# quickBayes: An analytical approach to Bayesian marginal likelihoods

Anthony Lim[1]

1 Science and Technology Facilities Council, Rutherford Appleton Laboratory, Harwell Campus, Didcot, Oxfordshire, OX11 0QX

Summary

It is common in science to have multiple hypotheses that could describe your data. Each one of these hypothesis provides a mathematical description of the data and gives unique physical insight into the data. However, it is not always obvious which hypothesis is the correct one and to demonstrate this figure 1 show Quasi Elastic Neutron Scattering (QENS) data fitted with one and two Lorentzian peaks. The quickBayes package is designed to make it easier for users to determine which hypothesis is correct given their data. At present there are example workflows for:

- Determining the number of Lorentzians in quasielastic neutron scattering data.
- Determining the number of exponential decays in MuSR (muon spectroscopy) data.

The quickBayes package comes with an API that can be easily be extended by users to calculate the most likely hypothesis given their data.
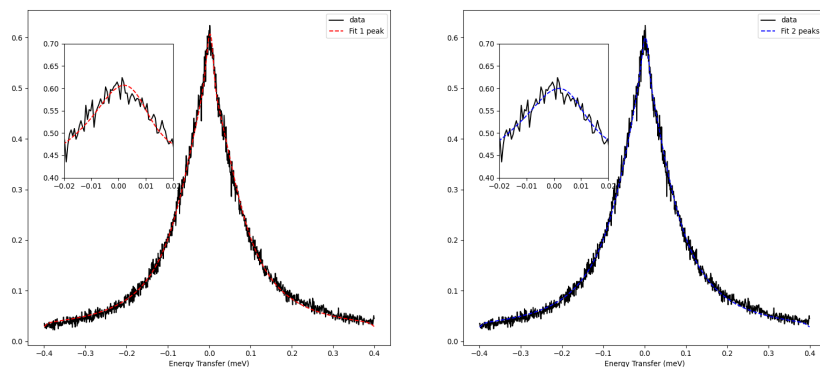
Figure 1: Two plots of the same raw data from a Quasi Elastic Neutron Scattering (QENS) experiment, showing the fits for one and two peaks (it also includes a linear background and an elastic peak). The inserts show zoomed in images of the peak centre. From the loglikelihood calculation the most likely number of peaks is two.

Statement of need

The quickBayes package started as a replacement for the quasielasticbayes package (Sivia & Howells, 2023), which was used for:

- Determining the most likely number of Lorentzians in QENS data.

- Calculating the loglikelihood of a stretched exponential in QENS data.
- Calculating a contour of the probability of a stretched exponential as a function of beta and sigma (full width half maxima).

The `quasielasticbayes` package is highly successful within the field of QENS (Sivia et al., 1992). During the development of `quickBayes` it became clear that the fundamental concept would benefit other areas of research such as MuSR and was written to be easily extendable into new domains. The new code is written in Python with clear sections for fitting functions, fitting engines and workflows. The workflows are designed to use the fitting functions and engines as building blocks, allowing the workflow to focus on the specific steps of the data analysis (e.g. splines, rebinning). The code has been developed to make it easy to read and understand, with comprehensive automated testing of the functionality. These changes make the new code more maintainable and reliable, while ensuring it is easily extendable by the user community. In previous iterations of the quickBayes and quasielasticbayes packages, the term loglikelihood was used as shorthand to describe the natural logarithm of the marginal likelihood or model evidence.

Traditionally the calculation of the marginal likelihood for a model, also known as the evidence, is not analytic and needs to be done with Bayesian inference. These are computationally expensive methods, such as Markov chain Monte Carlo (von Toussaint, 2011) or Nested Sampling (Skilling, 2006). Simpler and far less computationally demanding methods for model comparison exist, such as the Bayesian Information Criterion or Akaike Information Criterion. These use the maximum likelihood and some basic information theory to prevent over-fitting by incorporating a penalty for more complex models. The packages `quasielasticbayes` and `quickBayes` attempt to provide a middle ground by using a more robust Bayesian approach, while maintaining a fast analytical expression for the evidence (Sivia & Skilling, 2006). These simple assumptions reduce the computational cost significantly allowing users to quickly determine the most likely hypothesis given their data. In the QENS community it is difficult to determine if the data contains one or two Lorentzians, the `quasielasticbayes` and now `quickBayes` packages provide a way to quickly calculate which is more likely.

The `quasielasticbayes` package was written in Fortran and had become almost impossible to maintain due to poor code quality and a lack of unit testing. Furthermore, compiling the package only worked in very specific cases due to floats being frequently converted between real and complex variables when passed between functions. Therefore, a new code was required that captured the key functionality of the `quasielasticbayes` package and led to the development of `quickBayes`.

Acknowledgments

References

Sivia, D. S., Carlile, C. J., Howells, W. S., & König, S. (1992). Bayesian analysis of quasielastic neutron scattering data. *Physica B: Condensed Matter*, *182*(4), 341–348. https://doi.org/10.1016/0921-4526(92)90036-R

Sivia, D. S., & Howells, W. S. (2023). *Quasielasticbayes version 0.2.0*. https://github.com/mantidproject/quasielasticbayes

Sivia, D. S., & Skilling, J. (2006). *Data analysis: A Bayesian tutorial* (Second). Oxford University Press. ISBN: 9780198568322

Skilling, J. (2006). Nested sampling for general Bayesian computation. *Bayesian Analysis*, *1*(4), 833–859. https://doi.org/10.1214/06-BA127

von Toussaint, U. (2011). Bayesian inference in physics. *Rev. Mod. Phys.*, *83*, 943–999. https://doi.org/10.1103/RevModPhys.83.943