MixSIR (Version 1.0) Manual

Developed by:

B.X. Semmens and J.W. Moore

Please Cite As:
B.X. Semmens and J.W. Moore 2008. MixSIR: A Bayesian stable isotope mixing model, Version 1.0. http://www.ecologybox.org. Date of Download: ???

Also see:

Moore, J. W. and Semmens, B. X. 2008. Incorporating uncertainty and prior information into stable isotope mixing models. Ecology Letters, 11, 470-480.

DISCLAIMER

Be extremely careful when using this program for serious statistical analysis. Before accepting model results, please familiarize yourself with common problems in Bayesian statistics, namely-- the general problem of developing appropriate posteriors, assumptions regarding the distribution of the data (in this case Gaussian and Dirichlet distributions), and the blessings and curses of prior specification. In addition, a model is only as good as the data—please understand the limitations of your data and use common sense. This manual only concerns the functionality of MixSIR.

INTRODUCTION

MixSIR is a graphical user interface (GUI) program built on the MATLAB platform that carries out Bayesian analysis of stable isotope mixing models using sampling-importance-resampling (SIR). A Bayesian approach to stable isotope mixing models is advantageous because it allows one to: 1) explicitly account for uncertainty in isotope values when estimating the contribution of sources to an isotope mixture, 2) characterize uncertainty in the estimates of source contributions based on underlying uncertainty in the mixture and source isotope values, as well as uncertainty due to 'source overparameterization' (i.e.- too many sources to allow a unique solution), and 3) include prior knowledge in the analysis. Because the program is based on the MATLAB platform, the MATLAB run-time library files must be installed (only once) on the machine you intend to run the program on. The MATLAB run-time library is freely distributed by Mathworks and is available at http://bio.research.ucsc.edu/people/moore/mixsir/MCRInstaller.exe. Note that this runtime library must be installed even if you already have Matlab installed on your computer. We have chosen to implement MixSIR in this way so that the underlying MATLAB code can be scrutinized and improved upon by others.

This manual briefly describes the theory behind MixSIR, the analytic techniques upon which MixSIR is built, and how to use the model for data analysis. At the front of this manual we have provided a 'Quick Start' guide, but we (again) caution that failing to grasp the theory behind the model before carrying out and reporting on analyses is hazardous. For a more thorough discussion of the model please see Moore and Semmens 2007 (citation given on the first page of this manual). This manual does not address the theory and science underlying stable isotope mixing models.

INSTALLATION

All the files needed to install and run the MixSIR program can be downloaded as a '.zip' file from http://www.ecologybox.org (part of the Greenboxes code sharing network). In theory this program will run on PCs, Unix, and Mac platforms, although it has only been tested on the PC platform to date.

PC Windows installations without MATLAB (R2006b)

Download the mcrinstaller.exe from http://bio.research.ucsc.edu/people/moore/mixsir/MCRInstaller.exe and install it on your machine. NOTE THAT YOU MUST HAVE ADMINISTRATIVE PRIVILAGES in order for mcrinstaller.exe to execute properly. The mcrInstaller.exe automatically:

1. Copies the necessary files to the target directory you specified.
2. Registers the components as needed.
3. Updates the system path to point to the MCR binary directory, which is <target_directory>/<version>/runtime/bin/win32.

When the installation completes, click Close on the Installation Completed dialog box to exit. Next download the MixSIR program from http://www.ecologybox.org and *double-click on the MixSIR.exe file within the directory you have just created in order to run the program*. We recommend you create a directory since the first time the mixsir.exe program is run, it will create a series of files necessary for it to run correctly. Please note that the program can take a considerable amount of time to start, depending on the performance of the computer you are using.

QUICK START GUIDE

1) Develop the data files outlined in the 'GETTING STARTED' section below. We recommend that you edit the existing files, as this will help you follow the correct format. You can easily open, edit and save tab delimited files with a spreadsheet program like Microsoft Excel or in a simple text editor such as Wordpad.

2) If you wish to use informative priors, edit the appropriate prior text file in the same manner. Note that improperly specified priors can compromise model function.

3) Specify the number of model iterations. We strongly suggest that you use at least 1,000,000 iterations. In some cases 10,000,000 or more iterations may be required to assure proper model function.

4) Specify whether you wish the products of the model run to be output to text files using the check box just above the iterations text box.

5) Click 'Giddyup', and enjoy the randomly colored progress bar.

