

# SYSBIONS: nested sampling for systems biology

Rob Johnson<sup>†,\*</sup>, Paul Kirk<sup>†</sup> and Michael P. H. Stumpf<sup>\*</sup>

Centre for Integrative Systems Biology and Bioinformatics, Department of Life Sciences, Imperial College London, London SW7 2AZ, UK

Associate Editor: Janet Kelson

## ABSTRACT

**Motivation:** Model selection is a fundamental part of the scientific process in systems biology. Given a set of competing hypotheses, we routinely wish to choose the one that best explains the observed data. In the Bayesian framework, models are compared via Bayes factors (the ratio of evidences), where a model's evidence is the support given to the model by the data. A parallel interest is inferring the distribution of the parameters that define a model. Nested sampling is a method for the computation of a model's evidence and the generation of samples from the posterior parameter distribution.

**Results:** We present a C-based, GPU-accelerated implementation of nested sampling that is designed for biological applications. The algorithm follows a standard routine with optional extensions and additional features. We provide a number of methods for sampling from the prior subject to a likelihood constraint.

**Availability and implementation:** The software SYSBIONS is available from <http://www.theosysbio.bio.ic.ac.uk/resources/sysbions/>

**Contact:** m.stumpf@imperial.ac.uk, robert.johnson11@imperial.ac.uk

Received on April 9, 2014; revised on July 30, 2014; accepted on October 13, 2014

## 1 INTRODUCTION

Given a set of models proposed to explain some observation, we seek to rank them according to the extent to which they are supported by some data. Likelihood-based approaches find the point at which the likelihood function is maximized, and compare models based on these maxima (Burnham and Anderson, 2002). Bayesian approaches for model selection rest on Bayes factors: the ratio of evidences of competing models. A number of methods exist to estimate the evidence (Kirk *et al.*, 2013), a metric of the support afforded to a model by some data.

Nested sampling is a Bayesian method for evidence estimation and parameter inference for systems where a likelihood function can be defined (Skilling, 2006). As the algorithm progresses, it generates samples from the posterior parameter distribution directly.

We present a C-based nested sampling tool for computational biologists. The user supplies a likelihood function, some experimental data and the prior parameter distribution. The program returns a value for the evidence alongside samples from the posterior parameter distribution. There exists a Fortran-based nested sampling package, MultiNest (Feroz *et al.*, 2009), used in the astrophysics community. Our work is aimed specifically at

the biological community and includes an SBML (Systems Biology Markup Language, Rodriguez *et al.*, 2007) parser so that models can be specified according to current standards. The recent growing use of nested sampling in systems biology invites the release of a tool implementing the method (Aitken and Akman, 2013; Burkoff *et al.*, 2012; Dybowski *et al.*, 2013; Kirk *et al.*, 2013; Pullen and Morris, 2014).

## 2 APPROACH

The evidence is defined as  $Z = \int_{\Theta} \ell(\theta)\pi(\theta) d\theta$ , where  $\theta$  is the parameter set (and  $\Theta$  the parameter space),  $\ell$  the likelihood function and  $\pi$  the prior. The change in notation  $\pi(\theta) d\theta = dX(\theta)$ , where  $X(\theta)$  is the cumulative density function, allows the integral to be written  $Z = \int_0^1 \ell(\theta) dX(\theta)$ . This can be approximated as a sum,  $Z \approx \sum_{i=1}^N \ell_i W_i$ , where  $N$  points are sampled and  $W_i$  is the proportion of prior mass represented by point  $i$ , calculated as the difference between the volume enclosed by the contour of constant likelihood through  $\ell_i$  and that through  $\ell_{i-1}$ . Nested sampling is a method for generating the sequence of points  $\{\ell_i, W_i\}$ .

For a thorough presentation of nested sampling, we refer the reader to the work of Skilling (2006) and Sivia and Skilling (2006). For our purposes, we follow the general algorithm:

1. Initialise  $Z = 0$
2. Generate  $N$  points from  $\pi(\theta)$
3. **for**  $i = 1 : M$ 
  - a. Find  $\theta^*$  with lowest likelihood,  $\ell^*$
  - b. Calculate  $W_i = \exp(-\frac{\ell^*}{N}) - \exp(-\frac{\ell_{i-1}}{N})$
  - c. Set  $Z = Z + \ell^* W_i$
  - d. Resample  $\theta^* \sim \pi(\theta) |_{\ell(\theta) > \ell^*}$
4. **end for**
5. Set  $Z = Z + \sum_{j=1}^N \ell_j \exp(-\frac{M}{N}) / N$

Our program is written primarily in C with additional capability for GPU acceleration. Other features include an SBML parser for automated generation of likelihood functions (Lieve *et al.*, 2010) and plotting tools. For the task of sampling from the prior subject to a likelihood constraint (step 3d), we provide three methods. The accuracy of the approximation in step 3b depends on the population of  $N$  points (live points) being truly distributed as the prior within the given likelihood constraint (Skilling, 2006).

## 3 METHODS

Our nested sampling package is a command-line tool for Linux and MacOSX platforms. Pre-requisites are listed in the accompanying

\*To whom correspondence should be addressed.

†The authors wish it to be known that, in their opinion, the first two authors should be regarded as Joint First Authors.

