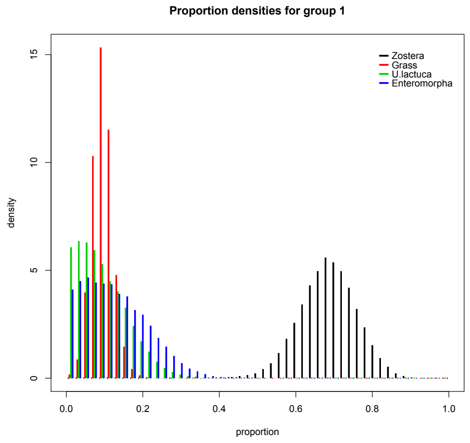
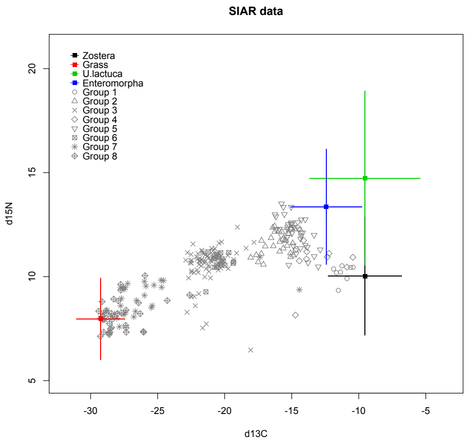
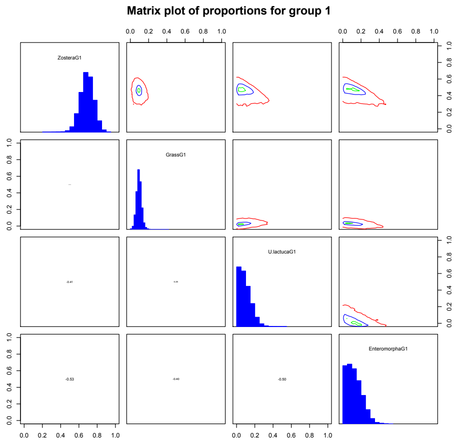
SIAR V4
(Stable Isotope Analysis in R)

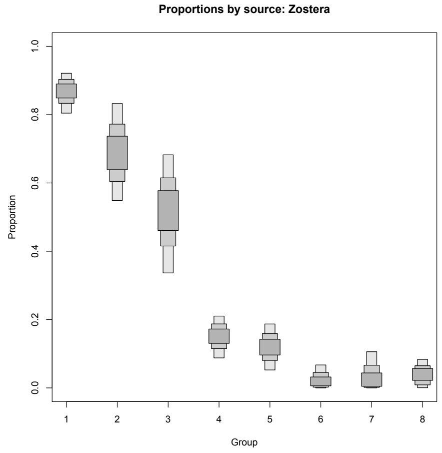
An Ecologist's Guide

Richard Inger, Andrew Jackson, Andrew Parnell & Stuart Bearhop









Contents Page

1. Introduction 3

2. Installation of R 3

3. Installation of SIAR 3

4. Working with SIAR 5

4.1. Menu System or Scripts 5

4.2. Data Structure 5

4.3. Multiple Data Points 6

4.3.1. Structure of the Input Files 6

4.3.2. Data Input 7

4.3.3. Running the Model 7

4.3.4. Plotting the Results 8

4.3.4.1. Plotting the Raw Data 8

4.3.4.2. Diagnostic matrix plot 8

4.3.4.3. Histogram Plot 9

4.3.4.4. Boxplot of Proportions by Group 10

4.3.4.5. Boxplot of Proportions by Source 11

4.3.5. Accessing the Raw Output Data 11

4.4. Single Data Points 12

4.5. Concentration Dependence Models 12

4.6 Incorporating Prior Information into Models 12

4.7. Troubleshooting 13

5. Recommended Reading 14

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.

We assume that you have a sound working knowledge of stable isotopic mixing models, and the assumptions and potential pitfalls associated with these models. Failure to take account of these assumptions could lead to erroneous results. A list of required reading is presented in Appendix A of this guide. We strongly recommend that all users unfamiliar with isotopic mixing models read these papers.