6) Once the model runs are complete, a results and diagnostics window will appear, and results are graphed.

GETTING STARTED

Before you can use the MixSIR program, you will need to generate a series of tab-delimited data files that are specific to your mixing model. These data files must be named exactly as outlined, and placed in the same directory as MixSIR.exe. We have found that users make the fewest mistakes when they edit the existing data files rather than making new ones. **In the following section we provide a worked example of how to generate the data files**. The required files are as follows:

Mix_data.txt – This file contains the isotope data for the mix. Each isotope is a separate column, and each row is a mixture from a sample. For instance, if you are interested in determining the contribution of prey items to a predator mixture, each row of this file would give individual predator isotope values.

mean_source.txt—This file contains a matrix of mean isotope values (columns) for each of the sources (rows) used in the mixing model. The order of the isotope values and sources must be consistent across data files.

SD_source.txt—This file contains a matrix of the standard deviation of isotope values (columns) for each of the sources (rows) used in the mixing model. The order of the isotope values and sources must be consistent across data files.

mean_frac.txt—This file contains a matrix of mean isotope fractionation values (columns) for each of the sources (rows) used in the mixing model. The order of the isotope values must be consistent across data files. In most cases isotope-specific fractionation will be the same across sources, and thus all values in a given column are identical. These data can be found in the literature (see Moore and Semmens 2008).

SD_frac.txt—This file contains a matrix of isotope fractionation standard deviation values (columns) for each of the sources (rows) used in the mixing model. The order of the isotope values must be consistent across data files. In most cases isotope-specific fractionation will be the same across sources, and thus all values in a given column are identical. These data can be found in the literature (see Moore and Semmens 2008).

define.txt —(OPTIONAL- only necessary if you wish to specify informative priors) This file contains the vector of α values that parameterize the Dirichlet distirbution (one row) . These α values define the prior belief regarding the contribution of each source using the Dirichlet distribution – Dir(α).

GENERATING DATA FILES- An Example

In this example, we wish to use isotope data to estimate the proportional contribution of prey items to the diet of a predator (rainbow trout). We make the assumption that isotopic fractionation is identical across prey items. The data we have are as follows:

Type	Sample	Taxa	15N	13C
predator	individual	rainbow trout	10.78	-25.53
predator	individual	rainbow trout	12.99	-21.56
predator	individual	rainbow trout	13.76	-20.33
predator	individual	rainbow trout	12.86	-20.39
predator	individual	rainbow trout	12.67	-21.63
predator	individual	rainbow trout	12.69	-20.92
predator	individual	rainbow trout	11.96	-22.55
predator	individual	rainbow trout	11.22	-22.97
prey	average	benthic invert	3.34	-19.67
prey	average	salmon egg	11.63	-22.99
prey	average	sculpin	8.31	-23.44
prey	average	shrew	5.88	-23.87
prey	average	terrestrial invert	6.71	-26.69
prey	SD	benthic invert	0.78	2.45
prey	SD	salmon egg	0.46	0.91
prey	SD	sculpin	1.09	2.89
prey	SD	shrew	1.05	1.11
prey	SD	terrestrial invert	2.44	1.84
fractionation	average		2.30	0.40
fractionation	SD		1.61	1.20

The 'mix_data.txt' file would contain:

10.78	-25.53
12.99	-21.56
13.76	-20.33
12.86	-20.39
12.67	-21.63
12.69	-20.92
11.96	-22.55
11.22	-22.97

The 'mean_source.txt' file would contain:

3.34	-19.67
11.63	-22.99
8.31	-23.44
5.88	-23.87
6.71	-26.69

The ‘SD_source.txt’ file would contain:

0.78	2.45
0.46	0.91
1.09	2.89
1.05	1.11
2.44	1.84

The ‘mean_frac.txt’ file would contain:

2.30	0.40
2.30	0.40
2.30	0.40
2.30	0.40
2.30	0.40

The ‘SD_frac.txt’ file would contain:

1.61	1.20
1.61	1.20
1.61	1.20
1.61	1.20
1.61	1.20

RUNNING MIXSIR

After installation, executing the MixSIR.exe file will open up the MixSIR GUI. There will also be a DOS box that runs in the background and gives Matlab-specific notifications and warnings (which you can generally ignore). Do not close the DOS Box, as it will also quit the GUI. You will immediately note that there are only a few user-customizable features on the GUI. Specifically, the user can specify whether or not to use priors, the number of samples to be performed, and whether or not you would like the program to output model results in the form of text files. Except for the user input boxes on the GUI, all the other user-specified information required in order to run the program are designated in the text files outlined above (see 'Getting Started' above).

