

# Package ‘QFASA’

May 8, 2026

**Title** Quantitative Fatty Acid Signature Analysis

**Version** 1.2.1

**Date** 2024-08-27

**Description** Accurate estimates of the diets of predators are required in many areas of ecology, but for many species current methods are imprecise, limited to the last meal, and often biased. The diversity of fatty acids and their patterns in organisms, coupled with the narrow limitations on their biosynthesis, properties of digestion in monogastric animals, and the prevalence of large storage reservoirs of lipid in many predators, led to the development of quantitative fatty acid signature analysis (QFASA) to study predator diets.

**Depends** R (>= 3.5.0)

**License** MIT + file LICENSE

**Encoding** UTF-8

**LazyData** true

**Imports** stats, Rsolnp, boot, futile.logger, gamlss, gamlss.dist, vegan, bootstrap, Compositional, TMB, compositions, MASS, dplyr, mvtnorm,

**LinkingTo** TMB, Rcpp, RcppEigen,

**RoxygenNote** 7.3.2

**VignetteBuilder** rmarkdown, knitr

**Suggests** plyr, knitr, rmarkdown, testthat, gtools

**BugReports** <https://github.com/cstewartGH/QFASA/issues>

**URL** <https://CRAN.R-project.org/package=QFASA>

**NeedsCompilation** yes

**Author** Connie Stewart [cre, aut, cph],  
Sara Iverson [aut, cph],  
Chris Field [aut],  
Don Bowen [aut],  
Wade Blanchard [aut],

Shelley Lang [aut],  
 Justin Kamerman [aut],  
 Hongchang Bao [ctb],  
 Holly Steeves [aut],  
 Jennifer McNichol [aut],  
 Tyler Rideout [aut]

**Maintainer** Connie Stewart <connie.stewart@unb.ca>

**Repository** CRAN

**Date/Publication** 2024-08-27 19:40:02 UTC

## Contents

AIT.dist . . . . .	3
AIT.more . . . . .	3
AIT.obj . . . . .	4
backward.elimination . . . . .	4
bal.diet.data . . . . .	7
CC . . . . .	8
chisq.CA . . . . .	8
chisq.dist . . . . .	9
comp.rep . . . . .	9
conf.meth . . . . .	11
create.d.mat . . . . .	13
CS.more . . . . .	14
CS.obj . . . . .	14
FAsset . . . . .	15
forward.selection . . . . .	15
KL.dist . . . . .	18
KL.more . . . . .	18
KL.obj . . . . .	19
MEANmeth . . . . .	19
mean_geometric . . . . .	20
multiplicativeReplacement . . . . .	20
p.MLE . . . . .	21
p.MUFASA . . . . .	22
p.QFASA . . . . .	24
p.sim.QFASA . . . . .	26
p.SMUFASA . . . . .	28
POOLVARmeth . . . . .	29
predatorFAs . . . . .	30
prey.cluster . . . . .	32
prey.on.prey . . . . .	33
preyFAs . . . . .	34
pseudo.pred . . . . .	37
pseudo.pred.norm . . . . .	38
QFASA . . . . .	39
QFASA.const.eqn . . . . .	39

<i>AIT.dist</i>	3
split_prey . . . . .	40
testfordiff.ind.boot . . . . .	40
testfordiff.ind.boot.fun . . . . .	41
testfordiff.ind.pval . . . . .	41
unbal.diet.data . . . . .	42
<b>Index</b>	<b>44</b>

---

<code>AIT.dist</code>	<i>Returns the distance between two compositional vectors using Aitchison's distance measure.</i>
-----------------------	---

---

### Description

Returns the distance between two compositional vectors using Aitchison's distance measure.

### Usage

```
AIT.dist(x.1, x.2)
```

### Arguments

<code>x.1</code>	compositional vector
<code>x.2</code>	compositional vector

### References

Aitchison, J., (1992) On criteria for measures of compositional difference. *Mathematical Geology*, 24(4), pp.365-379.

Connie Stewart (2017) An approach to measure distance between compositional diet estimates containing essential zeros, *Journal of Applied Statistics*, 44:7, 1137-1152, DOI: 10.1080/02664763.2016.1193846

---

<code>AIT.more</code>	<i>Used to provide additional information on various model components evaluated at the optimal solution, i.e., using the QFASA diet estimates and Aitchison distance measure.</i>
-----------------------	---

---

### Description

Used to provide additional information on various model components evaluated at the optimal solution, i.e., using the QFASA diet estimates and Aitchison distance measure.

### Usage

```
AIT.more(alpha, predator, prey.quantiles)
```

**Arguments**

alpha	compositional QFASA diet estimate.
predator	fatty acid signature of predator.
prey.quantiles	matrix of fatty acid signatures of prey. Each row contains an individual prey signature from a different species.

---

AIT.obj	<i>Used in solnp() as the objective function to be minimized when Aitchison distance measure is chosen.</i>
---------	---

---

**Description**

Used in solnp() as the objective function to be minimized when Aitchison distance measure is chosen.

**Usage**

```
AIT.obj(alpha, predator, prey.quantiles)
```

**Arguments**

alpha	vector over which minimization takes place.
predator	fatty acid signature of predator.
prey.quantiles	matrix of fatty acid signatures of prey. Each row contains an individual prey signature from a different species.

---

backward.elimination	<i>Returns diet estimates corresponding to a sample of predators based on a backward elimination algorithm that chooses the prey species to be included in the modelling.</i>
----------------------	---

---

**Description**

Returns diet estimates corresponding to a sample of predators based on a backward elimination algorithm that chooses the prey species to be included in the modelling.

**Usage**

```
backward.elimination(
  pred.mat,
  prey.mat,
  cal.vec,
  FC,
  ext.fa,
  k = 2,
  cutoff = 0.1,
  silence = FALSE
)
```

**Arguments**

pred.mat	matrix containing the FA signatures of the predators, where each row corresponds to a predator and each column to a FA.
prey.mat	data frame containing the FA signatures of the prey, where each row corresponds to a single individual prey. The first column must index the prey group, while the remaining columns correspond to FAs.
cal.vec	numeric vector of calibration coefficients corresponding to the FAs contained in the modelling subset ext.fa. A vector of ones in the length of ext.fa may be used for modelling without calibration coefficients.
FC	numeric vector of the average lipid contents for each prey group, in the order of the <i>alphabetized</i> prey groups. vector of ones equal in length to the number of prey groups may be used for modelling without adjustment for fat content.
ext.fa	character vector containing the subset of FAs to be used in modelling.
k	scaling factor to be used in calculating the value of the information criterion (IC). The default value of 2 corresponds to the Akaike Information Criterion. For a sample of size $n$ , $k = \log(n)$ corresponds to the Bayesian Information Criterion. As this factor is numeric, values corresponding to other IC may be used freely.
cutoff	numeric proportion to be used as the threshold for the candidacy for removal of a species in backward elimination. If initial diet estimates for any individual predator find a species to be present in proportions greater than this threshold, that species will not be considered for removal from the model. This reduces computation times and safeguards against the removal of species which may be present in the diets of few predators. All species are considered candidates for removal at a value of 1, while lower values are more conservative. The default value is 0.1.
silence	if true, additional information is printed. Default is false.

**Details**

The function uses a backward elimination algorithm and the simplified MLE method to choose the prey species to be included in the model and then returns the diet estimates corresponding to these species.

