

# Package ‘UpScaling’

May 27, 2015

**Type** Package

**Title** Upscaling species richness

**Version** 1.0

**Date** 2015-02-25

**Author** Yoni Gavish <gavishyoni@gmail.com>

**Maintainer** Yoni Gavish <gavishyoni@gmail.com>

**Depends** raster

**Suggests**

**Imports** SDMTTools, fields, nlme, minpack.lm, AICcmodavg, gmp, vegan

**Description** Various methods for predicting the number of species in large extents from limited information at smaller extents.

**License** GPL-2

## R topics documented:

UpScaling-package . . . . .	1
OptimNestedSac . . . . .	3
PlotOnePath . . . . .	8
PredictNestedSac . . . . .	9
PrepareNestedSacData . . . . .	11
RandomNestedSacData . . . . .	14
RandPAData . . . . .	15
ShenHe08Model . . . . .	16
SimplMaxEntropyHarte . . . . .	18
ToSpCurve . . . . .	20
<b>Index</b>	<b>23</b>

---

UpScaling-package	<i>UpScaling: Predicting number of species in larger extents.</i>
-------------------	---

---

## Description

This package provide a platform for aggregating various methods to predict the number of species in a large extent from a limited information on the number of species at a smaller extent.

Package: UpScaling  
 Type: Package  
 Version: 1.0  
 Date: 2015-05-12  
 License: GPL-2

### The methods covered in this package

Name	Source	Required input data	Main functions
Simplified Maximum Entropy	Harte et al. (2009)	<ul style="list-style-type: none"> <li>• Mean number of individuals per species</li> <li>• Mean number of species per sample</li> </ul>	• <code>SimplMaxEntropyHarte</code>
Shen and HE (2008)	Shen and HE (2008)	<ul style="list-style-type: none"> <li>• Presence/absence data</li> </ul>	• <code>ShenHe08Model</code>
T-S Curve	Ugland et al. (2003)	<ul style="list-style-type: none"> <li>• Presence/absence data</li> <li>• Habitat identity of each sample (recommended)</li> </ul>	• <code>ToSpCurve</code>
Nested SAC	Gavish et al. (unpublished)	<ul style="list-style-type: none"> <li>• Presence/absence data</li> <li>• Mean number of species per sample</li> <li>• X and Y coordinates of each sample</li> <li>• Ascii map of the entire extent</li> </ul>	<ul style="list-style-type: none"> <li>• <code>PrepareNestedSacData</code></li> <li>• <code>PlotOnePath</code></li> <li>• <code>OptimNestedSac</code></li> <li>• <code>predict.NestedSac</code></li> </ul>

### Credits

This package was created as part of deliverable D3.1 of WP3 of the project:

**EU-BON: Building the European Biodiversity Observation Network –**

a 7th Framework Programme funded by the European Union under Contract No. 308454.

### Special thanks

The functions [ToSpCurve](#) and [ShenHe08Model](#) are based on original codes written by Han Xu (see details in Xu et al. 2012).

### Author(s)

Written and maintained by: Yoni Gavish <gavishyoni@gmail.com>

For reporting bugs, please use 'UpScaling-bug:' in the subject line. For any communication regarding this package, please include 'UpScaling' in the subject line.

### References

1. **Harte J.**, Smith A. B., and Storch D. (2009) Biodiversity scales from plots to biomes with a universal species-area curve. *Ecology letters* 12(8): 789-797 .
2. **Shen T. J.**, and He F. (2008) An incidence-based richness estimator for quadrats sampled without replacement. *Ecology* 89(7): 2052-2060.
3. **Ugland K. I.**, Gray J. S., and Ellingsen K. E. (2003) The species-accumulation curve and estimation of species richness. *Journal of Animal Ecology* 72(5): 888-897.

4. **Xu H.**, Liu S., Li Y., Zang R., and He F.(2012) Assessing non-parametric and area-based methods for estimating regional species richness. *Journal of Vegetation Science* 23(6): 1006-1012.

OptimNestedSac

*Optimization function for the nested SAC models*

## Description

This function takes as input the values for the extent, number of samples, the observed number of species and the mean alpha diversity. Next, the function find the parameter values that minimize the sum of square differences between the observed and predicted number of species according to (up to) 22 different nested SAC models. Finally, for each model, the function predict the expected number of species if we had sampled all samples within the extent, as well as AICc, AICc weights and model averaged predictions.

## Usage

```
OptimNestedSac(object, extent, num.sample, num.spec, mean.alpha,
  sar.model = "Pow", predict.part = TRUE, predict.full = TRUE,
  mod.ave = TRUE, start.par = NULL, lower = NULL, upper = NULL,
  control = minpack.lm::nls.lm.control(), verbose = TRUE)
```

## Arguments

object	Data frame, containing the extents, number of samples and observed number of species for each case.
extent	Character or integer specifying the column name or number in object that contains the extent data.
num.sample	Character or integer specifying the column name or number in object that contains the number of samples data.
num.spec	Character or integer specifying the column name or number in object that contains the observed number of species.
mean.alpha	Positive numeric indicating the mean number of species in a single sample (alpha diversity).
sar.model	The species-area relationship model on which the nested SAC is based. Use the <b>code</b> in the following table to refer to any of the nested SAC model. The # column state the number of parameters in the model, and the following columns are starting values for each parameter.

Code	Name	#	z	b	d	n	C1	C2
Pow	Power	1	0.5	-	-	-	-	-
EP1	Extended Power 1	2	-	-0.5	2	-	-	-
EP2	Extended Power 2	2	-	0.5	0.5	-	-	-
Pe1	persistence Model 1	2	-	10	10	-	-	-
Pe2	persistence Model 2	2	-	0.5	10	-	-	-
Log	Logarithmic Model	0	-	-	-	-	-	-
Kob	Kobayashi logarithmic	1	-	0.5	-	-	-	-
NeE	Negative Exponential	1	-	0.5	-	-	-	-
CoL	Common Logistic	2	-	10	10	-	-	-
ArL	Archibald Logistic	2	-	-0.5	2	-	-	-