MODEL FUNCTION

We developed and implemented a stable isotope mixing model, hereafter referred to as MixSIR, using a Bayesian framework to determine the probability distributions for the proportional contribution (f_i) of each source *i* to the mixture of interest. Bayesian statistics offer a powerful means to interpret data because they can incorporate prior information, integrate across sources of uncertainty, and explicitly compare the strength of support for competing models or parameter values (Hilborn & Mangel 1997; Ellison 2004). For this application, Bayesian techniques allow for the estimation of posterior probability distributions for all f_i through numerical integration. This numerical integration requires randomly generating *q* proposed vectors of proportional source contributions $\mathbf{f}_q$ representing possible states of nature, where all f_i elements in $\mathbf{f}_q$ sum to unity. Based on Bayes theorem, the probability of each $\mathbf{f}_q$ is then calculated based on data and prior information (Hilborn & Mangel 1997; Ellison 2004) such that:

$$P(\mathbf{f}_q|data) = \frac{L(data|\mathbf{f}_q)*p(\mathbf{f}_q)}{\sum L(data|\mathbf{f}_q)*p(\mathbf{f}_q)} \quad (2)$$

where $L(data|\mathbf{f}_q)$ is the likelihood of the data given $\mathbf{f}_q$, $p(\mathbf{f}_q)$ represents the prior probability of the given state of nature being true based on prior information, and the denominator is a numerical approximation of the marginal probability of the data (a normalizing constant). The numerator $L(data|\mathbf{f}_q) * p(\mathbf{f}_q)$, hereafter referred to as the unnormalized posterior probability (Gelman *et al.* 2003), yields the absolute probability of a given $\mathbf{f}_q$ based on data and prior beliefs.

Suppose we are trying to estimate the contribution of *i* sources to a mixture of *j* isotopes. In MixSIR, isotope signatures from the mixture population constitute the data, and are assumed to be normally distributed. For instance, if we wish to determine the contribution of prey items to a predator

diet, the data would be isotope signatures from individual predators. Uncertainty in source isotope values are factored into the model by defining mean and variance parameters for each *i, j*. Prior beliefs regarding proportional source contributions are defined using Dirichlet distributions on the interval [0, 1].

In order to calculate the likelihood of the data given $\mathbf{f}_q$, the proposed proportional contributions are combined with both user-specified source isotope distributions and their associated user-specified fractionation distributions in order to develop resultant proposed isotope distributions for the mixture. The likelihood of this distribution given the mixture data is then determined by calculating the product of the likelihoods of each individual mixture isotope value given the proposed mixture distribution specific to that isotope. The proposed isotope distributions for the mixture are determined by solving for the proposed means $\hat{\mu}_j$ and standard deviations $\hat{\sigma}_j$ of the mixture based on the randomly drawn f_i values comprising a vector $\mathbf{f}_q$:

$$\hat{\mu}_j = \sum_{i=1}^{n} \left[f_i * \left(m_{j_{source_i}} + m_{j_{frac_i}} \right) \right] \tag{4}$$

$$\hat{\sigma}_j = \sqrt{\sum_{i=1}^{n} \left[{f_i}^2 * \left(s^2_{j\ source_i} + s^2_{j\ frac_i} \right) \right]} \tag{5}$$

Where $m_{j_{source_i}}$ is the mean of the *j*th isotope of the *i*th source, $m_{j_{frac_i}}$ is the mean fractionation of the *j*th isotope of the *i*th source, $s^2_{j\ source_i}$ is the variance of the *j*th isotope of the *i*th source, and $s^2_{j\ frac_i}$ is the

variance in fractionation of the the j^{th} isotope of the i^{th} source. Once the $\hat{\mu}_j$'s and $\hat{\sigma}_j$'s are determined, the likelihood of the data given the proposed mixture is calculated as:

$$L(x|\hat{\mu}_j, \hat{\sigma}_j) = \prod_{k=1}^{n} \prod_{j=1}^{n} \left[\frac{1}{\hat{\sigma}_j * \sqrt{2*\pi}} * \exp\left(-\frac{(x_{kj} - \hat{\mu}_j)^2}{2 * \hat{\sigma}_j^{\,2}} \right) \right] \quad (6)$$

where x_{kj} represents the j^{th} isotope of the k^{th} mixture in the data file. Next the likelihood of $\mathbf{f}_q$ given prior information (user-specified α each source i) is calculated according to a Dirichlet distribution:

$$L(\mathbf{f}_q|\alpha) = \frac{1}{B(\alpha)} \prod_{i=1}^{n} \mathbf{f}_{q_i}^{\alpha_i - 1} \quad (7)$$

Finally, the likelihood of $\mathbf{f}_q$ given prior information is multiplied by the likelihood of the mixture data given $\mathbf{f}_q$ in order to calculate the unnormalized posterior probability of $\mathbf{f}_q$ given priors and data.

