Tarroux, A., Ehrich, D., Lecomte, N., Jardine, T., Bêty, J. & Berteaux, D. (2010) Sensitivity of stable isotope mixing models to variation in isotopic ratios: evaluating consequences of lipid extraction. *Methods in Ecology and Evolution*, (*accepted 13-04-2010*)

`Fatsim`: an R-script to simulate the effect of extracting/not extracting lipids on the outcome of stable isotope mixing models

Lipids are more depleted in ^{13}C than other animal tissues due to the metabolic pathways of their synthesis (DeNiro & Epstein, 1977). Samples with a high content of lipids will thus have lower δ^{13}C values than leaner samples. When using stable isotope analysis and mixing models to reconstitute the diet of a consumer, such a bias in δ^{13}C values for one or several sources and/or for the consumer may affect the outcome of mixing models. Tarroux *et al.* (2010) suggested a simulation approach to evaluate the potential effect of lipid extraction on the outcome of mixing models for any particular data set, an approach which has been implemented in the present R-script, `Fatsim`.

The function `Fatsim` allows exploring the effect of lipid extraction or the effect of different levels of lipid content on the outcome of mixing models with two isotopes (carbon and nitrogen). Stable isotope ratios (δ^{13}C and δ^{15}N) for one or several consumer individuals and several sources are needed. Lipid extraction is simulated by gradually increasing the δ^{13}C value of one or several sources and/or of the consumer and recalculating a mixing model solution for each step. In a similar way, the effect of having samples with a high lipid content is simulated by decreasing the δ^{13}C value of the potentially affected data. As in some cases it has been shown that chemical lipid extraction also alters δ^{15}N values (Logan & Lutcavage, 2008), it is possible to simulate a source-specific constant change in δ^{15}N in addition to changes in δ^{13}C. In practice, the user defines a maximum shift in δ^{13}C to consider for each source and for the consumer. If it is certain that lipids are not a problem for a certain source, or for the consumer, this maximum can be set to 0 (e.g. in the case of a purely proteinaceous tissue). The δ^{13}C values for all sources and/or the consumer are then modified stepwise, with user-defined increments (e.g. 0.5‰), and a mixing model solution is estimated for all combinations of modified values. The results of the function `Fatsim` are stored in an R object and a summary is written to a text file. For each combination of shifts in isotopic ratios, the function compares the proportions of each source in the diet inferred from the modified data to the proportion inferred from the original, unmodified data. The maximum difference in proportions is reported in the output file. If this difference exceeds a user defined threshold, it is marked as “DIFFERENT”. In addition, a credibility interval for each proportion is listed.

Mixing model solutions are estimated with the R-package `siar` (Parnell et al., 2008; Parnell et al., 2010). This package implements a Bayesian approach to solve mixing problems. For a group of consumers the function `siarmcmcdirichletv4` of the `siar` package is used, and for single consumer individuals the function `siarsolomcmcv4`. The results of these functions are posterior probability distributions of the proportions of each source in the mixture. Point estimates for the source proportions are obtained by taking the mean, median or mode of the posterior distributions. In addition, credibility intervals for the proportions can be estimated. Isotope mixing models cannot be solved if the values of the consumer lay outside the polygon defined by the sources in the two-dimensional isotope space, taking standard deviation into account (Phillips, 2001). For combinations of shifts in isotopic ratios which result in such situations, the functions of `siar` cannot estimate posterior probabilities properly. These combinations of modified values are marked by “problem with the data!” in the output of `Fatsim`.

In addition to the main simulation function `Fatsim`, the script contains two plotting functions. `Fatsim.datplot` produces two scatterplots (δ^{13}C x δ^{15}N) side by side: original data and data resulting from any specific combination of shifts in isotopic ratios (defined by the user).

`Fatsim.resplot` plots the mixing model results of the original, unmodified data and of a combination of any specific combination of shifts in isotopic ratios, side by side. Both functions are based on the plotting functions of `siar`.

