}_{i,j} = \ln \mathcal{E}_i - \ln \mathcal{E}_j \;?$$

Test the significance of the l=3 mode by means of the Bayesian evidence.
Provide the value of the natural logarithm of the Bayes factor and say what is the strength of evidence that you have found.
What should you change in the set up of the fitting to test this peak?
The peak is strongly significance (A natural logarithm of the Bayes factor > 40), hence we assume it is detected with a probability of 100%.

Commissariat à l'énergie atomique et aux énergies alternatives
Centre de Saclay | 91191 Gif-sur-Yvette Cedex
T. +33 (0)1 69 08 27 25 | F. +33 (0)1 69 63 73 88

DSM
IRFU
SAp

Etablissement public à caractère industriel et commercial | RCS Paris B 775 685 019

1. - BAYESIAN STATISTICS

2. - NESTED SAMPLING MONTE CARLO

3. - THE DIAMONDS CODE

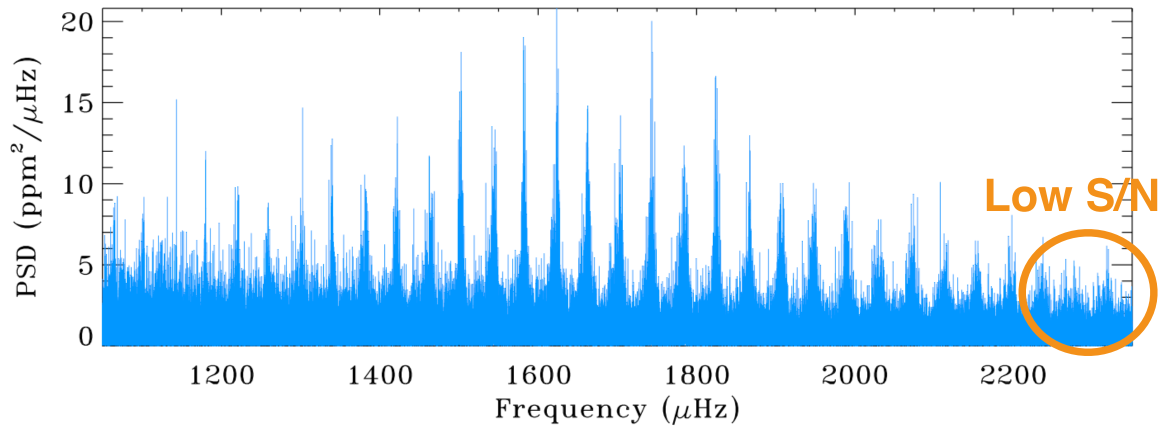
4. - FITTING THE BACKGROUND SIGNAL

5. - FITTING THE OSCILLATION MODES

6. - PEAK SIGNIFICANCE TEST

7. - MULTI-MODAL FITTING

Problems in the fitting of the oscillations



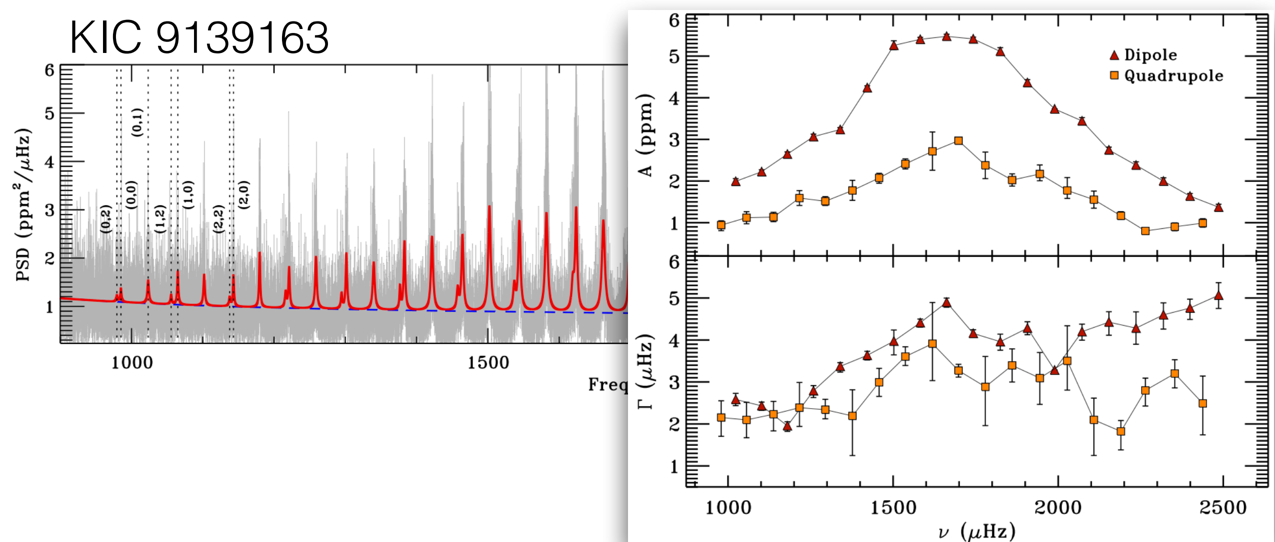
- **Problem 1**: big dataset + fitting numerous oscillation modes (peaks) per star (can be more than 100)
- **Problem 2**: testing if a peak is real or not (noise)

Problem 1

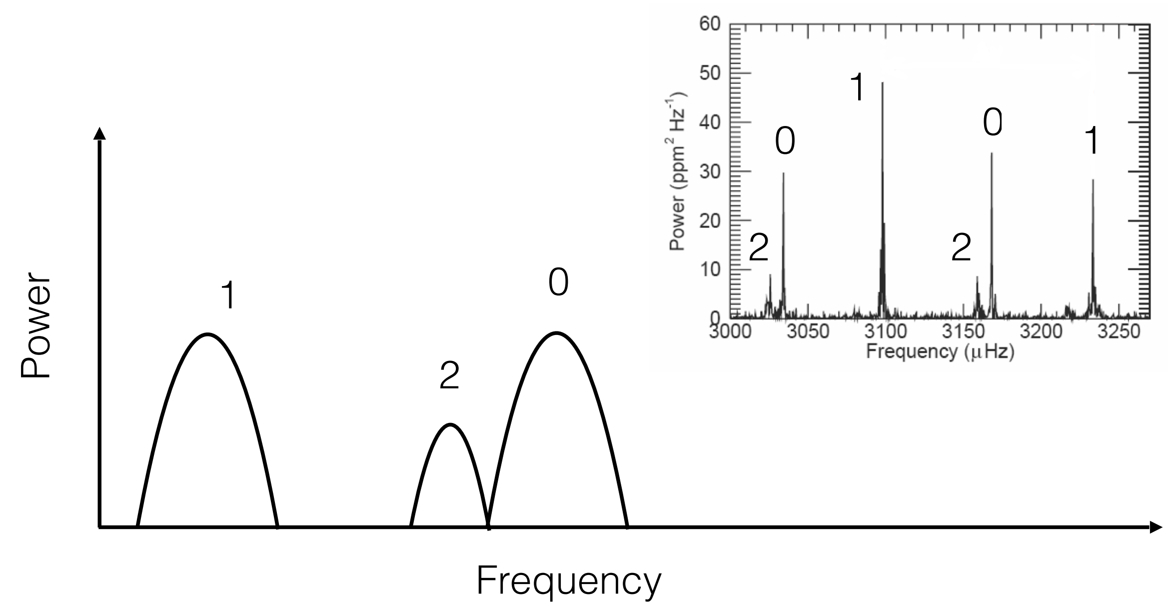
Solving a high-dimensional fitting problem