We implemented the Hilborn (after Professor Ray Hilborn) sampling-importance-re-sampling (SIR) algorithm (Rubin 1988) to examine the posterior probability of a vector of proportional source contributions($\mathbf{f}_q$) through numerical integration. The Hilborn SIR method is functionally equivalent to a basic SIR algorithm with a uniform importance function such that the re-sample weight for a given state of nature $w(\mathbf{f}_q)$ is equal to the unnormalized posterior probability (Rubin 1988; McAllister & Ianelli 1997). However, rather than saving all initial samples in a file and subsequently re-sampling from this file based on $w(\mathbf{f}_q)$, the Hilborn SIR method establishes a threshold acceptance value (T) prior to sampling, and uses it to simultaneously re-sample as the unnormalized posterior probabilities for each $\mathbf{f}_q$ sample

are calculated. We used the Hilborn method because it is programmatically intuitive, and because it does not require all initial samples to be stored (advantageous for large model runs). The method works as follows:

1. Use 10% of user-specified model iterations to establish a threshold (*T*):
    a. set *T* to 0 before beginning iterations;
    b. for each threshold iteration, randomly draw values for each f_i in $\mathbf{f}_q$ (e.g. for a 3 source model, a contribution parameter draw might be 0.1, 0.1, 0.8);
        i. calculate the unnormalized posterior probability (*L*) of the parameter draw based on prior information and data;
        ii. if L is greater than the current *T*, then $T = L$.
2. Use all user-specified model iterations to develop samples and simultaneously resample based on *T* and a cumulative likelihood value (*C*):
    a. set *C* to 0 before beginning iterations;
    b. for each iteration, randomly draw values for each f_i in $\mathbf{f}_q$;
        i. calculate the *L* of the parameter draw based on prior information and data;
        ii. add *L* to cumulative likelihood, $C = C + L$.
        iii. If *C* exceeds *T* then save the $\mathbf{f}_q$ for that iteration in list of re-samples and adjust the cumulative likelihood value, $C = C - T$.

The SIR algorithm is well suited to models having relatively few parameters with well defined intervals. Because all of the parameters in the model are proportional contributions of each source, models will generally have few parameters. Additionally, since these parameters are proportions they are bounded in the interval 0-1. Finally, because bounded proportions must all sum to 1, parameter values have cross dependencies that can result in multi-modal posterior distributions, and thus may compromise basic Markov Chain Monte Carlo sampling techniques. Given these constraints, a SIR

algorithm is an effective method for re-sampling proportional parameter space in order to develop accurate posterior distributions.

Incorporating prior information: The Bayesian framework allows a user to establish informative priors to guide model estimates. As stated above, the probabilities of source contributions are evaluated against prior information according to the Dirichlet distribution (the multivariate generalization of the beta distribution). Because all elements of $\mathbf{f}_q$ must sum to unity, priors on the elements of $\mathbf{f}_q$ are not independent (Connor & Mosiman 1969). The Dirichlet prior formulation explicitly accounts for this unity constraint. Visualizing the Dirichlet distribution when there are greater than 3 sources is difficult. Fortunately, because the Dirichlet is the multivariate generalization of the beta distribution, it can easily discretized into individual beta distributions, where each marginal is defined as:

$$X_i \sim Beta\left[\alpha_i, \left(\sum_{i=1}^{n} \alpha_i\right) - \alpha_i\right]$$

Highly informative priors will often "sharpen" the peaks in the likelihood surface, and the model will consequently require more iterations to develop an appropriate posterior. Generally, the more data a user provides the model, the less influential prior information will be on the model. When all α's are set to 1, all source contributions are *a priori* equally likely (uninformative priors).