How to use Fatsim

(these instructions are intended for inexperienced R users. See below for a shorter version)

1. Using R

   If you have not used R before, the first step is to download and install R from: http://www.r-project.org/

   R is a free software environment for statistical computing and graphics. Numerous additional packages can be downloaded for different specialized analyses, making R a powerful tool for data analysis notably in ecology. It is definitely worth discovering… R is a software based on writing commands. This may seem difficult to beginners, therefore in this manual all steps necessary to run `Fatsim` will be described in detail.

2. `siar`

   `siar` (Parnell et al., 2008; Parnell et al., 2010) is a package which needs to be installed additionally, once R has been installed. To do so, run R and select Packages -> Install Package(s) in the main menu. Then choose a mirror close to you, select the package `siar` in the list of available packages, and click OK. This needs to be done only the first time you want to use the package in order to install it. After having installed the package, you have to load it into R: choose Package -> load package in the main menu, and choose `siar` from the list and click OK. The package needs to be loaded into R **each time** you start the program and want to use `siar`**.**

3. `Fatsim`

   The `Fatsim.r` script can be loaded into R through the R menu. Choose File -> Source R code… in the menu and browse to the folder where you have the script, then click open to read it into R.

4. Data

   `Fatsim` contains an example of data for a consumer and four sources. The consumer data are named `predator` and the sources data `sour`, and they are defined as “objects” in the R language. These data can be used to test `Fatsim`. You can look at these data by typing the name of the R object directly in the main R window, for example

   ```
   predator
   ```

   An easy way to enter your own data in order to use them with `Fatsim` (or with `siar`) is to start by entering them into a spreadsheet such as Excel.

For the consumer data, enter two columns, the first with the δ^{13}C values for each individual and the second with the δ^{15}N values for each individual. Put column names in the first line. For example:

d13C	d15N
-20.8	10
-19.9	10.2
-19.6	10.1
-20.2	9.9
-20.1	9.8

If you have only one consumer individual, enter only one row of data. `Fatsim` will automatically detect if you have one or several consumer individuals, and then use the appropriate `siar` function. Example with one consumer individual:

d13C	d15N
-19.67	10.5

Save this file as a tab delimited text file (e.g. `consumer.txt`) using "Save as" in Excel and setting "text file (Tab delimited)(*.txt)" in the field "Save as type:".

For each source, an average and a standard deviation of the signatures for each isotope have to be entered as follows: The δ^{13}C values should be in the second and third column and δ^{15}N values in the fourth and fifth. Source names should be in the first column and column names in the first row. For example:

source	d13Cmean	d13Csd	d15Nmean	d15Nsd
bread	-21	0.5	11	0.5
cheese	-15	0.5	15	0.5
chocolate	-25	0.5	5	0.5
potato	-19	0.5	9	0.5

Column names can be modified. This file should also be saved as a tab delimited text file (e.g. `sources.txt`).

If your data are not corrected for diet-tissue discrimination, these values may optionally be entered for each isotope and each source in a similar format as the source data. For example:

source	d13Cmean	d13Csd	d15Nmean	d15Nsd
bread	0.6	0.5	2.6	0.5
cheese	0.6	0.5	2.6	0.5
chocolate	0.6	0.5	2.6	0.5
potato	0.6	0.5	2.6	0.5

Column names can be modified. This file should also be saved as a tab delimited text file (e.g. `fractionation.txt`).

Place all these text files in the same folder.

In R, move to the folder with your files choosing File -> Change dir… in the main menu, and browse to the folder, where your files are.

Read your data into R from the text files using the function `read.delim`. For example:

```
mypred <- read.delim("consumer.txt")
mysources <- read.delim("sources.txt")
```

The sign <- in R means that you assign a name to data. The name assigned to the data is chosen by the user.

Check if the data are entered correctly by typing the name of your R object in the main R window, for example

```
mysources
```

The data for the sources should then appear into the R console.