**Table 1.** Input options

| Variable              | Tag           | Input            | Default                                    |
|-----------------------|---------------|------------------|--|
| Number of live points | nLive         | **integer        | 1000                                       |
| Number of iterations  | maxIter       | **integer        | on, 10 000                                 |
| Tolerance             | tol           | decimal          | off, 0.001                                 |
| *Parameters           | constant      | **value          | none                                       |
|                       | uniform       | ** bounds        | none                                       |
| Sampling method       | rejection     | none             | off  |
|                       | rw            | none             | off  |
|                       | ellipsoid     | expansion factor | on, 2                                      |
| Restart from file     | Restart       | file paths       | _restart_points.txt,<br>_restart_input.txt |
|                       |               | file path root   | _restart                                   |
| Write restart         | write_restart | file path root   | _restart                                   |
| Points to leap        | nLeap         | **integer        | 1  |
| Adaptive leaping      | adaptive      | none             | off  |
| CUDA                  | cuda          | **nLeap,         | off, none,                                 |
|                       |               | **max. threads   | none                                       |

\*required; \*\*required if tag given.

manual. The user supplies a likelihood function, either by editing a template file or using an SBML file. An executable is then made that receives input from the command line. When the program is run, live points are generated and their likelihoods evaluated according to the function supplied by the user. On completion, it returns the calculated evidence with standard deviation, samples from the posterior, trajectories generated by points from the posterior and files from which the algorithm can be restarted.

### 3.1 Algorithm options

Options available to the user are listed in Table 1. The only required input is the parameter set (all other variables have default values). Parameters may be constant or inferred subject to a uniform prior distribution, for which lower and upper bounds must be supplied. The algorithm can be terminated either by specifying the number of iterations, or by monitoring the rate at which the evidence accumulates: the loop terminates at iteration  $m$  if  $\ell_m W_m / \sum_{i=1}^m \ell_i W_i < tol$ .

### 3.2 Sampling methods

We include three sampling methods for step 3d of the algorithm: rejection, for perfectly sampling from the prior, and random walk (following Sivia and Skilling, 2006) and ellipsoidal (following Feroz *et al.*, 2009) for refined sampling with reduced computational cost.

**Rejection:** The rejection method samples from the prior as initially defined, accepting the point if its likelihood value is within the constraint and rejecting otherwise. This method remains true to the requirement that samples are taken from the prior subject to the likelihood constraint, but its efficiency is poor: as the lowest likelihood increases, the acceptance rate becomes prohibitively small.

**Random walk:** The random-walk method duplicates a point randomly chosen from the current live-point population and walks it 20 steps, accepting the new point at each step if its likelihood is within the constraint. The steps are scaled according to the covariance among the present population, and scaled further to converge to an acceptance rate of 0.5 (Sivia and Skilling, 2006).

**Ellipsoidal:** The ellipsoidal method (Mukherjee *et al.*, 2006) creates an ellipsoid surrounding the current population of live points, expanded by some user-supplied factor. The new point is sampled from within

the ellipsoid. This increases the acceptance rate but risks excluding areas of prior mass that lie inside the current likelihood constraint.

### 3.3 Output

A summary file of input and output information is created, documenting the number of live points, number of iterations, tolerance, sampling method and parameter ranges, followed by the evidence with standard deviation, the prior-to-posterior information gain and the means of all parameters and their standard deviations. Posterior distributions of the parameters can be plotted individually as histograms and in pair-wise scatter plots using the data stored in the posterior file. Finally, a file of trajectories is created that can be compared against the input data.

Restart files are created, documenting input parameters that must persist upon restart (such as the number of live points) and listing all points, live and discarded. These files can be used to restart the program from where it completed. It is also possible to specify the path to where the restart files are written.

## 4 SUMMARY

We present SYSBIONS, a computational tool for model selection and parameter inference using nested sampling. Using a data-based likelihood function, our package calculates the evidence of a model and the corresponding posterior parameter distribution.

## ACKNOWLEDGEMENT

We gratefully acknowledge the help and support of the Theoretical Systems Biology Group at Imperial College London.

**Funding:** This work was supported through a BBSRC PhD studentship to R.J. and an HFSP grant [RGP0061/2011] to P.K. and M.P.H.S.

**Conflict of interest:** none declared.

## REFERENCES

- Aitken,S. and Akman,O.E. (2013) Nested sampling for parameter inference in systems biology: application to an exemplar circadian model. *BMC Syst. Biol.*, **7**, 72.
- Burkoff,N.S. *et al.* (2012) Exploring the energy landscapes of protein folding simulations with Bayesian computation. *Biophys. J.*, **102**, 878–886.
- Burnham,K.P. and Anderson,D.R. (2002) *Model Selection and Multimodel Inference: A Practical Information-Theoretic Approach*. Springer, New York.
- Dybowski,R. *et al.* (2013) Nested sampling for Bayesian model comparison in the context of *Salmonella* disease dynamics. *PLoS One*, **8**, e82317.
- Feroz,F. *et al.* (2009) MultiNest: an efficient and robust Bayesian inference tool for cosmology and particle physics. *Mon. Not. R. Astron. Soc.*, **398**, 1601–1614.
- Kirk,P. *et al.* (2013) Model selection in systems and synthetic biology. *Curr. Opin. Biotech.*, **21**, 1–8.
- Liepe,J. *et al.* (2010) ABC-SysBio—approximate Bayesian computation in Python with GPU support. *Bioinformatics*, **26**, 1797–1799.
- Mukherjee,P. *et al.* (2006) A nested sampling algorithm for cosmological model selection. *Astrophys. J.*, **638**, L51–L54.
- Pullen,N. and Morris,R.J. (2014) Bayesian model comparison and parameter inference in systems biology using nested sampling. *PLoS One*, **9**, e88419.
- Rodriguez,N. *et al.* (2007) SBMLeditor: effective creation of models in the Systems Biology Markup Language (SBML). *BMC Bioinformatics*, **8**, 79.
- Sivia,D. and Skilling,J. (2006) *Data Analysis: A Bayesian Tutorial*. Oxford University Press, Oxford.
- Skilling,J. (2006) Nested sampling for general Bayesian computation. *Bayesian Anal.*, **1**, 833–860.