Advantage of the SIR

The SIR algorithm is well suited to models having relatively few parameters with well defined intervals. Because all of the parameters in the model are proportional contributions of each source, models will rarely have more than 2-10 parameters to evaluate. Additionally, since these parameters are proportions they are bounded in the interval 0-1. Finally, because the bounded proportions must all sum to 1, the parameter values have cross dependencies that can result in multi-modal posterior

distributions, and thus limit the applicability of basic Markov Chain Monte Carlo sampling techniques. Given these constraints, a SIR algorithm is a simple and effective method for re-sampling proportional parameter space in order to develop accurate posterior distributions.

Model Performance

Because the SIR algorithm we have implemented draws proposals uniformly over proportional parameter space, it is at heart a 'brute force' method of Bayesian analysis. That said, the more iterations you carry out, the more accurate the posterior (model output) will be. For best results MAKE MODEL RUNS AS LONG AS YOU CAN AFFORD. A good rule of thumb is that the model run size you choose should produce 1,000 or more posterior draws. The actual number of iterations required (and thus the amount of time required) to generate these posterior draws will depend on the variance estimates and the extent to which the isotope mixture precludes the contribution sources included in the model. If, for instance, you specify source isotope values with very little or no variance, then very few of the random draws representing proportional contributions will be resampled because most draws will have very low likelihoods (note that as the variance in insotope values approaches 0, the results of MixSIR will converge to those of mixing models that rely on algebraic solutions). Similarly, the inclusion of implausible sources based on the isotope mixture and fractionation will lower the resampling rate because the model will coincidentally sample implausible (~0 likelihood) parameter space.

Incorporating prior information

The Bayesian framework allows a user to establish informative priors to guide model estimates. As stated above, the probabilities of source contributions are evaluated against prior information according to the Dirichlet distribution.

When you select the 'uniformative' radio button in the priors section of the MixSIR GUI, the model specifies Dirichlet(α_i=1) for priors on all source contributions. Thus, all possible combinations of source contributions are *a priori* equally likely.

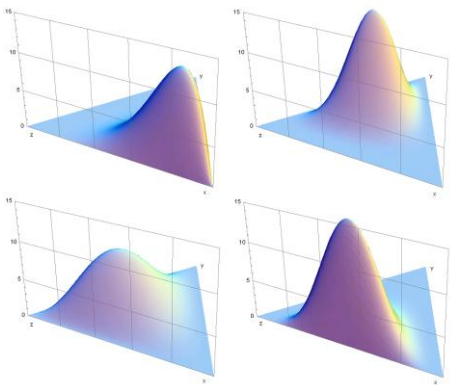

Figure 1. Several images of the probability density of the Dirichlet distribution (3 sources) for various parameter vectors α. Clockwise from top left: α=(6, 2, 2), (3, 7, 5), (6, 2, 6), (2, 3, 4) (graph from Wikipedia)

If you wish to specify informative priors by defining a Dirichlet distribution for the source contributions, you must generate a text file with α_i's as a single row. Upon selecting the radio button specifying 'Define Priors', MixSIR will check for the presence of this file, and then load these priors into memory for the model run.

How Do I Identify Appropriate Informative Priors?

We applied a bootstrap routine (Matlab) to gut content data in order to calculate priors for each of our sources. This code is freely available over http://www.ecologybox.org. This is certainly not the only way to develop appropriate priors, and we expect that others will refine/improve our methods.

MODEL RESULTS

Upon model run completion, a diagnostics window will appear that describes model results and performance. This window will give:

1) **The number of posterior draws**

    a. There should ideally be more than 1,000 posterior draws. As posterior draws increase, the histogram surface converges to the true posterior likelihood surface.

2) **the number of duplicate draws**

    a. If the posterior draws contain a maximum duplicate draw of 3-5 or more you should re-run the model with more iterations. A high degree of duplicate draws suggests the threshold value (T) is too low (too few iterations). Iterations are cheap! If the problem can't be solved with increased iterations, assess the geometry of the model you are trying to solve and the plausibility of the priors. Remember, priors with with α_i's and β_i's less than 1 will cause poor model function.

3) **the unique parameter vectors in the resample**

    a. If with very many iterations, the model re-samples the same parameter vector multiple times, the model may have implausible geometry or poorly defined priors.

4) **the maximum importance ratio (MIR)**

    a. This is calculated by determining the ratio of the maximum unnormalized posterior probability re-sample to the sum of all unnormalized posterior probability re-samples. The value should be below 0.001 (McAllister & Pikitch 1997). If it is not, you should increase the number of iterations. If, regardless of model iterations, the MIR remains above 0.001, the model may have implausible geometry or poorly defined priors.