**Value**

A list with components:

**Diet\_Estimates** A matrix of the diet estimates for each predator where each row corresponds to a predator and each column to a prey species. The estimates are expressed as proportions summing to one.

**Selection\_Order**

A data frame summarizing each step of the algorithm, giving the order of species removal and the corresponding IC values.

**Selection\_Tables**

A list containing a data frame for each step of the selection process, providing the IC values associated with removing any one candidate species at that step.

**See Also**

*forward.selection()*

**Examples**

```
## This example takes some time to run.
## Please uncomment code below to run.

#library(dplyr)
#library(compositions)
## Package data: FAs
#data(FAset)
#fa.set = as.vector(unlist(FAset))

## Package data: Prey
#data(prexFAs)
#prey.sub=(prexFAs[,4:(ncol(prexFAs))][fa.set])
#prey.sub=prey.sub/apply(prex.sub,1,sum)
#group=as.vector(prexFAs$Species)
#prey.sub = cbind(group,prey.sub)
#sort.preytype <- order(prex.sub[, 1])
#prey.matrix <- prey.sub[sort.preytype,]

## Package data: Predators
#data(predatorFAs)
#tombstone.info = predatorFAs[,1:4]
#predator.matrix = predatorFAs[,5:(ncol(predatorFAs))]
#npredators = nrow(predator.matrix)

## Package data: Fat content
#FC = preyFAs[,c(2,3)]
#FC = as.vector(tapply(FC$lipid,FC$Species,mean,na.rm=TRUE))

## Package data: Calibration coefficients
#data(CC)
#cal.vec = CC[,2]
#cal.mat = replicate(npredators, cal.vec)
```

```

#rownames(cal.mat) <- CC$FA
#names(cal.vec) <- rownames(cal.mat)

## QFASA (KL)
#sample.qfasa <- p.QFASA(predator.matrix,MEANmeth(pre.y.matrix),cal.mat,
#dist.meas = 1,gamma=1,FC,
#start.val = rep(1,nrow(MEANmeth(pre.y.matrix))),fa.set)

## Backward Elimination
#sample.be <- backward.elimination(predator.matrix, prey.matrix, cal.vec,FC,
#fa.set,cutoff = 0.01)

## Output
#be.estimate <- sample.be$`Diet Estimates`

```

---

bal.diet.data	<i>Sample example of balanced repeatability diet estimates data with only two repeated measurements per predator.</i>
---------------	---

---

### Description

Sample example of balanced repeatability diet estimates data with only two repeated measurements per predator.

### Usage

```
bal.diet.data
```

### Format

A data frame with 100 predator diets (50 unique predators) and 13 variables:

**Seal.ID** Predator (1 to 50)

**Year** Either 1 or 2

**capelin** estimated diet proportion

**coho** estimated diet proportion

**eulachon** estimated diet proportion

**herring** estimated diet proportion

**mackerel** estimated diet proportion

**pilchard** estimated diet proportion

**pollock** estimated diet proportion

**sandlance** estimated diet proportion

**squid** estimated diet proportion

**surfsmelt\_s** estimated diet proportion

**sufsmelt\_lg** estimated diet proportion

---

CC	<i>Fatty acid calibration coefficients.</i>
----	---

---

**Description**

Fatty acid calibration coefficients.

**Usage**

CC

**Format**

A data frame with 66 observations and 2 variables:

**FA** fatty acid names

**CC** calibration coefficient for corresponding fatty acid

---

chisq.CA	<i>Called by create.d.mat() to compute the chi-square distance.</i>
----------	---

---

**Description**

Called by create.d.mat() to compute the chi-square distance.

**Usage**

chisq.CA(x1, x2)

**Arguments**

x1                    vector

x2                    vector

---

chisq.dist	Returns the distance between two compositional vectors using the chi-square distance.
------------	---

---

**Description**

Returns the distance between two compositional vectors using the chi-square distance.

**Usage**

```
chisq.dist(x.1, x.2, gamma)
```

**Arguments**

x.1	compositional vector
x.2	compositional vector
gamma	power transform taken to be 1.

**References**

Stewart, C., Iverson, S. and Field, C. (2014) Testing for a change in diet using fatty acid signatures. *Environmental and Ecological Statistics* 21, pp. 775-792.

Connie Stewart (2017) An approach to measure distance between compositional diet estimates containing essential zeros, *Journal of Applied Statistics*, 44:7, 1137-1152, DOI: 10.1080/02664763.2016.1193846

---

comp.rep	Repeatability in Diet Estimates
----------	---------------------------------

---

**Description**

Computes a measure of repeatability for a sample of predators with repeated diet estimate measurements.

**Usage**

```
comp.rep(  
  data,  
  prey.database,  
  fatcont.mat,  
  dist.meas,  
  ext.fa,  
  B = 50,  
  R = 100,  
  CI = FALSE,  
  alpha = 0.05,
```

```

    gamma.QFASA = 1,
    gamma.rho = 1
  )

```

### Arguments

data	data frame of diet estimates. First column must denote the predator and second column the second factor (eg. year or season).
prey.database	prey data base that was used to compute the QFASA diet estimates in data. Will be used to generate pseudo predators.
fatcont.mat	data frame or matrix of length equal to the number of prey FA signatures in prey data base. First column is name of species and second column is lipid.
dist.meas	distance measure to use in p.QFASA.
ext.fa	subset of FAs to use.
B	number of pseudo predators samples to generate for bias calculation. Default is set to 50 because is slow to run.
R	number of bootstrap samples (i.e. R samples for each generated sample of pseudo predators). Default is set to 100 because it is slow to run.
CI	indicates if a confidence interval for rho is to be calculated. Default is FALSE since this is time consuming to obtain.
alpha	a (1-alpha/2)X100 percent confidence interval is calculated for rho if CI=TRUE.
gamma.QFASA	if dist.meas=3, gamma is required. Default is 1.
gamma.rho	value of gamma to be used to compute CS distance in repeatability functions. Default is 1.

### Value

Bias corrected measure of repeatability, estimate of the bias and (if CI=TRUE) a confidence interval for the true repeatability.

### References

"Repeatability for Compositional Diet Estimates with Zeros". Contact Connie Stewart (cstewart@unb.ca).

### Examples

```

## These examples take some time to run.
## Please uncomment code below to run them.

# data(prexFAs)
# data(FAs)

## Balanced Diet Data

#my.preybase <- preyFAs[,-c(1,3)]
#my.preybase[, -1] <- my.preybase[, -1]/rowSums(my.preybase[, -1])

```

```

#set.seed(10)

#comp.rep(data = bal.diet.data,prey.database=my.preybase,
#fatcont.mat = as.data.frame(prexFAs[,c(2,3)]),dist.meas=2,
#ext.fa = as.vector(unlist(FAset)))

## Unbalanced Diet Data

# my.preybase <- preyFAs[,-c(1,3)]
# my.preybase[,-1] <- my.preybase[,-1]/rowSums(my.preybase[,-1])

# set.seed(10)

# comp.rep(unbal.diet.data,my.preybase,as.data.frame(prexFAs[,c(2,3)]),2,
# as.vector(unlist(FAset)))

```

---

conf.meth

*Confidence Intervals for Diet Proportions*


---

### Description

Returns simultaneous confidence intervals for the diet of each species in the prey database.