Gom	Gompertz	2	-	-0.5	2	-	-	-
EVF	Extreme-Value function	2	-	-0.5	2	-	-	-
Mon	Monod	1	-	1000	-	-	-	-
AsR	Asymptotic regression	1	-	-	-0.5	-	-	-
Rat	Rational Function	2	0.5	-	2	-	-	-
ChR	Chapman-Richards	2	-	0.5	2	-	-	-
CuW	Cumulative Weibull	2	-	0.5	-0.5	-	-	-
MMF	Morgan-Mercer-Flodin	2	-	10	1000	-	-	-
Lom	Lomolino Function	2	-	100	100	-	-	-
CuB	Cumulative Beta-P	3	0.5	0.5	0.5	-	-	-
Tm1	Tjorve (2012)-model 1	6	0.5	0.5	0.5	0.5	0.5	0.5
Tm2	Tjorve (2012)-model 2	6	0.5	0.5	0.5	0.5	0.5	0.5

<code>predict.part</code>	Logical, if TRUE, the predicted partial number of species is added for each <code>sar.model</code> . The partial stand for only sampling a subset of <code>num.sample</code> from the total extent.
<code>predict.full</code>	Logical, if TRUE, the predicted full number of species is added for each <code>sar.model</code> . The full stand for setting <code>num.sample = extent</code> , i.e., predicting the number of species if we had sampled the entire extent.
<code>mod.ave</code>	Logical if TRUE, A weighted mean value for the number of species (partial and full will be estimated. AICc weights will be used as weights.
<code>start.par</code>	a list containing starting parameter values for the optimization algorithm. Note- starting parameters can be provided only when calling a single <code>sar.model</code> . Otherwise default starting values are used for each model.
<code>lower</code>	A numeric vector of lower bounds on each parameter. If not given, the default lower bound for each parameter is set to <code>-Inf</code> . See details in <a href="#">nls.lm.control</a> . Note- lower bounds can be provided only when calling a single <code>sar.model</code> .
<code>upper</code>	A numeric vector of upper bounds on each parameter. If not given, the default upper bound for each parameter is set to <code>Inf</code> . See details in <a href="#">nls.lm.control</a> . Note- upper bounds can be provided only when calling a single <code>sar.model</code> .
<code>control</code>	An optional list of control settings. See <a href="#">nls.lm.control</a> for the names of the settable control values and their effect.
<code>verbose</code>	Logical, if TRUE, prints update on progress and status.

## Details

The function uses the Levenberg-Marquardt algorithm to find the parameters values that minimize the sum of square differences between the observed and predicted species richness. See details below.

## Value

the function returns a list with an object of class "nls.lm" for each species area relationship model called by `sar.model`. See [nls.lm](#) in the package `minpack.lm` for details. In addition, the function returns two data frames. The "object" data frame contains the original input data with additional columns with predicted partial and/or full number of species. The "`sar.model.full`" data frame summarizes the output of all SAR models and include columns with the code and name of the model, the values of the model parameters (`z`, `b`, `d`, `n`, `C1` and `C2`), the deviance between the observed and predicted richness, the log likelihood, AICc values, delta AICc, the likelihood and AICc weight.

## Nested SAC models

Eq. 1: $E(S a, A) = \sum_{k=1}^{O(S A)} \left[ 1 - \frac{\binom{A-n_k}}{\binom{A}} \right] = \sum_{k=1}^A \left[ 1 - \frac{\binom{A-k}}{\binom{A}} \right] \cdot R_k$	$O(S a, A)$	The observed number of species in <b>a</b> samples in extent <b>A</b> .
Eq. 2: $(\hat{\delta} k, a, A) = \left[ 1 - \frac{\binom{A-k}}{\binom{A}} \right]$	$O(S A)$	The observed number of species in extent <b>A</b> , if all the area is sampled.
Eq. 3: $E(S a, A) = \sum_{k=1}^{O(S A)} [(\hat{\delta} n_k, a, A)] = \sum_{k=1}^A [(\hat{\delta} k, a, A) \cdot R_k]$	$E(S a, A)$	The expected number of species in <b>a</b> samples in extent <b>A</b> .
Eq. 4: $R_{k,par} = E(S A - k + 1, par) - E(S A - k, par)$	$E(S A, par)$	The expected number of species in extent <b>A</b> , given the parameters ( <b>par</b> ) of the SAR model
Eq. 5: $E(S a, A, par) = \sum_{k=1}^A [(\hat{\delta} k, a, A) \cdot [E(S A - k + 1, par) - E(S A - k, par)]]$	$E(S a, A, par)$	The expected number of species in <b>a</b> samples within extent <b>A</b> , given the parameters ( <b>par</b> ) of the SAR model
Eq. 6: $1 = (\hat{\delta} \cdot A) / \sum_{k=1}^A (k \cdot R_{k,par})$	$n_s$	The number of samples in which species <b>s</b> occur if the entire extent <b>A</b> is sampled
Eq. 7: $E(S a, A, par) = (\hat{\delta} \cdot A) \cdot \sum_{k=1}^A \left[ (\hat{\delta} k, a, A) \cdot \frac{[E(S A - k + 1, par) - E(S A - k, par)]}{\sum_{k=1}^A [k \cdot [E(S A - k + 1, par) - E(S A - k, par)]]} \right]$	$R_k$	The number of species that occur in exactly <b>k</b> samples if the entire extent <b>A</b> is sampled
Eq. 8: $E(S a, A, par) = (\hat{\delta} \cdot A) \cdot \sum_{k=1}^A \left[ (\hat{\delta} k, a, A) \cdot \frac{R_{k,par}}{\sum_{k=1}^A [k \cdot R_{k,par}]} \right]$	$R_{k,par}$	The number of species that occur in exactly <b>k</b> samples if the entire extent <b>A</b> is sampled, given the SAR model parameters ( <b>par</b> )
	$\hat{\alpha}$	Alpha diversity- the mean number of species per sample

In general, the expected number of species in a set of **a** samples taken within an extent **A** (with the units being the area of a single sample) is given by eq.1 above (see Ugland et al. (2003)), which is further simplified to eq.3 by the definition in eq. 2.

In an ideal research, **O(S|A)** as well as the shape of the species occupancy distribution (SOD, the values of all **R<sub>k</sub>**) are known. In such cases, **E(S|a, A)** is the expected number of species under random sampling of exactly **a** samples from the **A** available ones. Unfortunately, in most cases, we have a known extent (**A**), a set of samples (**a**) and the observed number of species that were recorded in the set of samples (**O(S|a, A)**), while **O(S|A)** and **R<sub>k</sub>** are unknown.