SIAR will always try to fit a model even if the data are nonsensical.

Further information on the SIAR package is available from:
http://cran.r-project.org/web/packages/siar/siar.pdf

The examples shown here are based on the dataset presented in Inger et al. 2006. A subset of this dataset is provided with the SIAR package

2. Installation of R

R is available for virtually all operating systems, including Microsoft Windows, Mac OS, Linux and a wide variety of UNIX platforms. R can be downloaded from http://www.r-project.org/ and installed by following the instructions on the R website.
From the home page click the **download R** link.
Then select the CRAN mirror nearest to you.
Then select your operating system.

Windows - select the **base** ‘Binaries for base distribution’ option. Then click on the **Download R** link. An executable file will then be downloaded. Double click on this to run the installation program and follow the on screen instructions.

Max OS - select the **R-2.11.0.pkg** link, then click on the downloaded package, which will run the installer, then follow the instructions.

Linux / UNIX – if your running either of these we assume you can do this!

3. Installation of the SIAR package

Windows

Start R and select packages from the menu, then install package(s)… Then select your nearest CRAN mirror from the list, and finally scroll down to find ‘siar’ on the list and click on it. You will then see lots of activity on the screen as the SIAR package and the other packages it uses are downloaded. The final line should then read:

```
package 'siar' successfully unpacked and MD5 sums checked
```

You then need to load the package. Type:

```
library(siar)
```

This will load the package and all the associated packages. You'll need to do this every time you start R, or you can make R install packages automatically on start up (see R help pages).

Alternatively, you can install the SIAR package by typing command;

```
install.packages('siar')
```

MacOS

Start R and select 'Packages and Data' then package installer at which point a new window opens. Type 'siar' into the search box, and click on 'get list', then ensure that the 'install dependencies' box is ticked, and click on siar, then click 'install selected'. You'll then see lots of on screen activity as the packages are downloaded. The final message should begin with:

```
The downloaded packages are in /var/folders . . . .
```

You then need to load the package. Type:

```
 library(siar)
```

This will load the package and all the associated packages. You'll need to do this every time you start R, or you can make R install packages automatically on start up (see R help pages).

4. Working with SIAR

Before getting started there are a couple of points to consider.

4.1. *Menu System or Scripts*

There are two ways to run the SIAR package. Firstly via the built in menu system, or secondly, via scripts. We strongly recommend using the latter option as the menu system was only ever designed as an introduction to the package. Accessing the package by running scripts offers a much greater level of flexibility and allows those with some coding experience to develop scripts to run multiple analyses or carry out post-hoc tests on the outputs from SIAR. Scripts can be written using the built in script editor or via external editors. In our experience the built in Mac editor is quite good, although the windows version is somewhat limited, and we would recommend using Tinn-R, which is available as a free download. Once your script has been written, save it and then either send it to R from the script editor, or call it from R using the 'source' command.

4.2. *Data Structure.*

Are you interested in determining the source proportions based on single or multiple data points? In most analyses you will most likely require some sort of grouping variable. The different groups could represent for example:

- Different demographic groups.
- Different sampling periods.
- Different sampling locations.
- Different samples from the same individual.

In the case of the example data that comes with the SIAR package the grouping variable is the month of capture/sampling. See Inger et al. 2006 for further details

This is the default mode for SIAR. Currently up to 30 groups can be used in a single analysis. You must include more than one observation per group, and really there should be at least 3 but preferably 5 or more per group in order to be able to estimate intra-group variances.

If you only have one data point for each individual / observational group then you should use the 'siarsolo' command. This works in exactly the same way as the default, but does not include the residual error term (See Parnell et al. 2010 for further details).

4.3. Multiple Data points

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.

There is an additional file that can also be incorporated into the model with the elemental concentration values. See Section 4.5.

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.

Note To run concentration dependant models see section 4.5.