5. Simulations

The function `Fatsim` starts simulations with isotopic data for a consumer and several sources. The $\delta^{13}C$ for each source will be shifted incrementally up to a maximum level. These maximum amounts are entered as the argument `maxmod`. In R you can combine a series of numbers into a vector using the function `c` as follows:

```
mymax <- c(5, 0, 3, 0)
```

In this case there are four sources. The first source will be modified up to an increase of 5‰ in $\delta^{13}C$, the second source will not be modified, etc. In R, arguments of a function are entered in parenthesis after the name of the function. Different arguments are separated by commas. Three arguments need to be entered for the function `Fatsim`: the name of the R object containing the consumer data, the name of the R object with the source data and `maxmod`. For example:

```
simultest <- Fatsim(pred = mypred, sources = mysources,
maxmod = c(5, 0, 3, 0))
```

or

```
simultest <- Fatsim(pred = mypred, sources = mysources,
maxmod = mymax)
```

The function has several optional arguments, for which default values will be used, if nothing is entered by the user. You can modify them by entering them following the three first ones and specifying values for them. Additional arguments for the function `Fatsim` are:

`consumer` which indicates the maximum change in δ^{13}C to simulate for the consumer. The default for this argument is 0, implying that you don't simulate any effect of lipid extraction for the consumer. If you want to simulate the effect of lipid extraction only for the consumer and not for any of the sources, enter a maximum change for the consumer and 0 maximum changes for all sources. For example:

```
simultest <- Fatsim(pred = mypred, sources = mysources,
maxmod = c(0, 0, 0, 0), consumer = 5)
```

`incrC` indicates the size of the steps by which the δ^{13}C values are modified. The default for this argument is 0.5‰ (meaning that the function will modify the δ^{13}C values by steps of 0.5‰). The size of the steps is identical for all sources and for the consumer.

`correc` allows the user to enter a matrix with diet-tissue discrimination values as shown in point 4. The default for this argument is 0, implying that your data are already corrected for discrimination.

`conc` allows the user to enter a matrix with concentration dependencies, as it can be used in `siar`. Consult `siar` and Phillips and Koch (2002) for more details. The default for this argument is 0, implying that there is no concentration dependency.

`incrN` allows the user to simulate a modification in δ^{15}N values as a result of lipid extraction for the sources. The shift in δ^{15}N values is a constant for each source and the values are entered as a vector similar to the argument `maxmod`. For example:

```
simultest <- Fatsim(pred = mypred, sources = mysources,
maxmod = c(5, 5, 0, 0), incrN = c(1, 0.7, 0, 1))
```

The default for this argument is 0, implying that you don't simulate any change in δ^{15}N values for the sources.

`incrNcons` allows the user to simulate a modification in δ^{15}N values as a result of lipid extraction for the consumer. The default for this argument is 0, implying that you don't simulate any change in δ^{15}N values for the consumer.

`shift` defines the threshold values above which the resulting proportions of each source in the mixture are considered different from the proportions resulting from the original data. The default for this argument is 0.05, implying that a difference of more than 0.1 in the proportion of one or several sources will be marked with "DIFFERENT" in the output file of `Fatsim`.

`cct` defines which summary statistic of the posterior distribution will be used to compare the results based on original data to those based on modified data. The default is `cct = "mode"`, in this case the modes of the distributions estimated by using a kernel function are used (see `siar` documentation for the function `siarhdrs` for more details). Other options are `cct = "mean"` and `cct = "median"`.

`int` defines the width of the posterior credibility interval to be reported in the output file in percent. The default for this argument is 95.

`burn` defines the length of the burn-in used for the MCMC estimation (see siar documentation for details). The default is 50000.