Therefore, the challenge is to reverse the usage of eq.3 and thereby estimate the values of **O(S|A)** and all **R<sub>k</sub>**. To do so, we assume that **O(S|A)** changes in a predicted manner with area (i.e., species area relationship- SAR) and that knowledge of this change also provide knowledge on the values of **R<sub>k</sub>**. From the various types of SARs, nested SAR provide the most promising direction since **R<sub>k</sub>** can be estimated directly from the SAR equation. More specifically, if **E(S|A, par)** is the expected number of species in the entire extent **A** according to a nested SAR model with parameters **par**, then **R<sub>k, par</sub>** is given in eq.4. Plugging eq.4 into eq.3 results with eq.5.

Next, we are adding a constrain such that when **a=1**, the nested SAC model will predict the mean alpha diversity (mean number of species per sample). To do so we force the total number of occupancies according to the SAR to equal the mean alpha diversity multiplied by the number of samples in the extent (eq.6). Then we multiply eq.5 with eq.6 and insert the denominator of eq.6 into the summation of eq.5. The resulting eq.7 and eq. 8 are the general form of the nested SAC model. In the table below, the specific terms of **R<sub>k, par</sub>** are given for each nested SAC model.

In most nested SAC models, one parameter from the original SAR equation is canceled out due to the constrain (see Tjorve (2009) for description of the original SAR equations). In the logarithmic model, both parameters are canceled out, while no parameter is canceled out in Tjorve (2012) model 1 and model 2.

This function finds the values of **par** that maximizes the fit of eq. 7 to **O(S|a, A)** by minimizing the residuals (**O(S|a, A) - E(S|a, A, par)**) sum of squares. In other words, the model estimate the parameters of a SAR function that describe the change of the 'true' number of species with area, by acknowledging the fact that our sampling is incomplete. After the parameters of the SAR are found,

the expected 'true' number of species can be found by setting  $a=A$  in eq. 7, and thereby collapsing the equation to  $E(S|A,par)$ .

Function	Code	$R_{k,par}$
Power	Pow	$(A - k + 1)^z - (A - k)^z$
Extended power 1	EP1	$(A - k + 1)^{b \cdot (A - k + 1)^d} - (A - k)^{b \cdot (A - k)^d}$
Extended power 2	EP2	$(A - k + 1)^{b \cdot (d / (A - k + 1))} - (A - k)^{b \cdot (d / (A - k))}$
Persistence model 1	Pe1	$(A - k + 1)^b \cdot \exp(-d \cdot (A - k + 1)) - (A - k)^b \cdot \exp(-d \cdot (A - k))$
Persistence model 2	Pe2	$(A - k + 1)^b \cdot \exp\left(\frac{-d}{(A - k + 1)}\right) - (A - k)^b \cdot \exp\left(\frac{-d}{(A - k)}\right)$
Logarithmic model	Log	$\log(A - k + 1) - \log(A - k)$
Kobayashi logarithmic	Kob	$\log\left(1 + \frac{(A - k + 1)}{b}\right) - \log\left(1 + \frac{(A - k)}{b}\right)$
Negative exponential	NeE	$\exp(-b \cdot (A - k)) - \exp(-b \cdot (A - k + 1))$
Common logistic	Col	$\frac{1}{(1 + \exp(-b \cdot (A - k + 1) + d))} - \frac{1}{(1 + \exp(-b \cdot (A - k) + d))}$
Archibald logistic	ArL	$\frac{1}{(b + d^{A - k + 1})} - \frac{1}{(b + d^{A - k})}$
Gompertz	Gom	$\exp(-\exp(-b \cdot (A - k + 1) + d)) - \exp(-\exp(-b \cdot (A - k) + d))$
Extreme-value function	EVF	$[\exp(-\exp(b \cdot (A - k) + d)) - \exp(-\exp(b \cdot (A - k + 1) + d))]$
Monod	Mon	$\frac{(A - k + 1)}{(b + A - k + 1)} - \frac{(A - k)}{(b + A - k)}$
Asymptotic regression	AsR	$d^{-(A - k)} - d^{-(A - k + 1)}$
Rational function	Rat	$\frac{(1 + z \cdot (A - k + 1))}{(1 + d \cdot (A - k + 1))} - \frac{(1 + z \cdot (A - k))}{(1 + d \cdot (A - k))}$
Chapman-Richards	ChR	$[1 - \exp(-b \cdot (A - k + 1))]^d - [1 - \exp(-b \cdot (A - k))]^d$
Cumulative Weibull	CuW	$\exp(-b \cdot (A - k)^d) - \exp(-b \cdot (A - k + 1)^d)$
Morgan-Mercer-Flodin	MMF	$\frac{(A - k + 1)^b}{(d + (A - k + 1)^b)} - \frac{(A - k)^b}{(d + (A - k)^b)}$
Lomolino function	Lom	$\frac{1}{(1 + b \log(d / (A - k + 1)))} - \frac{1}{(1 + b \log(d / (A - k)))}$
Cumulative beta-P	CuB	$[1 + ((A - k) / b)^z]^{-d} - [1 + ((A - k + 1) / b)^z]^{-d}$
Tjorve (2012)-model 1	Tm1	$\left[ [C_1 + b \cdot \log_{10}(A - k + 1)]^{\frac{(A - k + 1)}{(A - k + 1 + n)}} \cdot [C_2 \cdot (A - k + 1)^z]^{1 - \frac{(A - k + 1)}{(A - k + 1 + n)}} \right] - \left[ [C_1 + b \cdot \log_{10}(A - k)]^{\frac{(A - k)}{(A - k + n)}} \cdot [C_2 \cdot (A - k)^z]^{1 - \frac{(A - k)}{(A - k + n)}} \right]$
Tjorve (2012)-model 2	Tm2	$\left[ [C_2 \cdot (A - k + 1)^z]^{\frac{(A - k + 1)}{(A - k + 1 + n)}} \cdot [C_1 + b \cdot \log_{10}(A - k + 1)]^{1 - \frac{(A - k + 1)}{(A - k + 1 + n)}} \right] - \left[ [C_2 \cdot (A - k)^z]^{\frac{(A - k)}{(A - k + n)}} \cdot [C_1 + b \cdot \log_{10}(A - k)]^{1 - \frac{(A - k)}{(A - k + n)}} \right]$

## Author(s)

Yoni Gavish <gavishyoni@gmail.com>

## References

1. **Tjorve E.** (2009) Shapes and functions of species-area curves (II): a review of new models and parameterizations. *Journal of Biogeography* 36(8): 1435-1445.
2. **Tjorve E.** (2012). Arrhenius and Gleason revisited: new hybrid models resolve an old controversy. *Journal of Biogeography* 39(4): 629-639.
3. **Ugland K. I., Gray J. S., and Ellingsen K. E.** (2003) The species-accumulation curve and estimation of species richness. *Journal of Animal Ecology* 72(5): 888-897.