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.***

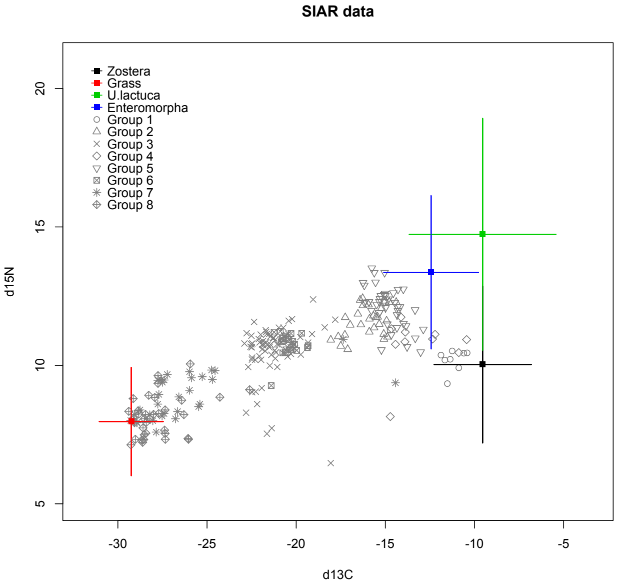
There are a number of plotting options available within the SIAR package.

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. Once the command is executed a crosshair will appear in the cursor position, click on the plot to position the legend.
This will produce something like this:



Note The TEFs are added to the sources rather than subtracted from the consumers. This is to allow different TEFs to be assigned to different sources.

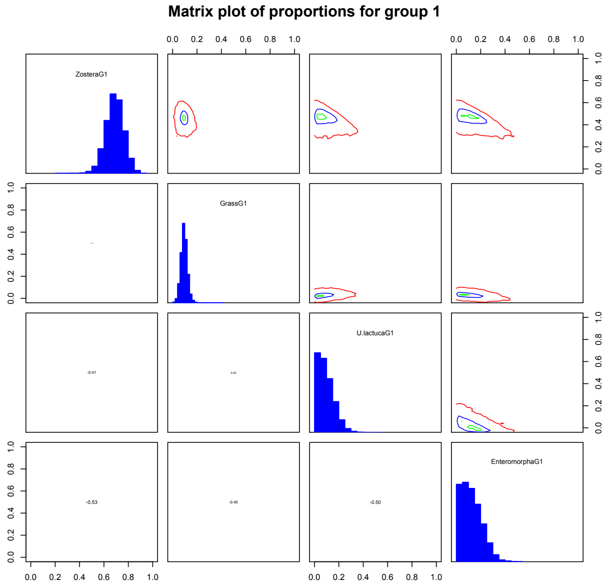
4.3.4.2. *Diagnostic Matrix Plot*

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. This is evident in the isotope biplot for Enteromorpha and U. lactuca, which then show negative correlation in the contour plot of their joint posterior distribution for Group 1. Positive correlations are also possible, where a solution that involves one source, may necessissarily require another source in some proportion so as to balance each other out 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.

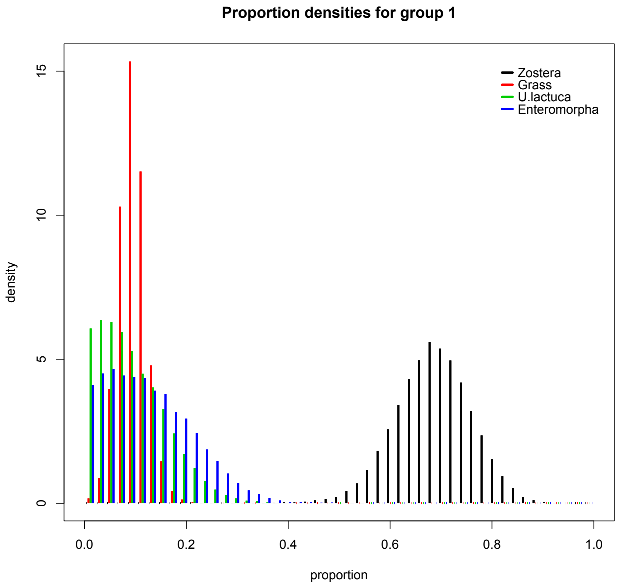


4.3.4.3. *Histogram Plot*

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. If you select the latter option you will get a plot like this:

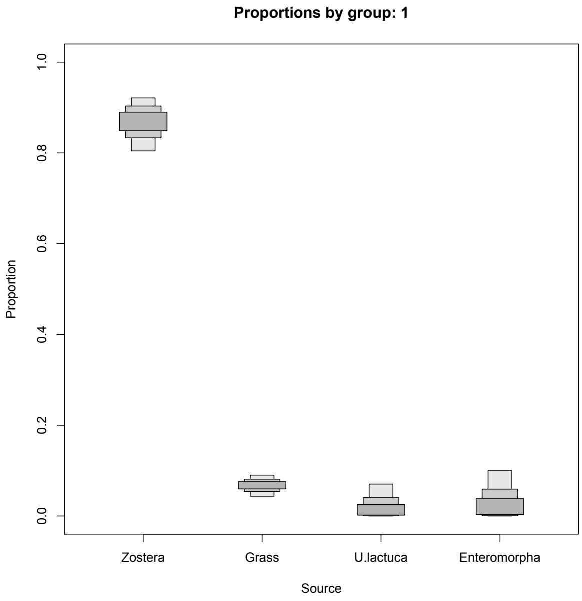


4.3.4.4. *Boxplot of Proportions by Group*

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, and SIAR will then produce something like this:



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))
```

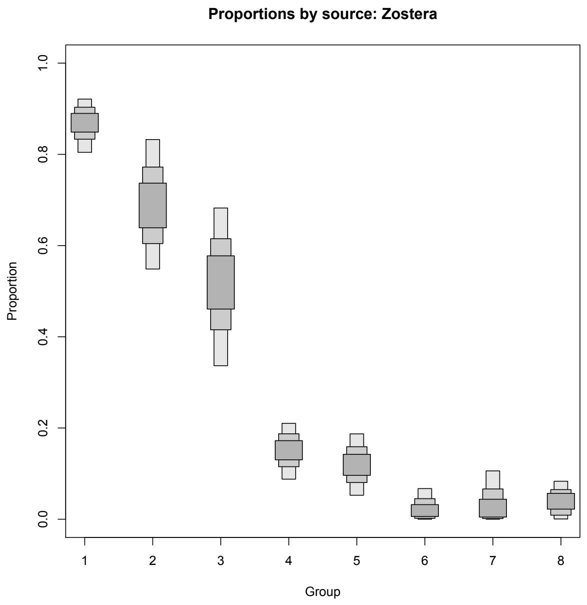
Will return the probability densities to the command line, produce a plot for group 1, and plot the 5, 25, 75 and 95% credibility internals.

4.3.4.5. *Boxplot of Proportions by Source*

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. SIAR will then produce a plot like this:



The additional arguments are the same as for the `siarproportionbysourceplot` command.

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
```

This can then be examined using the fix command.

```
fix(out)
```

This will start up a data editor containing the raw data. The output is separated into source by groups. Such that column 1 is source 1 of group 1. After all the sources for a particular group have been represented there are a further two columns containing data on the residual error from the model. For example the column SD1G1 is the residual error associated with isotope 1 for consumers in groups 1. This value tells you how variable the consumers are within a group after fitting the model.

Obtaining extra data from the raw data is easy. For example if you want the mean value for group1, source 1 enter:

```
mean(out[,1])
```

Square brackets are used to define subscripts in R, so [1,1] would return the values in the first row and first column. [,1] selects all of the first column.

4.4. Single Data points

If you only have single data points (or sets of data points for multiple isotopes) then almost everything is the same as for the multiple data points approach.
You data should be set up such that each data point is coded 1 to *n:*

Code	d15N	d13C
1	9.1	-10.48
2	10.16	-11.78
3	8.78	-11.41
4	9.42	-10.47
5	9.26	-10.58

Then instead of using the `siarmcmcdirichletv4` command use the `siarsolomcmcv4` command:

```
model1<-siarsolomcmcv4(data,sources,tef,concdep=0,500000,50000)
```

4.5. Concentration Dependence Models

Elemental concentrations can also be incorporated into SIAR models. To do this you'll need an extra file formatted like this:

Means	meand15N	sdd15N	meand13C	sdd13C
Zostera	2.97	0.15	35.93	0.29
Grass	3.55	0.23	40.26	0.42
U.lactuca	1.92	0.17	20.98	0.17
Enteromorpha	1.39	0.34	18.44	0.15