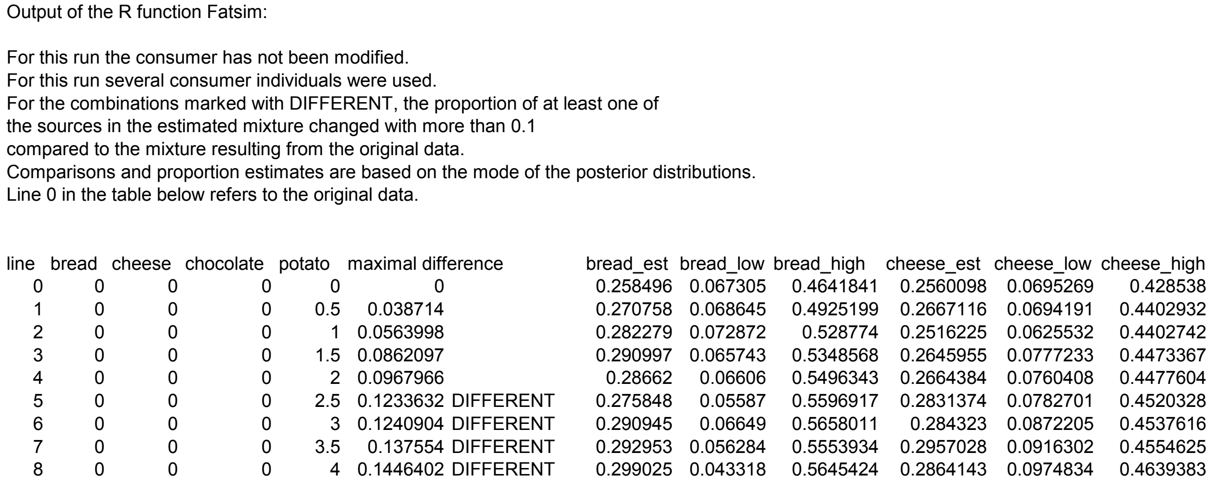
`iter` defines the total number f itnerations used for the MCMC estimation (see siar documentation for details). The default is 200000.

`thin` defines the interval between iterations to save during the MCMC estimation (see siar documentation for details). The default is 200000. `iter` – `burn` / `thin` has to be an integer.

It is not necessary to write the arguments for which you want to use the default values (see examples above).

6. Results

The results of the function `Fatsim` are stored in an R object (`simultest` in the examples above) and a summary is written to a text file called `fatsim-output.txt.` This file will be in the folder where R was carrying out the simulations and which you chose with File -> Change dir… in the R menu (point 4). It shows a table appearing as follows:



```
Output of the R function Fatsim:

For this run the consumer has not been modified.
For this run several consumer individuals were used.
For the combinations marked with DIFFERENT, the proportion of at least one of
the sources in the estimated mixture changed with more than 0.1
compared to the mixture resulting from the original data.
Comparisons and proportion estimates are based on the mode of the posterior distributions.
Line 0 in the table below refers to the original data.


line  bread  cheese  chocolate  potato  maximal difference          bread_est  bread_low  bread_high   cheese_est  cheese_low  cheese_high
   0      0       0          0       0              0               0.258496   0.067305   0.4641841   0.2560098   0.0695269     0.428538
   1      0       0          0     0.5       0.038714               0.270758   0.068645   0.4925199   0.2667116   0.0694191    0.4402932
   2      0       0          0       1      0.0563998               0.282279   0.072872    0.528774   0.2516225   0.0625532    0.4402742
   3      0       0          0     1.5      0.0862097               0.290997   0.065743   0.5348568   0.2645955   0.0777233    0.4473367
   4      0       0          0       2      0.0967966                0.28662    0.06606   0.5496343   0.2664384   0.0760408    0.4477604
   5      0       0          0     2.5      0.1233632 DIFFERENT     0.275848    0.05587   0.5596917   0.2831374   0.0782701    0.4520328
   6      0       0          0       3      0.1240904 DIFFERENT     0.290945    0.06649   0.5658011    0.284323   0.0872205    0.4537616
   7      0       0          0     3.5       0.137554 DIFFERENT     0.292953   0.056284   0.5553934   0.2957028   0.0916302    0.4554625
   8      0       0          0       4      0.1446402 DIFFERENT     0.299025   0.043318   0.5645424   0.2864143   0.0974834    0.4639383
```