5) **The 5^{th}, 25^{th}, 50^{th}, 75^{th}, and 95^{th} percentiles of the posterior contributions of each source**

    a. These percentile values describe the distributions associated with the proportional contribution of each source to the mixture. The 50% percentile represents the median source contribution value for each source. Note that when the posterior distribution is

multimodal , these percentiles may not adequately describe the posterior surface of the source contributions.

Output Files

If you wish to have the all the resampled draws from your model run written to files for later analysis outside of MixSIR, check the box entitled "Write results files?". Files called 'likelihoods.txt' and 'contrib_out.txt' will be written to the directory containing the MixSIR.exe program.

1) **'likelihood.txt'** contains a single column of unnormalized posterior probability values for the resampled draws resulting from your model run.

2) **'contrib_out.txt'** contains a matrix of the source contributions, such that each row contains the contribution draws associated with the unnormalized posterior probability in the same row of 'likelihood.txt'. The order of the source contributions columns will match the order of the sources used in your input files. For instance, if you used sources [a; b; c; d] for your rows in the input file 'mean_source.txt', then column 1 of 'contrib_out.txt' would give the contributions for source a, column 2 would give the contributions of source b, etc.

Graphs

Both graphs generated by MixSIR are shown in the program window by default. However, by checking the box titled "Open for editing / printing / saving", a resizable and editable version of each graph will pop up in a separate window. This window also allows the graph to be edited and saved in several different formats.

1) **Likelihood distribution graph**

    a. This graph presents a histogram of the re-sampled unnormalized posterior probability values relative to the single largest value in the set of posterior draws. If the SIR function places too little weight in the tails of the posterior distribution, it may be inefficient in approximating posterior distributions. Thus, the graph should not be heavily right skewed (the majority of the posterior draws are at or very near the 'best' draw based on the posterior likelihoods; McAllister & Ianelli 1997).

2) **Histogram of posterior source contributions**

a. This graph presents the distribution of posterior proportional contributions of each source to the mixture. The order of the source-specific histograms will match the order of the sources used in your input files. For instance, if you used sources [a; b; c; d] for your rows in the input file 'mean_source.txt', then the top histogram would give posterior distribution for source a, the 2nd from the top histogram would give the posterior distribution for source b, etc. The more posterior draws there are, the smoother this histogram surfaces will be. When the box "X axes over [0,1] interval?" is checked, the histogram bins are evenly distributed between 0 and 1. Without this box checked, the program will automatically generate a bin range appropriate for each of the individual posterior source distributions. The Y axes on the histogram graphs give the relative probability that each bin represents the true source contribution, such that the sum of all bin probabilities equals 1.

CITATIONS

Ellison, A.M. (2004). Bayesian inference in ecology. *Ecol. Lett.*, 7, 509-520.

Gelman, A., Carlin, J.B, Sterm, H.S. & Rubin, D.B. (2003). *Bayesian data analysis*. CRC Press, Boca Raton, FL.

Hilborn, R. & Mangel, M. (1997). *The Ecological Detective.* Princeton University Press, Princeton, NJ.

McAllister, M.K. & Ianelli, J.N. (1997). Bayesian stock assessment using catch-age data and the sampling-importance resampling algorithm. *Can. J. Fish. Aquat. Sci.*, 54, 284-300.

McAllister, M.K. & Pikitch, E.K. (1997). A Bayesian approach to choosing a design for surveying fishery resources: application to the eastern Bering Sea trawl survey, *Can. J. Fish. Aquat. Sci.* 54, 301–311.

McAllister, M.K., Pikitch, E.K., Punt, A.E. and Hilborn, R. (1994). A Bayesian approach to stock assessment and harvest decisions using the sampling / importance resampling algorithm. Can. J. Fish. Aquat. Sci. 51, 2673-2687.

Rubin, D.B. (1987). Comment on 'The calculation of posterior distributions by data augmentation. *J. Am. Stat. Assoc.*, 82, 543-546.

Rubin, D.B. (1988). Using the SIR algorithm to simulate posterior distributions. In: *Bayesian Statistics 3: Proceedings of the Third Valencia International Meeting, June 1-5, 1987* (eds Bernardo, J.M., Degroot, M.H., Lindley, D.V. & Smith A.M.). Clarendon Press, Oxford, pp. 385-402.