### Usage

```

conf.meth(
  predator.mat,
  prey.mat,
  p.mat,
  cal.mat = rep(1, length(ext.fa)),
  dist.meas,
  FC = rep(1, nrow(prex.mat)),
  alpha = 0.05,
  nprey = 30,
  R.p = 1,
  R.ps = 100,
  R = 100,
  R.bias = 100,
  noise = 0,
  ext.fa
)

```

### Arguments

predator.mat	matrix containing fatty acid signatures of the predators with fatty acids summing to one.
prey.mat	prey database. A data frame with first column a Species label and other columns fatty acid proportions summing to one..

p.mat	matrix of previously computed predator diet estimates needed for confidence interval calculation.
cal.mat	matrix or vector of calibration coefficients of predators. Each COLUMN corresponds to a different predator. Default is a vector of ones. The number of fatty acids should be the same as the number of predator and prey fatty acids.
dist.meas	distance measure to use for estimation: 1=KL, 2=AIT or 3=CS
FC	vector of prey fat content, one for each individual in prey database. Note that this vector is passed to the <code>gen.pseudo.seals</code> which expects fat content values for individual prey samples while <code>pseudo.seal</code> and <code>p.QFASA</code> expect a species average. Default is a vector of ones.
alpha	1-alpha is the family-wise or overall confidence level. Default is 0.05 for an overall confidence level of 0.95.
nprey	number of prey to sample from the prey database when generating pseudo predators for the nuisance parameter estimation using original QFASA simulating code. Default is 30.
R.p	number of times to re-sample data. Due to algorithm being slow, the default parameter is 1.
R.ps	number of pseudo predators to generate when estimating nuisance parameters. Default is 100.
R	number of bootstrap replicates to use when generating p-values for confidence interval estimation. Default is 100.
R.bias	number of replicates for bias computation. Default is 100.
noise	proportion of noise to include in the generation of pseudo predators using original QFASA simulating code.
ext.fa	subset of fatty acids to be used. These should be the same as those in predator.mat, prey.mat and cal.mat.

## Details

Intervals are biased corrected as recommended in Stewart, C. (2013). Intervals are slow to obtain, particularly if there are many prey types. See vignette on parallel execution to speed up calculations.

## Value

Simultaneous  $(1-\alpha)*100$  zero-inflated beta distribution.

## References

Stewart, C. (2013) Zero-inflated beta distribution for modeling the proportions in quantitative fatty acid signature analysis. *Journal of Applied Statistics*, 40(5), 985-992.

## Examples

```
## Reducing prey database to three species so that code below will run more quickly.
## Please uncomment code to run.
```

```

#set.seed(1234)
## Fatty Acids
#data(FASet)
#fa.set = as.vector(unlist(FASet))

## Sample of Predators
#data(predatorFAs)
#predator.matrix = predatorFAs[, -c(1:4)]
#predator.matrix.ext = predatorFAs[, fa.set]
#predator.matrix.ext = predator.matrix.ext/rowSums(predator.matrix.ext)

# Prey Database
#prey.red =
#preyFAs[preyFAs$Species=="capelin"|preyFAs$Species=="herring"|preyFAs$Species=="sandlance", ]
#prey.red = prey.red[, -c(1,3)]
#prey.red.ext = prey.red[, c("Species", fa.set)]
#prey.red.ext[, -1] <- prey.red.ext[, -1]/rowSums(prey.red.ext[, -1])
#prey.red.ext.means = MEANmeth(prey.red.ext)

## Calibration Coefficients

#data(CC)
#cal.vec = CC[CC$FA %in% fa.set, 2]

#diet.est <- p.QFASA(predator.mat = predator.matrix.ext,
#                    prey.mat = prey.red.ext.means,
#                    cal.mat = cal.vec,
#                    dist.meas = 2,
#                    start.val = rep(1, nrow(prey.red.ext.means)),
#                    ext.fa = fa.set)[['Diet Estimates']]

## conf.meth needs the full prey matrix unlike p.QFASA.
#ci <- conf.meth(predator.mat = predator.matrix.ext, prey.mat = prey.red.ext, cal.mat = cal.vec,
#               p.mat = diet.est, dist.meas = 2, FC = preyFAs[,3], ext.fa = fa.set)

```

---

create.d.mat                      *Called by testfordiff.ind.boot.fun() to create a matrix of distances.*

---

## Description

Called by testfordiff.ind.boot.fun() to create a matrix of distances.

## Usage

```
create.d.mat(Y.1, Y.2)
```

## Arguments

Y.1	vector
Y.2	vector

---

CS.more	<i>Used to provide additional information on various model components evaluated at the optimal solution, i.e., using the QFASA diet estimates and chi-square distance measure.</i>
---------	--

---

**Description**

Used to provide additional information on various model components evaluated at the optimal solution, i.e., using the QFASA diet estimates and chi-square distance measure.

**Usage**

```
CS.more(alpha, predator, prey.quantiles, gamma)
```

**Arguments**

alpha	compositional QFASA diet estimate.
predator	fatty acid signature of predator.
prey.quantiles	matrix of fatty acid signatures of prey. Each row contains an individual prey signature from a different species.
gamma	power transform exponent (see <code>chisq.dist()</code> ).

---

CS.obj	<i>Used in <code>solnp()</code> as the objective function to be minimized when chi-square distance measure is chosen. Unlike <code>AIT.obj()</code> and <code>KL.obj()</code>, does not require modifying zeros.</i>
--------	--

---

**Description**

Used in `solnp()` as the objective function to be minimized when chi-square distance measure is chosen. Unlike `AIT.obj()` and `KL.obj()`, does not require modifying zeros.

**Usage**

```
CS.obj(alpha, predator, prey.quantiles, gamma)
```

**Arguments**

alpha	vector over which minimization takes place.
predator	fatty acid signature of predator.
prey.quantiles	matrix of fatty acid signatures of prey. Each row contains an individual prey signature from a different species.
gamma	power transform exponent (see <code>chisq.dist()</code> ).

---

FAset	<i>List of fatty acids used in sample prey and predator data sets, preyFAs and predatorFAs respectively.</i>
-------	--

---

**Description**

List of fatty acids used in sample prey and predator data sets, preyFAs and predatorFAs respectively.

**Usage**

FAset

**Format**

A data frame with 39 observations and 1 variable:

**FA** Fatty acid name

---

forward.selection	<i>Returns diet estimates corresponding to a sample of predators based on a forward selection algorithm that chooses the prey species to be included in the modelling.</i>
-------------------	--

---

**Description**

Returns diet estimates corresponding to a sample of predators based on a forward selection algorithm that chooses the prey species to be included in the modelling.

**Usage**

```
forward.selection(  
  pred.mat,  
  prey.mat,  
  cal.vec,  
  FC,  
  ext.fa,  
  k = 2,  
  min.spec = 5,  
  starting.spec = NULL,  
  silence = FALSE  
)
```

**Arguments**

pred.mat	matrix containing the FA signatures of the predators, where each row corresponds to a predator and each column to a FA.
prey.mat	data frame containing the FA signatures of the prey, where each row corresponds to a single individual prey. The first column must index the prey group, while the remaining columns correspond to FAs.
cal.vec	numeric vector of calibration coefficients corresponding to the FAs contained in the modelling subset ext.fa. A vector of ones in the length of ext.fa may be used for modelling without calibration coefficients.
FC	numeric vector of the average lipid contents for each prey group, in the order of the <i>alphabetized</i> prey groups. vector of ones equal in length to the number of prey groups may be used for modelling without adjustment for fat content.
ext.fa	character vector containing the subset of FAs to be used in modelling.
k	scaling factor to be used in calculating the value of the information criterion (IC). The default value of 2 corresponds to the Akaike Information Criterion. For a sample of size $n$ , $k = \log(n)$ corresponds to the Bayesian Information Criterion. As this factor is numeric, values corresponding to other IC may be used freely.
min.spec	optional integer value specifying the minimum final model size for forward selection. By default, forward selection will add species to the model until the value of the chosen IC ceases to improve. If this parameter is increased, forward selection will add the best available species up to the specified minimum model size before continuing with the default selection process.
starting.spec	optional character vector specifying the starting species for the forward selection algorithm. Where known, two or more species may be specified to ensure their inclusion in the final model, reducing computation times for the algorithm. The default is NULL.
silence	if true, additional information is printed. Default is false.