It is convenient to open this table in Excel. On each line, a certain combination of shifts in isotopic ratios is shown, together with the maximum change in proportion of any of the sources in the mixing solution. If the proportion of a certain source in the resulting mix changed by more than the threshold value chosen for the argument "`shift`" (default: 0.1, point 5), this combination is marked as DIFFERENT. The following columns give the estimated proportion for each source, as well as a credibility interval for the proportion (_lo and _up indicate the lower and upper margins of the interval). The proportions estimated from the original data are listed on line 0. If the configuration of the data in the isotope space was modified in such a way that the consumer ended up being outside the polygon defined by the sources, this combination is marked with a warning: problem with the data!

This table can be used to identify for which sources a change in δ^{13}C due to lipid extraction (or to the presence of lipids in the samples) is likely to affect the outcome of the mixing model in an important way. It allows also identifying threshold values for the magnitude of changes in δ^{13}C which affect the outcome of the mixing model. The results for each combination of modified values listed in the table can be considered in more detail by plotting them (point 7).

In R, the result of the function `Fatsim` is an object called a list. This list object contains the output of `siar` for the original data and for all combinations of modified parameters. It also contains the consumer and source data. This list is used to plot the results.

7. Plot the results

There are two plotting functions associated with `Fatsim`. `Fatsim.resplot` plots the mixing model results of the original data and those of a certain combination of modified values. The combination you want to plot should be identified by the associated line number in the summary output (`fatsim-output.txt`). The function is used as follows:

```
    Fatsim.resplot (simultest, 23)
```

In this case it will plot the mixing model results of the original data and of the simulation for the combination on line 23 in the output file. Example of a plot:

The white dots indicate the mode of the posterior probability distributions.

The function has two additional optional arguments:

`cct` which allows the user to plot the mode (`cct="mode"`) or the mean (`cct="mean"`) of the posterior probability distributions instead of the median. The default is `cct="mode"`.

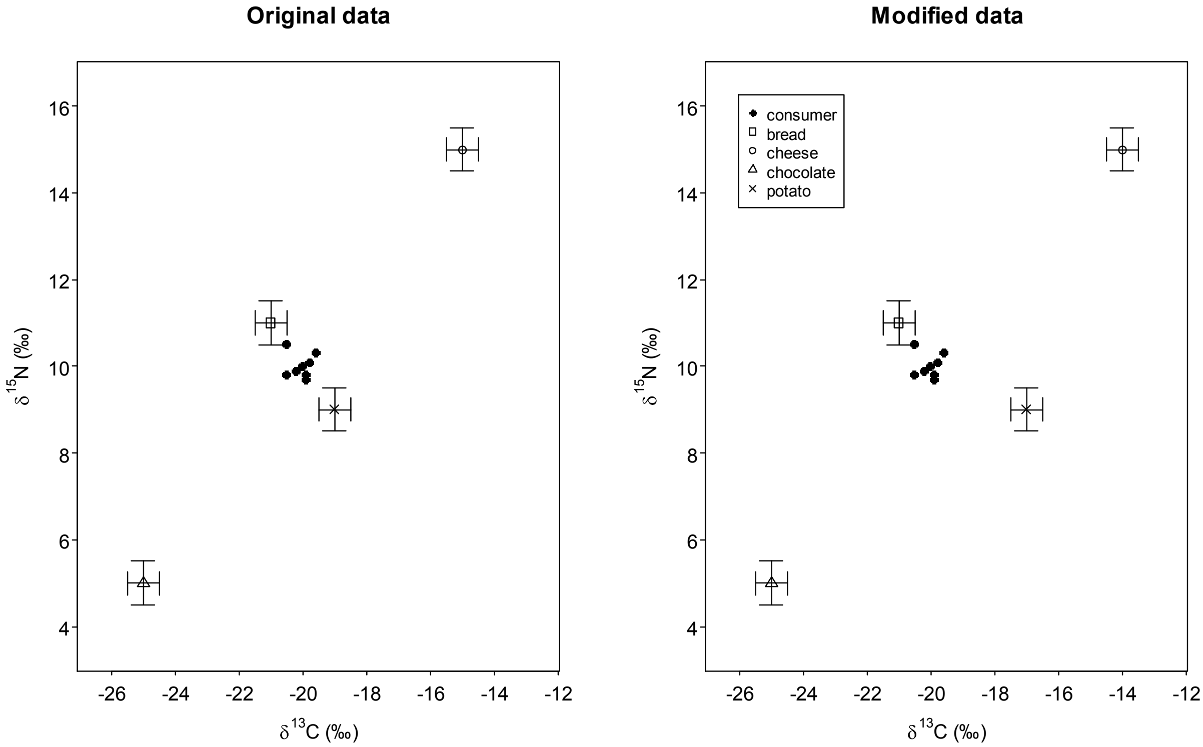
`pair`. If `pair=TRUE` (default), two plots are drawn side by side. The first one shows the mixing results for the original data and the second one shows the mixing results for the modified data. Writing `pair=FALSE` will draw only the second plot.