High-dimensional Model

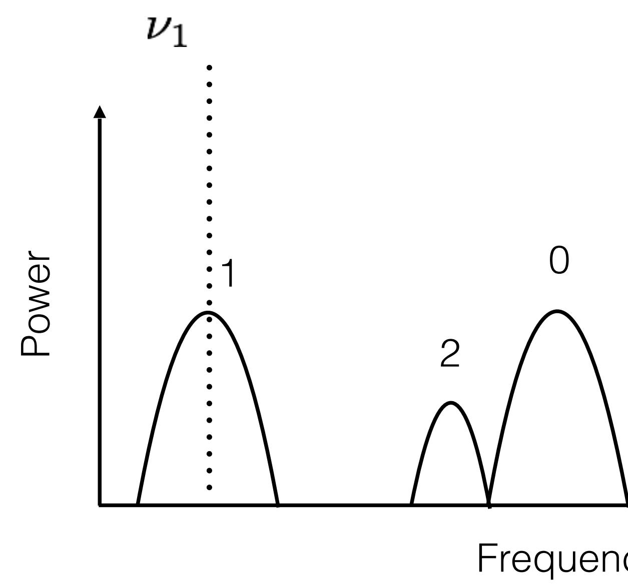
About 180 free parameters!
Computational time increases a lot

