SIAR V4
(Stable Isotope Analysis in R)

An Ecologist's Guide

Richard Inger, Andrew Jackson, Andrew Parnell & Stuart Bearhop

1. Introduction

SIAR is a package designed to solve mixing models for stable isotopic data within a Bayesian framework. This guide is designed to get researchers up and running with the package as quickly as possible. No expertise is required in the use of R.

you can install the SIAR package by typing command;

```
install.packages("siar")
```

4.3.1. *Structure of the input files.*

SIAR requires 3 input files. All files should be tab separated .txt files. These can be generated in MS Excel or in any basic text editor.

1) The consumer data file contains the stable isotopic data for you consumers and should consist of columns, firstly the grouping variable, secondly the data for isotope 1, thirdly the data for isotope 2. More or fewer isotopes can be used.

Example consumer data file:

Code	d15N	d13C
1	9.1	-10.48
1	10.16	-11.78
1	8.78	-11.41
2	9.42	-10.47
2	9.26	-10.58

2) The sources data file. This file contains the data on the different sources contributing to the consumer. The data can be input as mean and standard deviation for each source. N.B. the order of occurrence of each isotope must match that in the consumer data file.

Example source data file:

Source	Meand15N	SDd15N	Meand13C	SDd13C
Zostera	6.49	1.21	-11.17	1.21
Grass	4.43	0.64	-30.88	0.64
U.lactuca	11.19	1.96	-11.17	1.96
Enteromorpha	9.82	1.17	-14.06	1.17

3) The Trophic Enrichment Factor (TEF) data file. This contains the TEFs as mean and standard deviation for each source. TEFs can also be refereed to as fractionation / discrimination factors within the literature. Again, the order of occurrence of each isotope must match that in the consumer data file.

Example TEF data file:

Source	Meand15N	SDd15N	Meand13C	SDd13C
Zostera	3.54	0.74	1.63	0.63
Grass	3.54	0.74	1.63	0.63
U.lactuca	3.54	0.74	1.63	0.63
Enteromorpha	3.54	0.74	1.63	0.63

Note Different TEFs can be applied to different sources.

Note Ensure that there are no spaces in the header columns(as R will interpret the white spaces as a new entry and it will think there are more columns than you do), and that the order of the isotopes are the same for all the files.

4.3.2. *Data Input*

Once your data is in the correct format you'll need to read it into R. So, the first few lines of you script should look something like this:

```
data<-read.table('ConsumerData.txt',header=TRUE)

sources<-read.table('SourceData.txt',header=TRUE)

 tef<-read.table('TEFData.txt',header=TRUE)
```

This uses the read.table command to read the data from the .txt file and store it as a matrix in R. So, if you run this script, then type 'data' in the command line window of R the consumer data will be displayed.
Note that all the files need to be in the same folder as you specified using the setwd() function unless you specify the full path to that folder. Note that R uses "/" to code folder path structures, whereas windows uses "\".

4.3.3 *Running the model*

The next step is to actually run the model. This is achieved by the command:

```
model1<-siarmcmcdirichletv4(data,sources,tef,concdep=0,500000,50000)
```

This command takes the 3 matrices we created earlier and uses them as inputs for the model. It also tells SIAR that we have no data on concentration dependence (concdep=0), the final two numbers are the number of iterations for the model to run, and the 'burnin'

(the number of initial iterations to discard). These setting should be OK for most analyses. Finally it tells SIAR to put the results into an R object that we've called 'model1'.

As with any function in R, there are often many customizable options available to you. The help files provided for functions should tell you what you need to know. To access these for this particular function simply type:

```
?siarmcmcdirichletv4
```

The model will take between a few seconds to a few minutes to run depending on the number of sources, isotopes and groups and the speed of the computer your using. The progress of the model is displayed on the command line window, which shows the number of iterations, and when a particular group is finished. The command line window doesn't always update that often, this is just a feature of R, but you can update it by clicking on the screen or the down arrow of the scroll bar.

Once the model is finished you'll see something like:

```
Job completed successfully. Duration: 3.6 seconds.
```

The duration refers to the last groups, not the whole model.
So, that's the model run complete, now to have a look at some results.

4.3.4. *Plotting the results.*

There are a number of plotting options available within the SIAR package. 4.3.4.1. *Plotting the raw data*

Firstly lets have a look at the raw isotopic data. This can be done using:

```
siarplotdata(model1)
```

This will produce a biplot with the isotope that is in the first column on the x-axis, and the isotope in the second column on the y-axis.

4.3.4.2. *Diagnostic Matrix Plot*

Next, you can produce a matrix plot, which shows histograms for the estimate of each proportion on the main diagonal. The contour plots in the upper triangle shows how pairs (by rows and columns) of posterior distributions are correlated (if at all) and the actual correlation coefficient is given in the lower triangle. These correlation plots and statistics are generated by pairing simulated values of the dietary proportions drawn by each iteration of the MCMC process (so for each draw they must sum to one). This is a useful diagnostic tool as it will identify when the model is performing well, indicated by low correlations between sources, or when the model is struggling to differentiate between sources, as indicated by higher correlations. For instance, if two sources are very close together, then likely solutions could involve one or other of the sources but not both at the

same time. In such a scenario, these two sources would be expected to show negative correlation in their posterior distributions.

and provide a sensible solution to the mixing problem. See section 4.3.5. for details on how to access the full set of posterior draws should you wish to perform additional analyses.

To produce the plot:

```
siarmatrixplot(model1)
```

Once the command is executed you will be prompted to enter a group number, after which the plot will be created.

4.3.4.3. *Histogram Plot*

SIAR can produce histograms of the distribution of possible solutions for all sources in the model. These plots will be familiar to anyone's who's used ISOSOURCE. The difference here is that SIAR produces true probability density functions. To produce histograms:

```
siarhistograms(model1)
```

You will then be prompted to select the group you which to plot, and also if you require all the sources separately or on one plot.

SIAR can also produce boxplots of the proportions of different sources. To compare the

proportions of each source for a group:

```
siarproportionbygroupplot(model1)
```

At which point you will be asked to pick which source you want to plot

There are additional arguments that can be included in this command.
prn If prn=TRUE then the values for the probability densities are returned to the

command line.
grp Specifies the group to be plotted

probs Defines the credibility intervals to be plotted. The default is 95, 75 and 25%. Other values should be added in the format c(10, 50, 90).

type Defines the type of plot. The Default is boxes, type="lines" produces a plot using different thickness lines.

clr Sets the colour of the boxes. Default is greyscale.
scl Sets the width of the boxes or lines. Default is 1.

xspc Sets the amount of blank space before the first and after the last box / line.
leg Allows a legend to be added to the plot if leg=TRUE. Only works with line plots

So:

```
siarproportionbysourceplot(model1, prn=TRUE,grp=1,probs=c(5,25,75,95))
```

Will return the probability densities to the command line, produce a plot for group 1, and plot the 5, 25, 75 and 95% credibility internals.

SIAR also allows you to plot the proportions of each group by source, allowing a comparison

of sources for different groups:

```
siarproportionbygroupplot(model1)
```

Which will ask you for which group you want to plot.

4.3.5. ***Accessing the Raw Output Data.***
The raw output from the model is stored in a matrix called output, which is in the R object

created earlier called `model1`. So we can create a new matrix with this data in: `out<-model1$output`

Access entire document at:
http://sethnewsome.org/sethnewsome/EE_files/SIAR%20for%20Ecologists.pdf