The plots can be saved in different formats choosing File -> Save as in the main menu of the plot window in R.

The second plotting function plots the configuration of the data in the isotopic space. It also draws the original data and one combination of modified data side by side. It can be useful to inspect the configuration for combinations where there may be a problem with the data. The function is used as follows:

```
    Fatsim.datplot (simultest, 23)
```

In this case it will plot the configuration of the original data and of the combination on line 23 in the output file. After you write the plot command, a plot window with the plot will open. You have to click on the plot to place the legend where you want. Example of a plot:



The function has one optional argument:

`pair`. If `pair=TRUE` (default), two plots are drawn side by side. Writing `pair=FALSE` will draw only the plot with the modified data.

The function Fatsim

(this section is intended for R users and describes all the arguments of the function)

The simulating function has the following arguments:

```
Fatsim <- function(pred, sources, maxmod, incrC=0.5, correc=0,
 conc = 0, incrN = 0, consumer=0, incrNcons=0, shift=0.1, cct
 = "mode", int=95, burn=50000, iter=200000, thin=15)
```

The first three (`pred, sources, maxmod)` have to be given by the user (no default), whereas the other are optional. For these parameters the default values are as given above.

`pred` A data frame containing the data for the consumer. The data frame should have two columns, the first containing the δ13C values for each individual consumer and the second containing the δ15N values. There should be as many rows as consumer individuals. It is possible to have only one row, if there is only a single consumer individual (see example included in `Fatsim` script: `predator`).

`sources` A data frame with each source as a separate row. The matrix should have five columns: the first contains the names of the sources, the second and the third the mean and standard deviation for the δ13C values of each source, and the fourth and fifth the mean and standard deviation for the δ15N values of each source (see example included in `Fatsim` script: `sour`).

`maxmod` a vector of length = number of sources, indicating the maximum change in δ^{13}C that will be simulated for each source. A negative number indicates decrease of δ^{13}C (simulating samples with increasing lipid contents) and a positive number indicates increase of δ^{13}C (simulating lipid extraction). If one or several sources should not be modified, 0 can be entered.

`consumer` the maximum change in δ^{13}C for the consumer. If the consumer should not be modified, this argument is 0.

`incrC` The size of the incremental steps by which the δ^{13}C are modified. The default is 0.5‰.

`correc` A matrix containing the mean and standard deviations of the fractional correction values for each of the isotopes. This argument is optional. The matrix has to be formatted in the same way as for `siar` (consult the `siar` help or see above, *cf.* point 4).

`conc` A matrix containing the mean and standard deviations of the concentration dependence values for each of the isotopes. This argument is optional. The matrix has to be formatted in the same way as for `siar` (consult the `siar` help or see above, *cf.* point 4).