## See Also

[RandomNestedSacData](#) for creating random data to explore this function, [PlotOnePath](#) for plotting a single path, [PrepareNestedSacData](#) for preparing data for a nested SAC analysis , and [PredictNestedSac](#) for predicting the expected number of species according to a given nested SAC model and paramteres values.

## Examples

```
set.seed(350)
# create random data
rand.input <- RandomNestedSacData(num.col      = 15, num.row      = 20,
                                num.species = 95, num.sites   = 30)

Nested.Sac.Data <- PrepareNestedSacData(PA.data      = rand.input$PA.data,
                                       asc.in       = rand.input$asc.out)

# the mean alpha diversity and the data from the path
PA.data <- Nested.Sac.Data$PA.data[, -(1:5)]
mean.alpha <- sum(PA.data)/ 30
nested.SAC.object <- Nested.Sac.Data$nested.SAC.object
optim.out <- OptimNestedSac(object = nested.SAC.object,
                           extent = 2, num.sample = 1, num.spec = 3,
                           mean.alpha = mean.alpha,
                           sar.model = c("Pow", "CuB"),
                           control = minpack.lm::nls.lm.control(
                               maxiter = 1000))

# information on each model
sar.model.out <- optim.out$sar.model.full
sar.model.out[,c("code", "AICcWt")]

# Predicted species richness + plotting
sar.object <- optim.out$object

par(mfrow = c(2,2), mar=c(5.1, 4.1, 4.1, 2.1))
## 1 Pow num.sample
plot(sar.object$num.sample, sar.object$num.spec, pch= 17, col="black",
     xlab="num.sample", ylab = "species", ylim = c(0,150), main = "Pow")
lines(sar.object$num.sample, sar.object$part.Pow, col="blue")
lines(sar.object$num.sample, sar.object$full.Pow, col="red")

## 2 Pow extent
plot(sar.object$extent, sar.object$num.spec, pch= 17, col="black",
     xlab="extent", ylab = "species", ylim = c(0,150), main = "Pow")
lines(sar.object$extent, sar.object$part.Pow, col="blue")
lines(sar.object$extent, sar.object$full.Pow, col="red")

## 3 CuB num.sample
plot(sar.object$num.sample, sar.object$num.spec, pch= 17, col="black",
     xlab="num.sample", ylab = "species", ylim = c(0,150), main = "CuB")
lines(sar.object$num.sample, sar.object$part.CuB, col="blue")
lines(sar.object$num.sample, sar.object$full.CuB, col="red")

## 4 CuB extent
plot(sar.object$extent, sar.object$num.spec, pch= 17, col="black",
     xlab="extent", ylab = "species", ylim = c(0,150), main = "CuB")
lines(sar.object$extent, sar.object$part.CuB, col="blue")
```



```
lines(sar.object$extent, sar.object$full.CuB, col="red")
```

---

PlotOnePath

*Plots a path several cells at a time*


---

## Description

This function takes as input the raster, cell.info and a single path (from [PrepareNestedSacData](#)). Then, the points are added several at a time according to the path.

## Usage

```
PlotOnePath(raster.out, cell.info, path.cell.num, in.col = "gray70",
  out.col = "darkblue", sampled.col = "yellow", unsampled.col = "black",
  start.col.sam = "darkorchid1", start.col.unsam = "red", pch = 16,
  cex = 0.5, pause.sec = 0.8,
  every.n.cells = round(length(path.cell.num)/20, 0))
```

## Arguments

raster.out	The raster object from <a href="#">PrepareNestedSacData</a> .
cell.info	The cell.info data frame from <a href="#">PrepareNestedSacData</a> .
path.cell.num	A vector containing a single path from a starting cell to the last cell. Numbers in vector should refer to the internal cells number of the raster (valid and invalid cells) - i.e., to 'cell.num' of cell.info.
in.col	The color to use in the raster for valid cells.
out.col	The color to use in the raster for invalid cells (cells with NA for cell.ID).
sampled.col	The color to represent sampled cells when encountered along the path.
unsampled.col	The color to represent unsampled cells when encountered along the path.
start.col.sam	The color to represent the starting cell when it is at a sampled site.
start.col.unsam	The color to represent the starting cell when it is not at a sampled site.
pch	plotting character.
cex	numerical, relative size of the symbols.
pause.sec	The length of time in second between the addition of consecutive points to the plot.
every.n.cells	positive integer. The number of cells to be plotted in one step.

## Details

May take considerable time if the raster is big and every.n.cells is small. In addition, for large maps use small cex values.

## Value

A plot



**Author(s)**

Yoni Gavish <gavishyoni@gmail.com>

**Examples**

```
set.seed(350)
# create random data
rand.input <- RandomNestedSacData(num.col      = 15, num.row      = 20,
                                  num.species = 95, num.sites   = 30)

# note the differences between the 3 path growing methods in spread and holes
# starting from the same cell

# The 'min.dist.start' path growing method - very clumped, no holes
set.seed(250)
Nested.Sac.Data <- PrepareNestedSacData(PA.data      = rand.input$PA.data,
                                         asc.in       = rand.input$asc.out,
                                         grow.method = "min.dist.start")

# Plot a single path
PlotOnePath(raster.out      = Nested.Sac.Data$raster.out,
            cell.info       = Nested.Sac.Data$cell.info,
            path.cell.num    = Nested.Sac.Data$path.info$cell.num,
            cex = 1, every.n.cells = 15)

# The 'random.adj.4' path growing method - more dispersed, some holes
set.seed(250)
Nested.Sac.Data.2 <- PrepareNestedSacData(PA.data      = rand.input$PA.data,
                                         asc.in       = rand.input$asc.out,
                                         grow.method = "random.adj.4")

PlotOnePath(raster.out      = Nested.Sac.Data.2$raster.out,
            cell.info       = Nested.Sac.Data.2$cell.info,
            path.cell.num    = Nested.Sac.Data.2$path.info$cell.num,
            cex = 1, every.n.cells = 15)

# The 'random.adj.8' path growing method - very dispersed, many holes
set.seed(250)
Nested.Sac.Data.2 <- PrepareNestedSacData(PA.data      = rand.input$PA.data,
                                         asc.in       = rand.input$asc.out,
                                         grow.method = "random.adj.8")

PlotOnePath(raster.out      = Nested.Sac.Data.2$raster.out,
            cell.info       = Nested.Sac.Data.2$cell.info,
            path.cell.num    = Nested.Sac.Data.2$path.info$cell.num,
            cex = 1, every.n.cells = 15)
```