**Details**

The function uses a forward selection algorithm and the simplified MLE method to choose the prey species to be included in the model and then returns the diet estimates corresponding to these species.

**Value**

A list with components:

Diet_Estimates	A matrix of the diet estimates for each predator where each row corresponds to a predator and each column to a prey species. The estimates are expressed as proportions summing to one.
Selection_Order	A data frame summarizing each step of the algorithm, giving the order of species selection and the corresponding IC values.
Selection_Tables	A list containing a data frame for each step of the selection process, providing the IC values associated with adding any one candidate species at that step.

**See Also***backward.elimination()***Examples**

```

## This example takes some time to run.
## Please uncomment code below to run.

#library(dplyr)
#library(compositions)
## Package data: FAs
#data(FAs)
#fa.set = as.vector(unlist(FAs))

## Package data: Prey
#data(prexFAs)
#prey.sub=(prexFAs[,4:(ncol(prexFAs))])[fa.set]
#prey.sub=prey.sub/apply(prex.sub,1,sum)
#group=as.vector(prexFAs$Species)
#prey.sub = cbind(group,prey.sub)
#sort.preytype <- order(prex.sub[, 1])
#prey.matrix <- prey.sub[sort.preytype,]

## Package data: Predators
#data(predatorFAs)
#tombstone.info = predatorFAs[,1:4]
#predator.matrix = predatorFAs[,5:(ncol(predatorFAs))]
#npredators = nrow(predator.matrix)

## Package data: Fat content
#FC = preyFAs[,c(2,3)]
#FC = as.vector(tapply(FC$lipid,FC$Species,mean,na.rm=TRUE))

## Package data: Calibration coefficients
#data(CC)
#cal.vec = CC[,2]
#cal.mat = replicate(npredators, cal.vec)
#rownames(cal.mat) <- CC$FA
#names(cal.vec) <- rownames(cal.mat)

## QFASA (KL)
#sample.qfasa <- p.QFASA(predator.matrix,MEANmeth(prex.matrix),cal.mat,
#dist.meas = 1,gamma=1,FC,
#start.val = rep(1,nrow(MEANmeth(prex.matrix))),fa.set)

## Forward Selection
#sample.fs <- forward.selection(predator.matrix,prex.matrix,cal.vec,FC,fa.set,
#min.spec = 5,starting.spec = c("capelin", "herring"))
## Output
#fs.estimate <- sample.fs$`Diet Estimates`

```

---

KL.dist	<i>Returns the distance between two compositional vectors using Kullback–Leibler distance measure.</i>
---------	--

---

**Description**

Returns the distance between two compositional vectors using Kullback–Leibler distance measure.

**Usage**

```
KL.dist(x.1, x.2)
```

**Arguments**

x.1	compositional vector
x.2	compositional vector

**References**

S.J. Iverson, C. Field, W.D. Bowen, and W. Blanchard (2004) Quantitative fatty acid signature analysis: A new method of estimating predator diets, *Ecological Monographs* 72, pp. 211-235.

---

KL.more	<i>Used to provide additional information on various model components evaluated at the optimal solution, i.e., using the QFASA diet estimates and Kullback-Leibler distance measure.</i>
---------	--

---

**Description**

Used to provide additional information on various model components evaluated at the optimal solution, i.e., using the QFASA diet estimates and Kullback-Leibler distance measure.

**Usage**

```
KL.more(alpha, predator, prey.quantiles)
```

**Arguments**

alpha	compositional QFASA diet estimate.
predator	fatty acid signature of predator.
prey.quantiles	matrix of fatty acid signatures of prey. Each row contains an individual prey signature from a different species.

---

KL.obj	<i>Used in solnp() as the objective function to be minimized when Kullback–Leibler distance measure is chosen.</i>
--------	--

---

**Description**

Used in solnp() as the objective function to be minimized when Kullback–Leibler distance measure is chosen.

**Usage**

```
KL.obj(alpha, predator, prey.quantiles)
```

**Arguments**

alpha	vector over which minimization takes place.
predator	fatty acid signature of predator.
prey.quantiles	matrix of fatty acid signatures of prey. Each row contains an individual prey signature from a different species.

---

MEANmeth	<i>Returns the multivariate mean FA signature of each prey group entered into the QFASA model. Result can be passed to prey.mat in p.QFASA().</i>
----------	---

---

**Description**

Returns the multivariate mean FA signature of each prey group entered into the QFASA model. Result can be passed to prey.mat in p.QFASA().

**Usage**

```
MEANmeth(preymat)
```

**Arguments**

preymat	matrix containing the FA signatures of the prey. The first column indexes the prey group.
---------	---

mean\_geometric      *Returns the geometric mean of a compositional vector*

---

**Description**

Returns the geometric mean of a compositional vector

**Usage**

```
mean_geometric(x)
```

**Arguments**

x                      compositional vector

---

multiplicativeReplacement  
*Multiplicative replacement of zeroes*

---

**Description**

Multiplicative replacement of zeroes

**Usage**

```
multiplicativeReplacement(compositional.mat, delta = 1e-05)
```

**Arguments**

compositional.mat      matrix containing compositional data (for example, FA signatures) that may contain zeros.  
delta                    imputed value

**Value**

The compositional matrix with zeros modified.

---

p.MLE	Returns simplified MLE diet estimates corresponding to a sample of predators.
-------	---

---

### Description

Computes the diet estimate for each predator in *pred.mat* using the simplified MLE method, without the use of random effects.

### Usage

```
p.MLE(pred.mat, prey.mat, cal.mat, FC, ext.fa)
```

### Arguments

pred.mat	matrix containing the FA signatures of the predators.
prey.mat	matrix containing the FA signatures of the individual prey. The first column must index the prey group. <i>prey.mat</i> is the prey database.
cal.mat	matrix of calibration factors where the <i>i</i> th column is to be used with the <i>i</i> th predator. If modelling is to be done without calibration coefficients, simply pass a vector or matrix of ones.
FC	vector of fat content of length equal to the number of prey groups or species.
ext.fa	subset of fatty acids to be used to obtain diet estimates.

### Details

The assumed model is similar to the MUFASA model but the random effects are replaced by the prey species' sample means to speed up computations. Unlike *p.MUFASA*, this function does not require integration and hence is much faster.

### Value

A list with components:

Diet_Estimates	A matrix of the diet estimates for each predator where each row corresponds to a predator and the columns to prey species. The estimates are expressed as proportions summing to one.
Var_Epsilon	Optimized values of error variance. See reference.
nll	Negative log likelihood values. As per <i>solnp</i> documentation, <i>nll</i> is a "vector of function values during optimization with last one the value at the optimal".