This will then need to be read into R as before.

```
conc<-read.table('ConcData.txt',header=TRUE)
```

Then run the model.

```
model1<-siarmcmcdirichletv4(data,sources,tef,conc,500000,50000)
```

4.6. Incorporating Prior Information into Models

If external information is available to guide the model in the likely range of values for the dietary proportions, it can be used as a prior distribution in the SIAR modelling framework. External information may arise from previous runs of the SIAR model with different data sets, sources (such as papers) which provide likely ranges, or expert opinion about the dietary proportions. By default, SIAR assumes that there is little (also known as vague) prior information to guide the dietary proportions. The function siarelicit can be used to input a prior distribution in SIAR, by expressing the estimated mean of each proportion and a standard deviation for the first given proportion. This restriction (giving only one standard deviation) is required because of the Dirichlet prior distribution used by SIAR to model the dietary proportions. The function siarelicit can be called after a standard run of SIAR (ie with a non-informative prior). The command:

```
siarelicit(model1)
```

will bring up a set of instructions for entering prior data. Suppose we know that the 4 dietary proportion means should be in the range of 0.7, 0.15, 0.08, 0.07 (note that they must sum to 1), and that the first should be in the range from 0.65 to 0.75. (ie 95% CI width of 0.05), and therefore standard deviation width of approximately 0.025. We can then enter these proportions and standard deviations as instructed. siarelicit will return prior values for the new run. Here, these are: 234.5, 50.25, 26.8, 23.45. We use these in a new run of SIAR:

```
model2<-siarmcmcdirichletv4(data,sources,tef,conc,prior=c(234.5,50.25,26.8,23.45))
```

A matrix of histogram plot will now show vastly reduced uncertainties in the dietary proportions.

4.7. Troubleshooting

R crashes for no apparent reason shortly after a model run has been started.

This is likely due to a lack of available memory. When this happens it may be the result of a number of factors, including how much RAM your computer has, how much of this is available to R, the number of isotopes, sources, and groups in the model. For most computers this is not a problem for less for 10 groups. It may become problematic when using siarsolo to compare multiple individuals. SIAR will, if enough memory is available allow a maximum of 30 groups. If this is a problem for your analysis try to reduce the number of groups in a single model run and then run the remaining groups separately– running one big model with all groups together is identical (in SIAR's current form) to running separate models per group.

SIAR outputs are unexpected.

This problem can often occur if the order of the isotopes is different in different files. The order of the isotopes must be the same for all input files.

5. Recommended Reading

Phillips, D.L. (2001) Mixing models in analyses of diet using multiple stable isotopes: a critique. Oecologia, 127, 166-170

Phillips, D.L. & Gregg, J.W. (2001) Uncertainty in source partitioning using stable isotopes. Oecologia, 127, 171-179

Phillips ,D.L. & Koch, P.L. (2002) Incorporating concentration dependence in stable isotope mixing models. Oecologia, 130, 114-125

Phillips, D.L. & Gregg, J.W. (2003) Source partitioning using stable isotopes: coping with too many sources. Oecologia, 136, 261-269

Phillips, D.L., Newsome, S.D. & Gregg, J.W. (2005) Combining Sources in stable isotope mixing models: alternative methods. Oecologia, 144, 520-527

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

Jackson, A, Inger, R., Bearhop, S. & Parnell, A. (2009) Erroneous behaviour of MixSIR, a recently published Bayesian isotope mixing model: a discussion of Moore & Semmens, Ecology Letters, 2008. Ecology Letters, 12, E1–E5.

Parnell, A., Inger, R., Bearhop, S. & Jackson, A.L. (2010) Source partitioning using stable isotopes: coping with too much variation. PlosOne, 5(3): e9672.

Inger, R., Ruxton, G.D., Newton, J., Colhoun, K., Robinson, J.A., Jackson, A.L., & Bearhop, S. (2006) Temporal and intrapopulation variation in prey choice of wintering geese determined by stable isotope analysis. Journal of Animal Ecology, 75, 1190-1200.