---

PredictNestedSac

---

*Predict the number of species according to a nested SAC model*


---

**Description**

This function predicts the expected number of species for a given extent and number of samples, given a user defined SAR model and parameters values. See [OptimNestedSac](#) for additional details on the parameters of each nested SAC model

**Usage**

```
PredictNestedSac(object, extent, num.sample, mean.alpha, sar.model,
  par = NULL, verbose = TRUE)
```

**Arguments**

object	Data frame, containing the extents, number of samples and observed number of species for each case.
extent	Character or integer specifying the column name or number in object that contains the extent data.
num.sample	Character or integer specifying the column name or number in object that contains the number of samples data.
mean.alpha	Positive numeric indicating the mean number of species in a single sample (alpha diversity).
sar.model	The species-area relationship model on which the nested SAC is based. Should contain only 1 nested SAC model. See <a href="#">OptimNestedSac</a> for additional details on the various nested SAC models.
par	List containing the values of all parameters of the specified nested-SAC model.
verbose	Logical, if TRUE, prints update on progress and status.

**Details**

When `extent != num.sample`, the 'partial' number of species is predicted. When `extent == num.sample` the function predict the number of species if we had sampled the entire extent (the 'full' number of species)

**Value**

The function returns the input object with an additional column (`pred.spe.num`) containing the predicted number of species.

**Author(s)**

Yoni Gavish <gavishyoni@gmail.com>

**Examples**

```
set.seed(350)
# create random data
rand.input <- RandomNestedSacData(num.col = 10, num.row = 10,
  num.species = 50, num.sites = 25)
Nested.Sac.Data <- PrepareNestedSacData(PA.data = rand.input$PA.data,
  asc.in = rand.input$asc.out)

# the mean alpha diversity and the data from the path
mean.alpha <- sum(Nested.Sac.Data$PA.data[, -(1:5)]) / 25
nested.SAC.object <- Nested.Sac.Data$nested.SAC.object

# Power model - incomplete sampling with z = 0.9
# different column for extent and number of samples
predcit.Pow <- PredictNestedSac(object = nested.SAC.object, extent = 2,
  num.sample = 1, mean.alpha = mean.alpha,
```

```

sar.model = "Pow", par= list(z = 0.9))

# Power model - complete sampling with z = 0.9
# The same column for extent and number of samples
predcit.Pow.2<- PredictNestedSac(object = nested.SAC.object, extent = 2,
                                num.sample = 2, mean.alpha = mean.alpha,
                                sar.model = "Pow", par= list(z = 0.9))

# model with more then 1 parameters
predcit.EP2<- PredictNestedSac(object = nested.SAC.object, extent = 2,
                                num.sample = 1, mean.alpha = mean.alpha,
                                sar.model = "EP2", par= list(b = 0.9, d = 0.5))

```

---

PrepareNestedSacData    *Prepares the data for the nested SAC model*

---

## Description

The nested SAC models takes as input a data frame with the number of cells in the entire extent, the number of sampled cells, and the number of species observed in the set of sampled cells (the extent, num.sample and num.spec in [OptimNestedSac](#), respectively) . This function aims to create the data for the nested SAC model. It takes as input the presence/absence data and an ascii file, creates a raster from the ascii, and link each sampled site in the presence/absence table with a cell in the raster object. Next, the function creates paths that start with a random or user defined starting cell and adds adjacent cells until all valid cells are covered. Along this path, information on the extent, number of sampled sites and number of species is kept and returned to the user for further analysis with [OptimNestedSac](#). In addition, all information on the path itself is returned for further analysis and/or visualization.

## Usage

```

PrepareNestedSacData(PA.data, asc.in, num.paths = 1, start.cell = NULL,
                     grow.method = "min.dist.start", verbose = TRUE)

```

## Arguments

PA.data	data frame with the x coordinate, y coordinate, the site ID and the presence/absence data (species as columns, sites as rows) at this order. Presences and absences should be labeled with 1 and 0, respectively. All sampled cells should fall within valid cells in the ascii, and no ascii cell should contain more then 1 sampled site. See the output of <a href="#">RandomNestedSacData</a> \$PA.data for an example.
asc.in	An Ascii file with the study area. the ascii should have values of 1 for all sampled and unsampled sites that will be included in the analysis (valid cells) and 0 values for sites located outside the area of interest (invalid cells).
num.paths	Positive integer specifying the number of paths that will be grown.
start.cell	data frame with a column for x coordinate and a column for y coordinate specifying the starting location of each path (in this order). The number of rows must equal num.paths. If NULL (default), a random starting valid cell will be chosen.
grow.method	Character, the method by which the path is grown. all methods start from start.cell and add randomly chosen cells from a list of adjacent cells until the entire list of valid cells is covered. The methods differ in the creation of list of adjacent cells at each step. Currently the following options are supported:

- 'min.dist.start' -  
Always select a cell from the list of cells with minimum distance from the starting cell. This method results with the most clumped path that grows outwards from the starting cell. The method does not results with 'holes' in the distribution of travelled cells.
- 'random.adj.4' -  
The list of adjacent cells contain all the cells adjacent to all cells that were already travelled. Adjacent cells are based on 4 neighbouring cells. A random cell is selected from the list of adjacent cells, thereafter, the list is updated. The procedure continues until all cells are added. The method may result with 'holes' in the distribution of travelled cells. See the rook (or 4) option in [adjacent](#) in the raster package for details on adjacency.
- 'random.adj.8' -  
Same as above, only with 8 neighbouring cells. See the queen (or 8) option in [adjacent](#) in the raster package for details. This method results with a very dispersed path that may contain 'holes' in the distribution.

verbose                      Logical, if TRUE, prints update on progress and status.

## Value

The function returns a list with the following objects:

- 'PA.data' -  
The input PA.data data frame with two additional columns- 'cell.num' contains the number of the cell in the raster (both valid and invalid cells) and 'cell.ID' with the unique ID of each valid cell in the raster. In valid cells are listed as NA.
- 'raster.out' -  
An object of class 'raster', with 'cell.ID' as values.
- 'cell.info' -  
A data frame with information on each cell in the raster object. The data frame contain the 'cell.num' and 'cell.ID' as above, along with additional column that specifies if a cell was sampled or not ('sampled' - 1 and NA respectively), two columns with the x and y coordinates and a column with the site ID as specified by the user in PA.data.