### References

Steeves, Holly (2020) Maximum likelihood approach to diet estimation and inference based on fatty acid signatures. PhD thesis available at <https://dalspace.library.dal.ca/handle/10222/80034>.

**Examples**

```
## This example takes some time to run.
## Please uncomment the code below to run.

#library(dplyr)
#library(compositions)

## Fatty Acids
#data(FAs)
#ext.fa <- as.vector(unlist(FAs))

## Predators
#data(predatorFAs)
#pred.mat <- predatorFAs[, -c(1:4)]
#n.pred <- nrow(pred.mat)

## Prey
#data(prexFAs)
#prey.mat <- prexFAs[, -c(1,3)]

#FC = prexFAs[,c(2,3)]
#FC = as.vector(tapply(FC$lipid,FC$Species,mean,na.rm=TRUE))

## Calibration Coefficients
#data(CC)
#cal.vec = CC[,2]
#cal.mat = replicate(n.pred, cal.vec)
#rownames(cal.mat) <- CC$FA

## Diet Estimates
#mle.est <- p.MLE(pred.mat, prey.mat, cal.mat, FC, ext.fa)
#mle.est$"Diet Estimates"
```

---

p.MUFASA

*Returns MUFASA diet estimates corresponding to a sample of predators.*

---

**Description**

Computes the diet estimate for each predator in *pred.mat* using MLE method.

**Usage**

```
p.MUFASA(pred.mat, prey.mat, cal.mat, FC, ext.fa)
```

**Arguments**

pred.mat	matrix containing the FA signatures of the predators.
prey.mat	matrix containing FA signatures from each prey group The first column must index the prey group. <i>prey.mat</i> is the prey database.
cal.mat	matrix of calibration factors where the <i>i</i> th column is to be used with the <i>i</i> th predator. If modelling is to be done without calibration coefficients, simply pass a vector or matrix of ones. <i>cal.mat</i> must contain names of FAs.
FC	vector of fat content of length equal to the number of prey groups or species.
ext.fa	subset of fatty acids to be used to obtain QFASA diet estimates.

**Value**

A list with components:

Diet_Estimates	A matrix of the diet estimates for each predator where each row corresponds to a predator and the columns to prey species. The estimates are expressed as proportions summing to one.
nll	Negative log likelihood values. As per <i>solnp</i> documentation, <i>nll</i> is "Vector of function values during optimization with last one the value at the optimal".
Var_Epsilon	Optimized values of error variance. See reference.

**References**

Steeves, Holly (2020) Maximum likelihood approach to diet estimation and inference based on fatty acid signatures. PhD thesis available at <https://dalspace.library.dal.ca/handle/10222/80034>.

**See Also**

*p.MLE()* for a simplified version of *p.MUFASA()* that is faster to run.

**Examples**

```
## This example takes some time to run.
## Please uncomment code below to run.

#library(dplyr)
#library(compositions)
## Fatty Acids
#data(FAset)
#fa.set = as.vector(unlist(FAset))

## Predators
#data(predatorFAs)
#tombstone.info = predatorFAs[,1:4]
#predator.matrix = predatorFAs[,5:(ncol(predatorFAs))]
#npredators = nrow(predator.matrix)

## Prey
## Extracting a small number of species to speed up calculations for the example.
```

```

#data(prexFAs)
#prey.matrix = preyFAs[,-c(1,3)]
#spec.red <-c("capelin", "herring", "mackerel", "pilchard", "sandlance")
#spec.red <- sort(spec.red)
#prey.red <- prey.matrix %>% filter(Species %in% spec.red)

## Fat content
#FC = preyFAs[,c(2,3)]
#FC = FC %>% arrange(Species)
#FC.vec = tapply(FC$lipid,FC$Species,mean,na.rm=TRUE)
#FC.red <- FC.vec[spec.red]

## Calibration Coefficients
#data(CC)
#cal.vec = CC[,2]
#cal.m = replicate(npredators, cal.vec)
#rownames(cal.m) <- CC$FA

#M <- p.MUFASA(predator.matrix, prey.red, cal.m, FC.red, fa.set)

## Diet Estimates

#M$Diet_Estimates

```

---

p.QFASA

*Returns QFASA diet estimates corresponding to a sample of predators.*


---

## Description

Computes the diet estimate for each predator in *predator.mat* using either the Kullback-Leibler Distance (KL), the Aitchison Distance (AIT) or the Chi-Square Distance (CS).

## Usage

```

p.QFASA(
  predator.mat,
  prey.mat,
  cal.mat,
  dist.meas,
  gamma = 1,
  FC = rep(1, nrow(preymat)),
  start.val = rep(0.99999, nrow(preymat)),
  ext.fa
)

```

## Arguments

*predator.mat* matrix containing the FA signatures of the predators.

prey.mat	matrix containing a representative FA signature from each prey group (usually the mean). The first column must index the prey group. Note can use function <i>MEANmeth</i> to calculate the means.
cal.mat	matrix of calibration factors where the <i>i</i> th column is to be used with the <i>i</i> th predator. If modelling is to be done without calibration coefficients, simply pass a vector or matrix of ones.
dist.meas	distance measure to use for estimation: 1=KL, 2=AIT or 3=CS
gamma	parameter required for calculations using CS distance (passed to CS.obj). Currently being set to 1.
FC	vector of fat content of length equal to the number of prey groups or species.
start.val	initial vector of parameters to be optimized
ext.fa	subset of fatty acids to be used to obtain QFASA diet estimates.

### Details

Before carrying out an analysis using QFASA, rows of prey database must be normalized to sum to one. See Example for code that extracts a subset of FAs and then normalizes the prey database signatures.

### Value

A list with components:

Diet Estimates	A matrix of the diet estimates for each predator where each row corresponds to a predator and the columns to prey species. The estimates are expressed as proportions summing to one.
Additional Measures	For each predator for which a diet estimate was obtained:
ModFAS	the value of the modelled fatty acid. These are expressed as proportions summing to one.
DistCont	The contribution of each fatty acid to the final minimized distance.
PropDistCont	The contribution of each fatty acid to the final minimized distance as a proportion of the total.
MinDist	The final minimized distance.

### References

Iverson, Sara J., Field, Chris, Bowen, W. Don and Blanchard, Wade (2004) Quantitative Fatty Acid Signature Analysis: A New Method of Estimating Predator Diets. *Ecological Monographs*, 74(2), 211-235

### Examples

```
## Fatty Acids
data(FAs)
fa.set = as.vector(unlist(FAs))
```

```

## Predators
data(predatorFAs)
tombstone.info = predatorFAs[,1:4]
predator.matrix = predatorFAs[,5:(ncol(predatorFAs))]
npredators = nrow(predator.matrix)

## Prey
data(prexFAs)
prey.sub=(prexFAs[,4:(ncol(prexFAs))])[fa.set]
prey.sub=prey.sub/apply(prex.sub,1,sum)
group=as.vector(prexFAs$Species)
prey.matrix=cbind(group,prey.sub)
prey.matrix=MEANmeth(prey.matrix)

## Fat Content

FC = prexFAs[,c(2,3)]
FC = as.vector(tapply(FC$lipid,FC$Species,mean,na.rm=TRUE))

## Calibration Coefficients
data(CC)
cal.vec = CC[,2]
cal.mat = replicate(npredators, cal.vec)

## Run QFASA
Q = p.QFASA(predator.matrix,
            prey.matrix,
            cal.mat,
            dist.meas = 1,
            gamma=1,
            FC,
            start.val = rep(1,nrow(prey.matrix)),
            fa.set)

## Diet Estimates
DietEst = Q$'Diet Estimates'

```

---

p.sim.QFASA

*Simultaneous estimation of diet composition and calibration coefficients*

---

## Description

Computes the diet estimate for each predator in *pred.sig* as well as an overall estimate of the calibration coefficient vector.