`incrN` a vector of length = number of sources, giving a constant change in δ^{15}N for each source due to chemical lipid extraction. This argument is optional.

`incrNcons` a number giving a change in δ^{15}N due to chemical lipid extraction for the consumer. This argument is optional.

`shift` a number indicating the threshold for differences in the median of the posterior probability distributions which will be marked as DIFFERENT in the output file. The default is a difference of 0.1 in the proportion of one or several sources. This parameter has no influence on calculations or simulations and is simply intended to help reading the output files.

`cct` indicates which summary statistic of the posterior distribution will be used to compare the results from the original data to the results of the modified data. The default is `cct = “mode”`, in this case the modes of the distributions estimated by using a kernel function are used (see `siar` documentation for the function `siarhdrs` for more details). Other options are `cct = “mean”` and `cct = “median”`.

`int` indicates the width of the credibility interval which is listed in the output file in percent. The default is 95. This parameter has no influence on calculations or simulations.

`burn` defines the length of the burn-in used for the MCMC estimation (see siar documentation for details). The default is 50000.

`iter` defines the total number f itnerations used for the MCMC estimation (see siar documentation for details). The default is 200000.

`thin` defines the interval between iterations to save during the MCMC estimation (see siar documentation for details). The default is 200000.
`iter` – `burn` / `thin` has to be an integer.

The results of the function `Fatsim` are summarized in a text file (`fatsim-output.txt`), which can easily be opened in Excel. Each combination of modified δ^{13}C is listed, together with the maximum difference in the proportion of the sources in the resulting mixture compared to the result for the original data.

The result of the function `Fatsim` is a list. The first element is the output of `siar` for the original data. The subsequent elements are the outputs from siar for all combinations of modified values. Finally the list contains a matrix with all combinations of modified values, the consumer data, the source data…

Two plotting functions can be used to look at the simulation results in detail:

```
Fatsim.resplot <- function(resdat, nb, cct = "mode", pair =
TRUE)

Fatsim.datplot<- function(resdat, nb, pair = TRUE)
```

The first one produces two plots which show the results of the original data and the results of one combination of modified data side by side. The second one produces two plots which show the configuration of the original data and of one specific combination of modified data in the isotopic space (see examples above, *cf.* point 6). On the data plot, the legend is placed after the plot is drawn by clicking on the plot to choose the place for the legend.

The plotting functions have the following arguments:

`resdat` the result of the function `Fatsim` (a list object).

`nb` the line number of the combination of modified values to plot, as in the summary output of the function `Fatsim`.

`pair` if pair = TRUE (default), the original data are plotted side by side with the modified data. If pair = FALSE, only the modified data are plotted.

`cct` The white dots on the result plots indicate the median of the posterior probability distributions. Other possibilities are to plot the mean (cct = “mean”) or the median (cct = “median”). The mode is estimated using a kernel function, as in `siar` (see `siar` documentation for details).

References

DeNiro, M.J. & Epstein, S. (1977) Mechanism of carbon isotope fractionation associated with lipid-synthesis. *Science*, **197**, 261-263.

Logan, J. & Lutcavage, M. (2008) A comparison of carbon and nitrogen stable isotope ratios of fish tissues following lipid extractions with non-polar and traditional chloroform/methanol solvent systems. *Rapid Communications in Mass Spectrometry*, **22**.

Parnell, A., Inger, R., Bearhop, S. & Jackson, A.L. (2008) *SIAR: Stable Isotope Analysis in R*. http://cran.r-project.org/web/packages/siar. [accessed on 15 June 2009]

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

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

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

Tarroux, A., Ehrich, D., Lecomte, N., Jardine, T., Bêty, J. & Berteaux, D. (2010) Sensitivity of stable isotope mixing models to variation in isotopic ratios: evaluating consequences of lipid extraction. *Methods in Ecology and Evolution*, (*accepted 13-04-2010*).