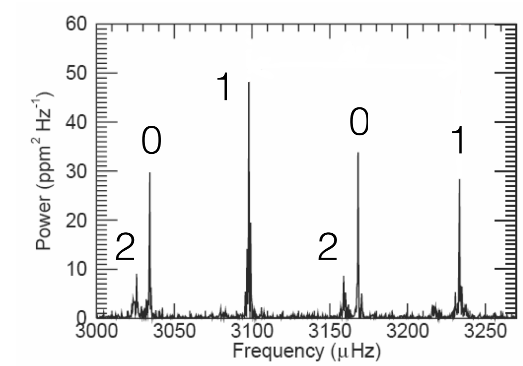


Multi-modal Model

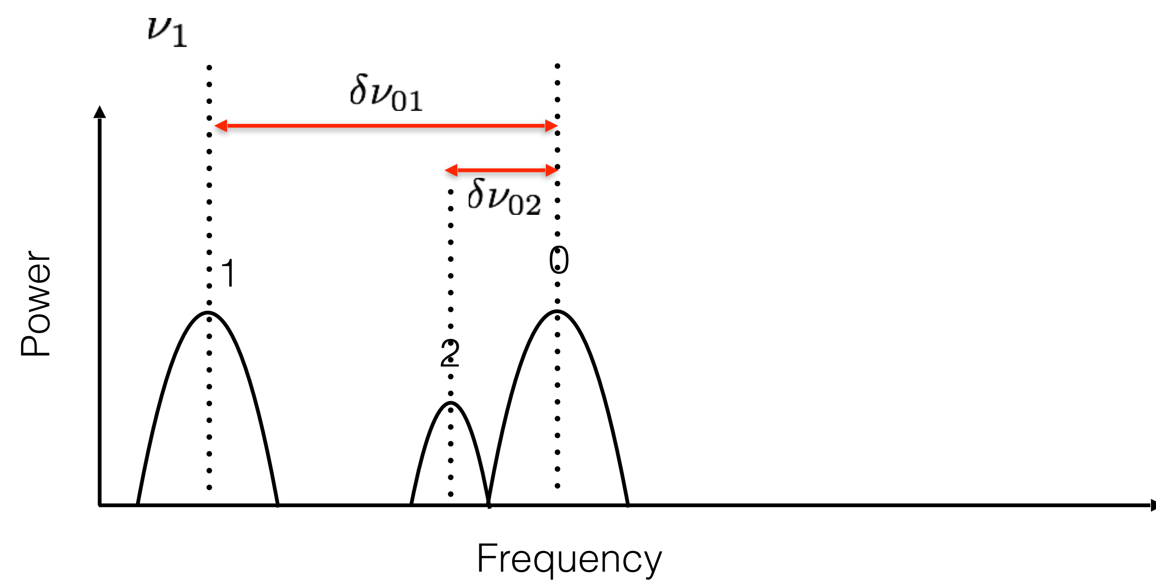


Multi-modal Model

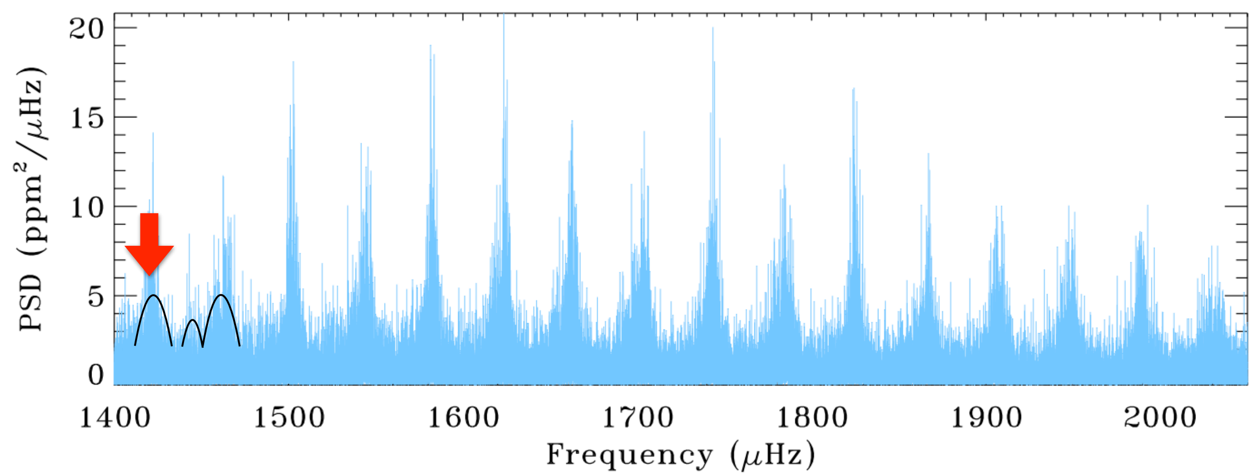




Multi-modal Model



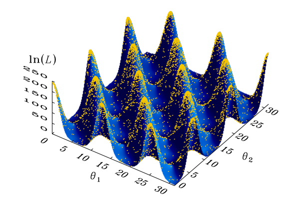
Multi-modal Model



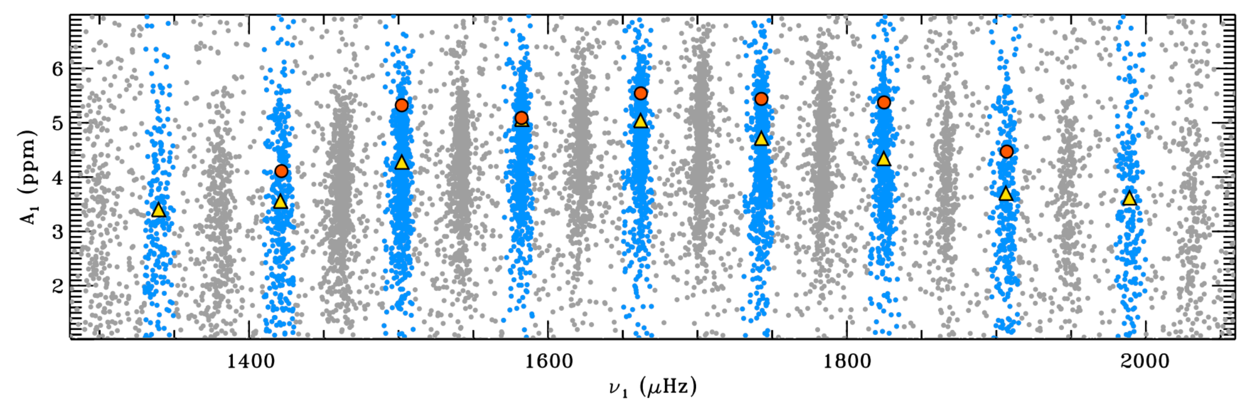
Corsaro & De Ridder 2014 A&A, 571, 71

Results

Multi-modal inference problem on 9 consecutive radial orders (27 peaks)