In addition to these objects the list contain additional data frames according to the value of num.paths.

**For num.paths == 1, the list contains two data frame:**

- 'path.info' -  
Data frame with information on the order of cells in the path, including all the columns of cell.info except the 'sampled' column.
- 'nested.SAC.object' -  
Data frame that can be used as input object to [OptimNestedSac](#), with a column for the number of samples, the extent in which the path reached each new sample and the number of species in the set of samples that were already covered.

**For num.paths > 1, the list contains the following data frames:**

- 'path.cell.num' -  
Data frame with the 'cell.num' of each cell (rows) along each path (columns).
- 'path.cell.ID' -  
Data frame with the 'cell.ID' of each cell (rows) along each path (columns).
- 'path.cell.X' -  
Data frame with the X coordinate of each cell (rows) along each path (columns).

- 'path.cell.Y' -  
Data frame with the Y coordinate of each cell (rows) along each path (columns).
- 'path.cell.site.ID' -  
Data frame with the site ID (from the original PA.data) of each cell (rows) along each path (columns).
- 'path.extent' -  
Data frame with the extent at which each new sample was reached in each path. The first column contains the number of samples (num.samples) followed by a column for each path.
- 'path.num.spec' -  
Same as 'path.extent' only with the number of species in the set of samples already encountered along the path.

### Note

For big ascii, creating many paths may result with memory limits issues.

### Author(s)

Yoni Gavish <gavishyoni@gmail.com>

### See Also

[RandomNestedSacData](#) for creating random data to explore this function, [PlotOnePath](#) for plotting a single path, [OptimNestedSac](#) for fitting the nested SAC models, and [PredictNestedSac](#) for predicting the expected number of species according to a given nested SAC model.

### Examples

```
set.seed(350)
# create random data
rand.input <- RandomNestedSacData(num.col      = 15, num.row      = 20,
                                  num.species = 95, num.sites   = 30)

# Example with a single path
Nested.Sac.Data <- PrepareNestedSacData(PA.data      = rand.input$PA.data,
                                       asc.in        = rand.input$asc.out)
names(Nested.Sac.Data)

# note the additional columns in PA.data
names(Nested.Sac.Data$PA.data)[1:10]

par(mfrow = c(2,2))
# plot the raster, all the valid and invalid cells, and the sampled cells
plot(Nested.Sac.Data$raster.out, colNA = "gray")
points(Nested.Sac.Data$cell.info$X.cord, Nested.Sac.Data$cell.info$Y.cord)
points(Nested.Sac.Data$PA.data$X.cord, Nested.Sac.Data$PA.data$Y.cord,
       pch=16, col="red")

nested.SAC.object <- Nested.Sac.Data$nested.SAC.object
# the extent t which each sample was added to the path
plot(nested.SAC.object$num.sample, nested.SAC.object$extent,
     xlab = "Number of samples", ylab= "Extent")

# the accumulation of species with extent and number of samples
plot(nested.SAC.object$extent, nested.SAC.object$num.spec,
```

```

      xlab = "Extent", ylab = "Number of species")
plot(nested.SAC.object$num.sample, nested.SAC.object$num.spec,
      xlab = "Number of samples", ylab= "Number of species")

# example with several paths
Nested.Sac.Data <- PrepareNestedSacData(PA.data      = rand.input$PA.data,
                                       asc.in       = rand.input$asc.out,
                                       num.paths    = 5)

names(Nested.Sac.Data)

```

---

RandomNestedSacData     *Create a random ascii file and presence absence data*

---

## Description

The function creates an ascii with user defined dimensions. Each cell in the ascii file is either valid (1) or invalid (0). All valid cells are connected to at-least one other valid cell. Next, the function creates a random presence/absence data frame, using the function [RandPAData](#) and assigns each sampled site to a valid cell. The number of sampled site can be lower than the number of valid cells, and no valid cell contain more than 1 sampled site. The package returns the presence/absence table and the ascii file along with a raster object and some information on each cell of the raster.

## Usage

```
RandomNestedSacData(num.col, num.row, num.species, num.sites = NULL,
  mat.fill = NULL, verbose = TRUE)
```

## Arguments

num.col	Integer, the number of columns in the ascii file.
num.row	Integer, the number of rows in the ascii file.
num.species	Integer, the number of species in the presence/absence data frame. See <a href="#">RandPAData</a> for details.
num.sites	Integer, number of sampled sites in the presence/absence data frame, should be at least 2 and and not larger than num.col*num.row. If NULL, a random number is drawn. See <a href="#">RandPAData</a> for details.
mat.fill	Integer, number of occupancies to be placed within the presence/absence data frame. If NULL, a random number is drawn. if num.sites is NULL, mat.fill is set to NULL as well. See <a href="#">RandPAData</a> for details.
verbose	Logical, if TRUE, prints update on progress and status.

## Value

The function returns a list with the following:

- 'PA.data' -  
Data frame containing the X coordinate values, the Y coordinate values, a site ID column and the presence/absence data.
- 'asc.out' -  
ascii file with 1 for cells within the extent of interest and 0 for cells outside the extent of interest.

- 'raster.in' -  
A raster with NA for cells outside the extent of interest and a unique cell.ID value for each cell within the extent of interest.
- 'cell.info' -  
Data frame with the cell running numbers and the unique cell Id for each cell in the raster.

### Author(s)

Yoni Gavish <gavishyoni@gmail.com>

### Examples

```
set.seed(350)
# create random data
rand.input <- RandomNestedSacData(num.col      = 15,
                                  num.row      = 20,
                                  num.species   = 95,
                                  num.sites    = 30,
                                  mat.fill     = 180)

PA.data    <- rand.input$PA.data
asc.out    <- rand.input$asc.out
raster.out <- rand.input$raster.out
cell.info  <- rand.input$cell.info

# colors according to the cell ID, invalid cells in blue
plot(raster.out, colNA= "blue")
# sampled sites
points(PA.data$X.cord, PA.data$Y.cord)
```

---

RandPAData

*Random presence / absence data*

---

### Description

The function takes as input the number of species, the number of sites and the number of occupancies and returns a species (columns) by site (rows) data frame with randomly distributed presence/absence data (1,0).

### Usage

```
RandPAData(num.species, num.sites, mat.fill)
```

### Arguments

num.species	Integer, the number of species in the presence-absence table.
num.sites	Integer, the number of sites in the presence-absence table.
mat.fill	Integer, the number of occupancies to distribute in the presence-absence matrix.