## Usage

```
p.sim.QFASA(pred.sig, prey.mat, FC = rep(1, nrow(prey.mat)))
```

**Arguments**

pred.sig	matrix containing the FA signatures of the predator
prey.mat	matrix containing a representative FA signature from each prey group (usually the mean). The first column must index the prey group.
FC	vector of fat content of length equal to the number of prey groups (or species)

**Details**

Starting values for the diet estimates are equal proportions and a vector of ones is used for the calibration coefficients.

**Value**

A list with components:

diet.est	A matrix of the diet estimates for each predator where each row corresponds to a predator and the columns to prey species. The estimates are expressed as proportions summing to one.
cc.est	Estimated vector of calibration coefficients

**References**

Bromaghin, Jeffrey F., Budge, Suzanne M., Thiemann, Gregory and Rode, Karyn D. (2017) Simultaneous estimation of the diet composition and calibration coefficients with fatty acid signature data. *Ecology and Evolution*, 7(16), 6103-6113

**Examples**

```
## This example takes some time to run.
## Please uncomment code below to run.

## Fatty Acids
#data(FAs)
#fa.set = as.vector(unlist(FAs))

## Predators
#data(predatorFAs)
#tombstone.info = predatorFAs[,1:4]
#predator.matrix = predatorFAs[,5:(ncol(predatorFAs))]
#npredators = nrow(predator.matrix)

## Need predator and prey to have same length

#predator.ext <- predator.matrix[fa.set]
#predator.ext <- predator.ext/rowSums(predator.ext)

## Prey
#data(prexFAs)
#prey.sub=(prexFAs[,4:(ncol(prexFAs))])[fa.set]
#prey.sub=prey.sub/apply(prex.sub,1,sum)
```

```

#group=as.vector(prexFAs$Species)
#prey.matrix=cbind(group,prey.sub)
#prey.matrix=MEANmeth(pre.matrix)

## Fat Content
#FC = preyFAs[,c(2,3)]
#FC = as.vector(tapply(FC$lipid,FC$Species,mean,na.rm=TRUE))

#Q.sim <-p.sim.QFASA(predator.ext,prey.matrix,FC)
## Average Diet Estimate
#round(colMeans(Q.sim[[1]]),3)
## Calibration Coefficients
#Q.sim[[2]]

```

---

p.SMUFASA

*Simultaneous maximum unified fatty acid signature analysis*


---

## Description

Returns SMUFASA calibration coefficient estimates and an average diet among a sample of predators.

## Usage

```
p.SMUFASA(pred.mat, prey.mat, FC, ext.fa)
```

## Arguments

pred.mat	matrix containing the FA signatures of the predators.
prey.mat	matrix containing FA signatures from each prey group. The first column must index the prey group. <i>prey.mat</i> is the prey database.
FC	vector of fat content of length equal to the number of prey groups or species.
ext.fa	subset of fatty acids to be used to obtain estimates.

## Details

Calibration coefficients (CCs) are not supplied but are instead estimated. While one overall diet is computed, the CCs can be used in p.QFASA or p.MUFASA to estimate individual diet estimates.

## Value

A list with components:

Cal_Estimates	A vector of estimated calibration coefficients common to all predators. The $k$ th value corresponds to the $k$ th fatty acid. The estimates sum to the number of fatty acids.
Diet_Estimate	A vector of estimates of the average diet among the predators. The estimates are expressed as proportions summing to one.

Var\_Epsilon      Optimized values of error variance.  
 nll                Negative log likelihood values. As per *solnp* documentation, *nll* is "Vector of function values during optimization with last one the value at the optimal".

### Examples

```
## This example takes some time to run.
## Please uncomment code below to run.

#library(dplyr)
#library(compositions)
## Fatty Acids
#data(FASet)
#fa.set = as.vector(unlist(FASet))

## Predators
#data(predatorFAS)
#tombstone.info = predatorFAS[,1:4]
#predator.matrix = predatorFAS[,5:(ncol(predatorFAS))]
#npredators = nrow(predator.matrix)

## Prey
## Extracting a small number of species to speed up calculations for the example.
#data(prexFAS)
#prey.matrix = preyFAS[,-c(1,3)]
#spec.red <-c("capelin", "herring", "mackerel", "pilchard", "sandlance")
#spec.red <- sort(spec.red)
#prey.red <- prey.matrix %>% filter(Species %in% spec.red)

## Fat content
#FC = preyFAS[,c(2,3)]
#FC = FC %>% arrange(Species)
#FC.vec = tapply(FC$lipid,FC$Species,mean,na.rm=TRUE)
#FC.red <- FC.vec[spec.red]

#out <- p.SMUFAASA(predator.matrix, prey.red, FC.red, fa.set)

#out$Cal_Estimates
```

---

POOLVARmeth

*Computes within species variance-covariance matrices on transformed scaled, along with a pooled estimate.*

---

### Description

Computes within species variance-covariance matrices on transformed scaled, along with a pooled estimate.

**Usage**

```
POOLVARmeth(preymat)
```

**Arguments**

`preymat` matrix containing transformed FA signatures of the prey. Note that the first column indexes prey type.

**Value**

Returns the variance-covariance matrix of each prey type as well as a pooled estimate of the variance-covariance matrix

---

predatorFAs	<i>Predator fatty acid signatures. Each predator signature is a row with fatty acid proportions in columns.</i>
-------------	---

---

**Description**

Fatty acid signatures are subsetted for the chosen fatty acid set and renormalized during the modelling so there is no need to subset and/or renormalize prior to running p.QFASA. However, make sure that the the same fatty acids appear in the predator and prey files (if a FA appears in one but not the other the code will give you an error).

**Usage**

```
predatorFAs
```

**Format**

A data frame with 10 observations and 70 variables:

```

SampleCode TODO
AnimalCode TODO
SampleGroup TODO
Biopsy TODO
c12.0
c13.0
Iso14
c14.0
c14.1w9
c14.1w7
c14.1w5
Iso15

```

**Anti15**

**c15.0**

**c15.1w8**

**c15.1w6**

**Iso16**

**c16.0**

**c16.1w11**

**c16.1w9**

**c16.1w7**

**c7Mec16.0**

**c16.1w5**

**c16.2w6**

**Iso17**

**c16.2w4**

**c16.3w6**

**c17.0**

**c16.3w4**

**c17.1**

**c16.4w3**

**c16.4w1**

**c18.0**

**c18.1w13**

**c18.1w11**

**c18.1w9**

**c18.1w7**

**c18.1w5**

**c18.2d5.11**

**c18.2w7**

**c18.2w6**

**c18.2w4**

**c18.3w6**

**c18.3w4**

**c18.3w3**

**c18.3w1**

**c18.4w3**

**c18.4w1**

**c20.0**

c20.1w11  
c20.1w9  
c20.1w7  
c20.2w9  
c20.2w6  
c20.3w6  
c20.4w6  
c20.3w3  
c20.4w3  
c20.5w3  
c22.1w11  
c22.1w9  
c22.1w7  
c22.2w6  
c21.5w3  
c22.4w6  
c22.5w6  
c22.4w3  
c22.5w3  
c22.6w3  
c24.1w9

### Details

Unlike the original QFASApack code the predator data can contain as much tombstone data in columns as you wish but the predator FA signatures must be extracted as a separate input in order to run in p.QFASA.