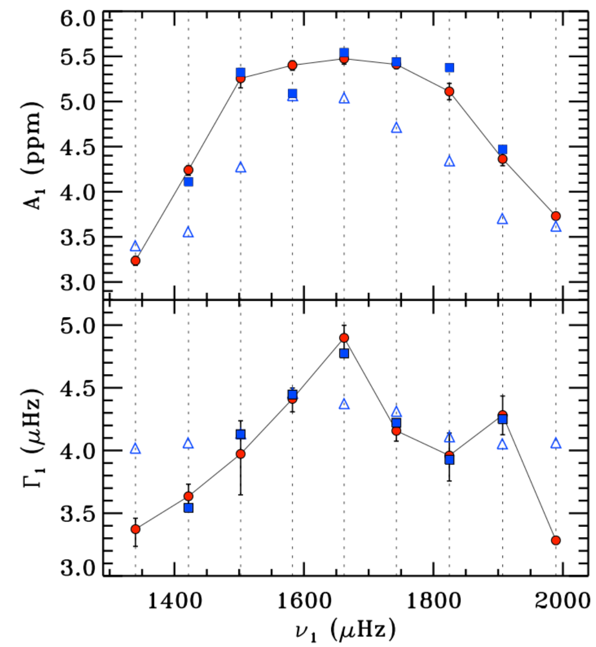
Only 9 free parameters!



Corsaro & De Ridder 2014 A&A, 571, 71

Comparison

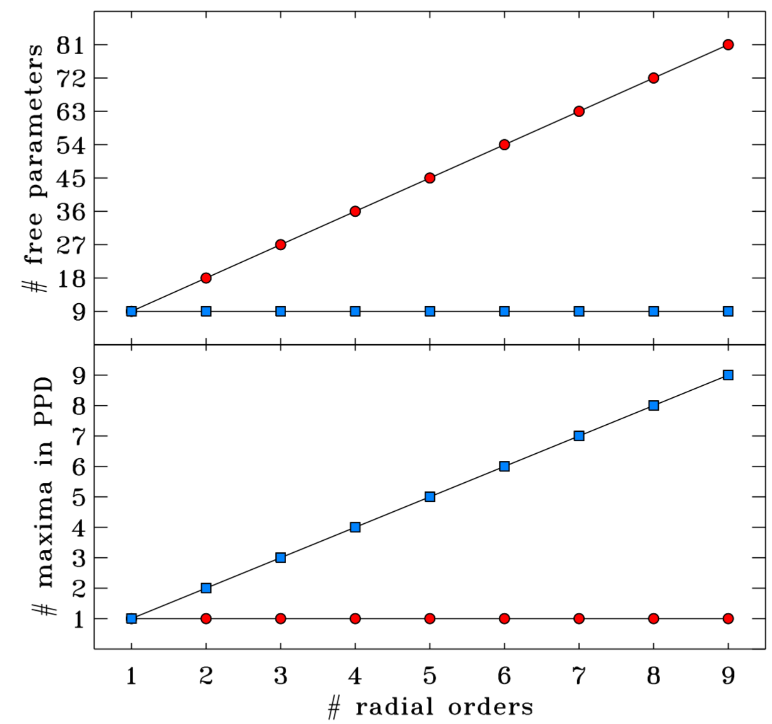
Red: uni-modal fit
Blue: multi-modal fit



Corsaro & De Ridder 2014 A&A, 571, 71

Comparison

Red: uni-modal fit
Blue: multi-modal fit



Corsaro & De Ridder 2014 A&A, 571, 71