### Details

The function ensures that each site or species have at least 1 occupancy. Therefore, the valid range of mat.fill is:

```
max(num.species, num.sites) <= mat.fill <= num.species * num.sites.
```



**Value**

A species (columns) by site (rows) data frame with presence/absence data.

**Author(s)**

Yoni Gavish <gavishyoni@gmail.com>

**Examples**

```
set.seed(865)
# the minimum value of mat.fill - at least one occupancy for every species/site
PA.data <- RandPAData(num.species = 40, num.sites = 50, mat.fill = 50)
min(colSums(PA.data))
min(rowSums(PA.data))

# random data
PA.data <- RandPAData(num.species = 40, num.sites = 50, mat.fill = 350)

# frequencies of col/row sums
freq.table <- data.frame(i = NA, num.spe = NA, num.site = NA)

for (i in 1:max(c(colSums(PA.data), rowSums(PA.data)))){
  freq.table[i, 1] <- i
  freq.table[i, 2] <- sum(colSums(PA.data)== i)
  freq.table[i, 3] <- sum(rowSums(PA.data)== i)}

barplot(freq.table$num.spe, names = freq.table$i,
        xlab = "number of occurrences", ylab = "number of species")
barplot(freq.table$num.site, names = freq.table$i,
        xlab = "number of occurrences", ylab = "number of site")
```

---

ShenHe08Model

*Fits Shen and He (2008) model*


---

**Description**

This function takes as input presence-absence data with species as columns and sampled sites as rows. The aim is to predict the number of species if we had sampled additional samples. Thus, the function fits Shen and He (2008) model to a vector of number of samples specified by the user.

**Usage**

```
ShenHe08Model(PA.data, num.samples, assess.var = TRUE, sample.area = NULL,
              verbose = TRUE)
```

**Arguments**

PA.data	Data frame with the species (columns) by samples (rows) presence/absence (1,0) data.
num.samples	Numeric vector with the number of samples for which the Shen and He model should be fitted. All values in the vector should be equal or larger to the number of sampled sites (number of rows in the PA.data data frame).



```

                                seq(2000, 10000, 1000)),
                                sample.area = 20)
# explore the results
exp.spec <- ShenHeResults$expected.spec
plot(log10(exp.spec$area),
     exp.spec$expec.spe,
     xlab = "Log10(Area)",
     ylab = "Species")

```

---

SimplMaxEntropyHarte     *Harte et al. (2009) simplified maximum entropy model*

---

## Description

This function takes as input the mean number of individual in a sample and the mean number of species per sample and uses Harte et al. (2009) simplified maximum entropy model to estimate the expected number of specie in consecutive doubling of the area.

## Usage

```

SimplMaxEntropyHarte(ind.per.sam, spe.per.sam, num.doubling = 10,
  lam.start = "auto", method = "Nelder-Mead", lower = -Inf, upper = Inf,
  verbose = TRUE)

```

## Arguments

ind.per.sam	Positive numeric, the mean number of individuals in a single sample.
spe.per.sam	Positive numeric, the mean number of species in a single sample.
num.doubling	Positive integer. the number of times the area should be doubled. Note that too large values will take very long time to process. May reach memory limits at around num.doubling = 22.
lam.start	Default of 'auto' means that starting values for the lambda parameter are chosen automatically. For the first doubling, lam.start is set to $(\text{spe.per.sam} / \text{ind.per.sam})^{1.8}$ . For all other doubling, the fitted lambda of the former iteration is used as the starting value for the current iteration. Alternatively, the user can provide a vector of starting values (vector length equal the length of num.doubling).
method	The optimization method used. See <a href="#">optim</a> for details. Note that the default method of "Nelder-Mead" returns a warning, yet seems to converge well. The suggested method for one-dimensional optimization ("Brent") requires finite lower and upper bound for lambda.
lower	The lower bound for lambda when using the optim function.
upper	The upper bound for lambda when using the optim function.
verbose	Logical, if TRUE, prints update on progress and status.

## Details

The estimated number of species at each doubling of the area is calculated through an iterative process. In each iteration step, the species richness and number of individuals from the former step are used as 'anchor'. Then equations 8 and 9 from Harte et al. (2009) are solved for the two parameters - lambda and the number of species.

**Value**

The function returns a data frame with the following:

- 'area' -  
The area in units of a single sample.
- 'num.ind' -  
The mean number of individuals per sample at a given scale.
- 'num.species' -  
The predicted number of species at a given scale.
- 'lam.fitted' -  
The fitted value of lambda at a given scale.
- 'lam.in' -  
The starting value of lambda used in the optimization at a given scale.

**Credit**

Most of this function is based on the original code written by Han Xu (see details in Xu et al. 2012).

**Author(s)**

Yoni Gavish <gavishyoni@gmail.com>

**References**

1. **Harte J.**, Smith A. B., and Storch D. (2009) Biodiversity scales from plots to biomes with a universal species-area curve. *Ecology letters* 12(8): 789-797 .
2. **Xu H.**, Liu S., Li Y., Zang R., and He F.(2012) Assessing non-parametric and area-based methods for estimating regional species richness. *Journal of Vegetation Science* 23(6): 1006-1012.

**Examples**

```
# Run the model
Harte.try <- SimplMaxEntropyHarte(ind.per.sam = 500,
                                spe.per.sam = 100,
                                num.doubling = 8)

# plot the species area relationship
plot(Harte.try$area,
     Harte.try$num.species,
     main = "Simplified maximum entropy - Harte et al. (2009)",
     xlab = "Number of samples",
     ylab = "Number of species")
lines(Harte.try$area, Harte.try$num.species)
```

ToSpCurve

*Ugland et al. (2003) Total Species Curve (T-S Curve)***Description**

This function takes as input a species by samples presence/absence data frame. The samples may be pre-allocated to different habitat categories or are allocated to a user defined number of categories using `kmeans` clustering. Then, the function applies species accumulation curves for all possible combinations of 1, 2, ..., k habitats. Next, the TS-Curve is built by fitting a model to the average total number of species in 1, 2, ..., k habitats. Here, the semi-logarithmic model,  $S = c * \ln(X) + b$ , is fitted with  $S$  being the number of species in  $X$  samples, while  $c$  and  $b$  are fitted parameters. Finally, the fitted curve is extrapolated to user defined number of samples (optional).