---

prey.cluster	<i>Produces a dendrogram using distances between the mean FA signatures of the prey types.</i>
--------------	--

---

### Description

Performs a hierarchical cluster analysis of mean prey fatty acid signatures using function hclust.

### Usage

```
prey.cluster(pre.f.a, method = "complete", dist.meas = 2)
```

**Arguments**

prey.fa	data frame of prey fatty acid signature samples. First column must be species used to group samples. Other columns are assumed to be fatty acid proportions.
method	the agglomeration method to be used. This should be one of the possible methods in hclust such as "single", "complete" or "average". Default is "complete".
dist.meas	distance measure to use for calculating dissimilarities: 1=KL, 2=AIT or 3=CS. Default is AIT.

**Value**

Plot (dendrogram)

**Examples**

```
## Fatty Acids
data(FAs)
fa.set = as.vector(unlist(FAs))

## prey.cluster requires full prey database.
data(prexFAs)
prey.sub=(prexFAs[,4:(ncol(prexFAs))])[fa.set]
prey.sub=prey.sub/apply(prex.sub,1,sum)
group=as.vector(prexFAs$Species)
prey.matrix=cbind(group,prey.sub)

prey.cluster(prex.matrix,method="average",dist.meas=3)
```

---

prey.on.prey	<i>Each prey fatty acid signature is systematically removed from the supplied prey database and its QFASA diet estimate is obtained by treating the individual as a predator.</i>
--------------	---

---

**Description**

Each prey fatty acid signature is systematically removed from the supplied prey database and its QFASA diet estimate is obtained by treating the individual as a predator.

**Usage**

```
prey.on.prey(preibase, dist.meas, gamma = 1)
```

**Arguments**

preibase	first column is name of species and remaining columns are fatty acids.
dist.meas	see help file for <a href="#">p.QFASA</a> .
gamma	see help file for <a href="#">p.QFASA</a> .

**Value**

diet estimate

**Examples**

```
data(prexFAs)
my.preybase <- preyFAs[, -c(1,3)]

# Note: uncomment examples to run. CRAN tests fail because execution time > 5 seconds
# diets.out <- prey.on.prey(my.preybase, 2)
# round(MEANmeth(diets.out), 3)
```

---

preyFAs	<i>Prey fatty acid signatures. Each prey signature is a row with fatty acid proportions in columns.</i>
---------	---

---

**Description**

The prey file should contain all of the individual fatty acid signatures of the prey and their lipid contents (where appropriate) - a matrix of the mean values for the FAs (prey.matrix) by the designated prey modelling group is then calculated using the MEANmeth function.

**Usage**

```
preyFAs
```

**Format**

A data frame with 302 observations and 70 variables:

**Lab.Code** TODO

**Species** TODO

**lipid** TODO

**c12.0**

**c13.0**

**Iso14**

**c14.0**

**c14.1w9**

**c14.1w7**

**c14.1w5**

**Iso15**

**Anti15**

**c15.0**

c15.1w8  
c15.1w6  
Iso16  
c16.0  
c16.1w11  
c16.1w9  
c16.1w7  
c7Me16.0  
c16.1w5  
c16.2w6  
Iso17  
c16.2w4  
c16.3w6  
c17.0  
c16.3w4  
c17.1  
c16.3w1  
c16.4w3  
c16.4w1  
c18.0  
c18.1w13  
c18.1w11  
c18.1w9  
c18.1w7  
c18.1w5  
c18.2d5.11  
c18.2w7  
c18.2w6  
c18.2w4  
c18.3w6  
c18.3w4  
c18.3w3  
c18.3w1  
c18.4w3  
c18.4w1  
c20.0  
c20.1w11

c20.1w9  
c20.1w7  
c20.2w9  
c20.2w6  
c20.3w6  
c20.4w6  
c20.3w3  
c20.4w3  
c20.5w3  
c22.1w11  
c22.1w9  
c22.1w7  
c22.2w6  
c21.5w3  
c22.4w6  
c22.5w6  
c22.4w3  
c22.5w3  
c22.6w3  
c24.1w9

### Details

Like the predator .csv file you can have as many tombstone data columns as required but there must be at least one column that identifies the modelling group, in this case, Species.

Unlike the predator data, the prey data is not subsetted and renormalized during the modelling so the prey file needs to be subsetted for the desired fatty acid set and renormalized to sum to 1 prior to calculating the mean values.

The full FA set is extracted from the data frame (columns 4 onward), subsetted for the FA set in use and then renormalized over 1. The modelling group names (the "Species" column in this case) is then added back to the subsetted and renormalized data (as the first column) and the average values calculated using the MEANmeth function. Note that for the MEANmeth function to work the modelling group name must be in the first column.

---

pseudo.pred	<i>Generate a pseudo predator by sampling with replacement from prey database.</i>
-------------	--

---

### Description

Generates a single pseudo predator by sampling with replacement from prey database. To generate a sample of pseudo predators, please refer to example code.

### Usage

```
pseudo.pred(diet, preybase, cal.vec, fat.vec, preysize = 2)
```

### Arguments

diet	the "true" or "desired" diet of the pseudo predator with prey species in alphabetical order (i.e.in the order of table <pre>preyFAs[,2]</pre> ). A compositional vector of proportions that sums to one with length equal to the number of prey species.
preybase	prey database from which to generate the pseudo predator. First column must provide the species name.
cal.vec	vector of calibration coefficients whose length is the same as the number of fatty acids in prey database.
fat.vec	vector of fat content whose length is the same as the number of species.
preysize	number of prey to sample from prey database. If preysize=1, then one prey is selected from each species. Otherwise, a sample of n_k signatures (where n_k is sample size for species k) is obtained by sampling with replacement.

### Details

The default is to re-sample all of the prey signatures within each species (that is, preysize=2). Alternatively, one prey may be randomly selected from each species yielding potentially more variable pseudo-predators. For details on simulating realistic predators signatures, see Bromaghin, J. (2015) Simulating realistic predator signatures in quantitative fatty acid signature analysis, *Ecological Informatics*, 30, 68-71.

### Value

A simulated predator FA signature.

### Examples

```
data

```
(preyFAs)

# Generating a sample of 10 pseudo predators each with "true" diet being
# (1/11,1/11,...1/11), no calibration effect and no fat content. The QFASA diet estimate
# is then computed for each pseudo predator.
```


```

```

# Note: To incorporate calibration and fat content in a simulation study,
# one set of calibration and fat content is generally used to simulate the pseudo predator
# and another is used to estimate the diet.

set.seed(11)
p.mat <- matrix(rep(NA,10*11),nrow=10)
for (i in 1: 10) {
  my.seal <- pseudo.pred(rep(1/11,11),
                        preyFAs[-c(1,3)],
                        rep(1,ncol(preyFAs[-c(1,3)])-1),
                        rep(1,11))
  p.mat[i,] <- p.QFASA(my.seal,
                     MEANmeth(preyFAs[-c(1,3)]),
                     rep(1,length(my.seal)),
                     2,
                     ext.fa=colnames(preyFAs[-c(1:3)]))$`Diet Estimates`
}

# Can verify that average diet estimate of the 10 pseudo predators is close to
# "true" diet.

colnames(p.mat) <- as.vector(rownames(MEANmeth(preyFAs[-c(1,3)])))
round(apply(p.mat,2,mean),3)

```

---

pseudo.pred.norm	<i>Generate a pseudo predator parametrically from multivariate normal distributions.</i>
------------------	--

---

## Description

Generate a pseudo predator parametrically from multivariate normal distributions.