**Usage**

```
ToSpCurve(PA.data, new.data = NULL, habitats = NULL, num.habitat = 5,
  nstart = 100, iter.max = 100, algorithm = "Hartigan-Wong",
  verbose = TRUE)
```

**Arguments**

PA.data	Data frame with the species (columns) by samples (rows) presence/absence (1,0) data. If <code>habitats != NULL</code> , then one additional column for habitats is needed.
new.data	a numeric vector of positive integers specifying the number(s) of samples to which the TS-Curve should be extrapolated. If <code>NULL</code> (default), the model will not be extrapolated.
habitats	If the samples are pre-allocated to habitats (recommended), then an integer or character specifying the column number or name in <code>PA.data</code> that contains the habitat data. If <code>NULL</code> (default, not recommended) the samples will be first classified to <code>num.habitats</code> based on species presence/absence using <code>kmeans</code> clustering.
num.habitat	Positive integer, if <code>habitats == NULL</code> the number of habitats to which the samples will be clustered. Ignored if <code>habitats != NULL</code> .
nstart	Control parameter for <code>kmeans</code> - the number of random starting sets to be chosen. Ignored if <code>habitats != NULL</code> .
iter.max	Control parameter for <code>kmeans</code> - the maximum number of iterations allowed. Ignored if <code>habitats != NULL</code> .
algorithm	Control parameter for <code>kmeans</code> - the clustering algorithm to use. Ignored if <code>habitats != NULL</code> .
verbose	Logical, if <code>TRUE</code> , prints update on progress and status.

**Details**

At any combination of  $k$  habitat, the species accumulation curve is terminated at the minimum number of samples. For example, assume that 10, 15, 30, and 40 samples were taken in habitats  $a$ ,  $b$ ,  $c$ , and  $d$ , respectively. There are 4 different combinations of 1 habitat, and we will average the number of species found in 10 samples. For 2 habitats there are 6 possible combinations ( $a+b$ ,  $a+c$ ,  $a+d$ ,  $b+c$ ,  $b+d$ ,  $c+d$ ) and we will take the average number of species found in  $10 + 15 = 25$  samples ( $a+b$ ). For 3 habitats there are 4 combinations ( $a+b+c$ ,  $a+b+d$ ,  $a+c+d$ ,  $b+c+d$ ) and we will

take the average of  $10 + 15 + 30 = 55$  samples ( $a+b+c$ ). Finally, there is a single combination of 4 habitats ( $a+b+c+d$ ) and we will use the number of species in all 95 samples. Species accumulation curves use the function `specaccum` in the `vegan` package.

### Value

The function returns a list with the following objects:

PA.data	The input PA.data data frame. If <code>habitats == NULL</code> , one additional column is added with the assigned habitat for each site according to the <code>kmeans</code> clustering.
SAC.info	Data frame containing the information on all the species accumulation curves (rows). The data frame starts with a column for each habitat. values of TRUE or FALSE state if the habitat was included in the combination or not. These are followed by two columns, one with the number of habitats in the combination ( <code>num.habitats</code> ) and the other with the number of samples in the combination ( <code>num.samples</code> ). The remaining columns state the number of species at each step of the accumulation, according to the <code>method = "exact"</code> option in the <code>specaccum</code> function of the <code>vegan</code> package.
TS.semi.log	An object of class <code>nls</code> . The fitted semi-logarithmic TS-curve. See <code>nls</code> for details.
TS.data	The TS-curve data as a data frame with the following columns: <ul style="list-style-type: none"> <li>'data.source' - The source of the data, either 'PA.data' for points that originate from the species accumulation curves or 'new.data' for user defined points.</li> <li>'num.habitats' - For the 'PA.data', the number of habitats in the combinations.</li> <li>'num.combin' - For the 'PA.data', the number of combinations with exactly <code>num.habitats</code> habitats.</li> <li>'num.samples' - For the 'PA.data', the minimum number of samples in the combinations of <code>num.habitats</code> habitats. For 'new.data' - the number of sampled specified by the user.</li> <li>'mean.species' - For the 'PA.data', the mean number of species in a combination of <code>num.samples</code> samples from <code>num.habitats</code> habitats.</li> <li>'sd.species' - For the 'PA.data', the standard deviation around <code>mean.species</code>.</li> <li>'se.species' - For the 'PA.data', the standard error around <code>mean.species</code>.</li> <li>'predicted.sp' - The expected number of species according to the fitted TS-curve.</li> </ul>
call	the call to the function.

### Author(s)

Yoni Gavish <gavishyoni@gmail.com>

### References

1. **Ugland K. I., Gray J. S., and Ellingsen K. E. (2003)** The species-accumulation curve and estimation of species richness. *Journal of Animal Ecology* 72(5): 888-897.

**Examples**

```

set.seed(865)
# create random presence/absence data
PA.data <- RandPAData(num.species = 25,
                     num.sites   = 50,
                     mat.fill    = 300)

# create 6 habitats, labeled H1, H2, ... H6
PA.data$Habitat <- paste("H", sample(1:6, 50, replace = TRUE), sep="")

# the number(s) of samples to which the TS-Curve should be extrapolated
new.data <- c(10,
             seq(25, 500, 25),
             seq(600, 2000, 200),
             seq(2500, 5000, 500),
             seq(6000, 25000, 1000),
             seq(30000, 75000, 5000))

try.ts <- ToSpCurve(PA.data = PA.data,
                   new.data = new.data,
                   habitats = "Habitat")

class(try.ts$TS.semi.log)
TS.data <- try.ts$TS.data
plot(TS.data$num.samples,
     TS.data$predicted.sp,
     xlab = "# samples",
     ylab = "Species")

# example without pre-allocated habitats and new.data
try.ts.2 <- ToSpCurve(PA.data = PA.data[, 1:25],
                   new.data = NULL,
                   habitats = NULL,
                   num.habitat = 5)

# Note the additional column in the PA.data with the assigned habitat for
# each sample
names(try.ts.2$PA.data)

```



# Index

adjacent, [12](#)

kmeans, [20](#)

nls, [21](#)

nls.lm, [4](#)

nls.lm.control, [4](#)

optim, [18](#)

OptimNestedSac, [3](#), [9–13](#)

PlotOnePath, [7](#), [8](#), [13](#)

PredictNestedSac, [7](#), [9](#), [13](#)

PrepareNestedSacData, [7](#), [8](#), [11](#)

RandomNestedSacData, [7](#), [11](#), [13](#), [14](#)

RandPAData, [14](#), [15](#)

ShenHe08Model, [2](#), [16](#)

SimplMaxEntropyHarte, [18](#)

specaccum, [21](#)

ToSpCurve, [2](#), [20](#)

UpScaling-package, [1](#)