## Usage

```
pseudo.pred.norm(mu.mat, sigma.pool, diet)
```

## Arguments

mu.mat	matrix where each row represents the mean transformed FA signature of each prey type
sigma.pool	pooled variance-covariance matrix of the transformed fatty acid signatures of prey types
diet	the "true" or "desired" diet of the pseudo predator with prey species in the same order as the rows of mu.mat. A compositional vector of proportions that sums to one with length equal to the number of prey species.

**Details**

Similar to *pseudo.pred* but instead generates the pseudo-predators parametrically by assuming ilr transformed FA signatures have a multivariate normal distribution.

**Value**

A simulated predator FA signature. See *pseudo.pred* for an example illustrating how to generate a sample of pseudo predators.

---

 QFASA

*QFASA: A package for Quantitative Fatty Acid Signature Analysis*


---

**Description**

Accurate estimates of the diets of predators are required in many areas of ecology, but for many species current methods are imprecise, limited to the last meal, and often biased. The diversity of fatty acids and their patterns in organisms, coupled with the narrow limitations on their biosynthesis, properties of digestion in monogastric animals, and the prevalence of large storage reservoirs of lipid in many predators, led us to propose the use of quantitative fatty acid signature analysis (QFASA) to study predator diets.

---

 QFASA.const.eqn

*Returns sum(alpha) and used in solnp().*


---

**Description**

Returns sum(alpha) and used in solnp().

**Usage**

```
QFASA.const.eqn(alpha, predator, prey.quantiles, gamma)
```

**Arguments**

alpha	vector over which minimization takes place.
predator	fatty acid signature of predator.
prey.quantiles	matrix of fatty acid signatures of prey. Each row contains an individual prey signature from a different species.
gamma	power transform exponent (see chisq.dist).

---

split_prej	<i>Splits prey database into a simulation set (1/3) and a modelling set (2/3). Returns a list:</i>
------------	--

---

### Description

1. simulation prey database
  2. modelling prey database
1. simulation prey database
  2. modelling prey database

### Usage

```
split_prej(prej.mat)
```

```
split_prej(prej.mat)
```

### Arguments

prej.mat	matrix of individual prey fatty acid signatures where the first column denotes the prey type
----------	--

### Details

IF number of samples of a prey type  $\leq 5$ , then prej.mod AND prej.sim are duplicated instead of split.

IF number of samples of a prey type  $\leq 5$ , then prej.mod AND prej.sim are duplicated instead of split.

---

testfordiff.ind.boot	<i>Called by testfordiff.ind.pval().</i>
----------------------	--

---

### Description

Called by testfordiff.ind.pval().

### Usage

```
testfordiff.ind.boot(data, ns1, R)
```

### Arguments

data	sample of compositional data
ns1	sample size of compdata.1
R	number of bootstrap samples. default is 500.

---

`testfordiff.ind.boot.fun`*Called by testfordiff.ind.boot().*

---

**Description**

Called by testfordiff.ind.boot().

**Usage**

```
testfordiff.ind.boot.fun(data, i, ns1, change.zero = 1e-05)
```

**Arguments**

<code>data</code>	sample of compositional data
<code>i</code>	row index
<code>ns1</code>	sample size of compdata.1
<code>change.zero</code>	tolerance

---

`testfordiff.ind.pval` *Test for a difference between two independent samples of compositional data. Zeros of any type are allowed.*

---

**Description**

Test for a difference between two independent samples of compositional data. Zeros of any type are allowed.

**Usage**

```
testfordiff.ind.pval(compdata.1, compdata.2, R = 500)
```

**Arguments**

<code>compdata.1</code>	sample of compositional data.
<code>compdata.2</code>	sample of compositional data.
<code>R</code>	number of bootstrap samples, default is 500.

**Value**

p-value obtained through a multivariate permutation test with test statistic based on chi-square distances.

## References

Stewart, C., Iverson, S. and Field, C. (2014) Testing for a change in diet using fatty acid signatures. *Environmental and Ecological Statistics* 21, pp. 775-792.

## Examples

```
## Prey
data(prexFAs)

## Capelin FA sig
capelin.sig=prexFAs[prexFAs$Species=="capelin",4:(ncol(prexFAs))]
capelin.sig=capelin.sig/apply(capelin.sig,1,sum)

## Sandlance FA sig
sandlance.sig=prexFAs[prexFAs$Species=="sandlance",4:(ncol(prexFAs))]
sandlance.sig=sandlance.sig/apply(sandlance.sig,1,sum)

# Note:
# # Uncomment examples to run. CRAN tests fail because execution time > 5 seconds
# testfordiff.ind.pval(as.matrix(capelin.sig),as.matrix(sandlance.sig))
```

---

unbal.diet.data	<i>Sample example of unbalanced repeatability diet estimates data with a max of two repeated measurements per predator.</i>
-----------------	---

---

## Description

Sample example of unbalanced repeatability diet estimates data with a max of two repeated measurements per predator.

## Usage

```
unbal.diet.data
```

## Format

A data frame with 96 predator diets (50 unique predators) and 13 variables:

**Seal.ID** Predator (1 to 50)

**Year** Either 1 or 2

**capelin** estimated diet proportion

**coho** estimated diet proportion

**eulachon** estimated diet proportion

**herring** estimated diet proportion

**mackerel** estimated diet proportion

**pilchard** estimated diet proportion  
**pollock** estimated diet proportion  
**sandlance** estimated diet proportion  
**squid** estimated diet proportion  
**surfsmelt\_s** estimated diet proportion  
**sufsmelt\_lg** estimated diet proportion

# Index

## \* datasets

- bal.diet.data, 7
- CC, 8
- FAs<sub>set</sub>, 15
- predatorFAs, 30
- preyFAs, 34
- unbal.diet.data, 42

- AIT.dist, 3
- AIT.more, 3
- AIT.obj, 4

- backward.elimination, 4
- bal.diet.data, 7

- CC, 8
- chisq.CA, 8
- chisq.dist, 9
- comp.rep, 9
- conf.meth, 11
- create.d.mat, 13
- CS.more, 14
- CS.obj, 14

- FAs<sub>set</sub>, 15
- forward.selection, 15

- gen.pseudo.seals, 12

- KL.dist, 18
- KL.more, 18
- KL.obj, 19

- mean\_geometric, 20
- MEANmeth, 19
- multiplicativeReplacement, 20

- p.MLE, 21
- p.MUFASA, 22
- p.QFASA, 12, 24, 33
- p.sim.QFASA, 26

- p.SMUFASA, 28
- POOLVARmeth, 29
- predatorFAs, 30
- prey.cluster, 32
- prey.on.prey, 33
- preyFAs, 34
- pseudo.pred, 37
- pseudo.pred.norm, 38
- pseudo.seal, 12

- QFASA, 39
- QFASA-package (QFASA), 39
- QFASA.const.eqn, 39

- split\_pre, 40

- testfordiff.ind.boot, 40
- testfordiff.ind.boot.fun, 41
- testfordiff.ind.pval, 41

- unbal.diet.data, 42