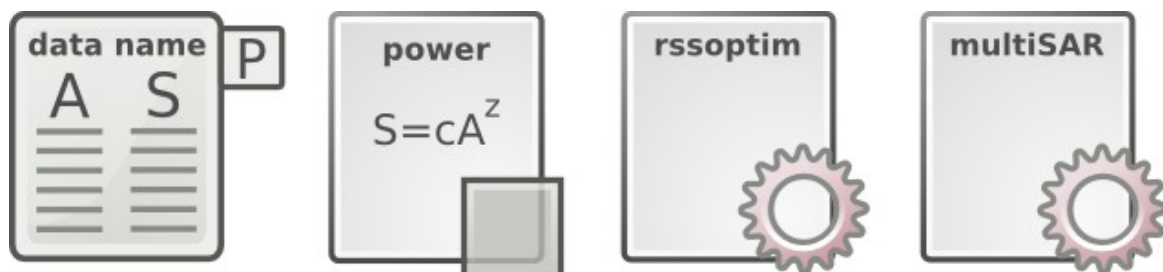


The mmSAR R package

multimodel Species-Area Relationships

mmSAR is an [R](#) package for the modelling of the Species-Area Relationship (SAR). The **mmSAR** package implements the multimodel SAR paradigm and provides ecologists with user friendly functions fit SAR models and calculate multimodel SAR inference.




Objects of the mmSAR package (data sets, models, optimization function and model averaging function).

mmSAR overview

mmSAR is a flexible and scalable implementation of the multimodel SAR framework for species-area datasets ([Guilhaumon et al. 2008](#)).

mmSAR implements the basic elements of the multimodel inference paradigm ([Burnham & Anderson, 2002](#)) with simple R list-objects and functions. The package is scalable and users can easily create their own model and data objects.

If you are new to multimodel inference or non linear SAR fitting, a good start is the [Guilhaumon et al. \(2008\)](#) paper (freely available at the PNAS website), particularly the methods section and the supplementary material .

In order to use **mmSAR**, a basic knowledge of R is expected. R and related documentations can be downloaded at <http://www.r-project.org/>.

Although **mmSAR** is user friendly, the knowledge of some concepts such as *functions* and some R objects types such as *lists* is required.

Installation

To install the **mmSAR** package directly within R :

```
install.packages("mmSAR", repos="http://R-Forge.R-project.org")
```

You need an internet connexion and administrators privileges under linux to do this.

Whenever you do not have admin rights on the system you are running R, you can install the package locally.

Assuming you own a directory located at 'myPackagesDirectoryPath', you can install the mmSAR package locally in the 'myPackagesDirectory' directory using the 'lib' argument of the 'install.packages' function :

```
install.packages("mmSAR", repos="http://R-Forge.R-project.org", lib="myPackagesDirectoryPath")
```

After, when loading the **mmSAR** package, use the 'lib.loc' argument of the 'library' function to find the package :

```
library(mmSAR, lib.loc="myPackagesDirectoryPath")
```

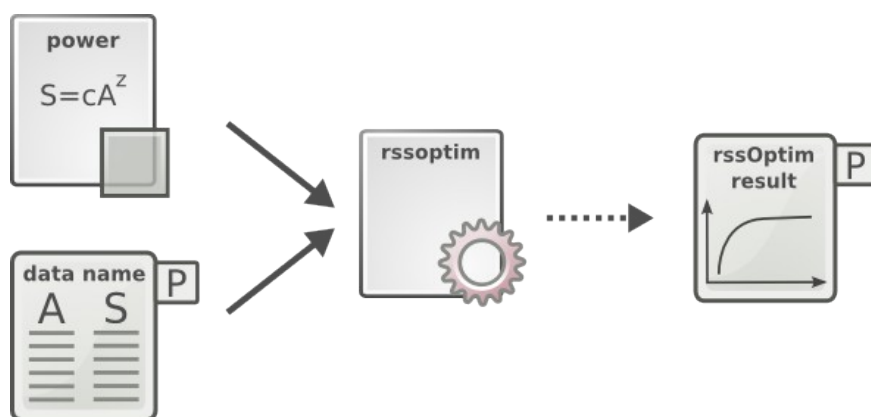
mmSAR usage

Loading mmSAR

After [installation](#), load mmSAR :

```
library(mmSAR)
```

Use case 1 : the simple case.



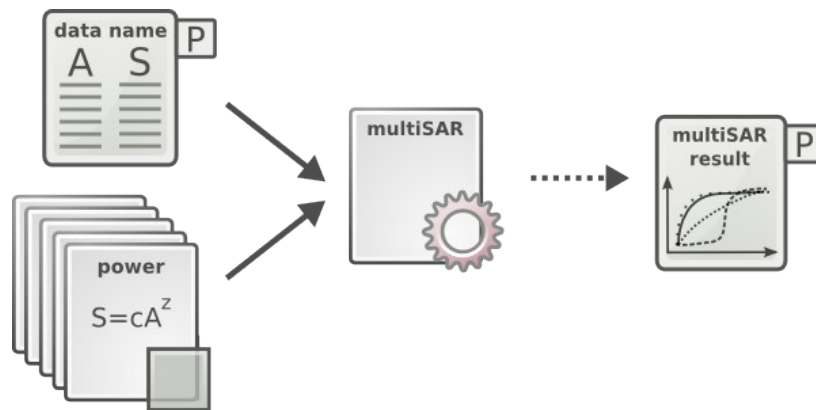
Basic non linear SAR model fits are obtained with the [rssoptim](#) function, this function takes for arguments a [model object](#) and a [data set object](#). Further arguments are available (see ['Functions'](#)). A basic run :

```
#loading the exponential model  
data(expo)
```

```
#loading the Galapagos Islands plants data set (Preston, 1962)  
data(data.glap)
```

```
#fitting the exponential model to the Galapagos dataset  
res <- rssoptim(expo,data.glap)
```

Use case 2 : multimodel SARs.



Multimodel SAR fits are obtained with the [multiSAR](#) function, this function takes for arguments a vector of character strings for model names and a [data set object](#). Further arguments are available (see '[Functions](#)'). A basic run :

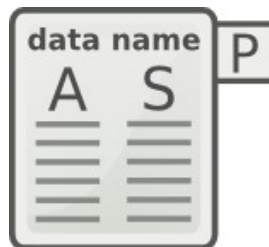
```
#loading all available models
data(power)
data(expo)
data(negexpo)
data(monod)
data(ratio)
data(logistic)
data(lomolino)
data(weibull)

#loading the Galapagos Islands plants data set (Preston, 1962)
data(data.galap)

#creating a vector of model names
mods = c("power","expo","negexpo","monod","logist","ratio","lomolino","weibull")

#fitting all the models to the Galapagos dataset and perform multimodel averaging
resAverage <- multiSAR(modelList=mods,data.galap)
```

Data sets



A description of available example data sets is expected shortly. The example data sets names are :

- data.arr
- data.atl
- data.galap
- data.glea
- data.gleas

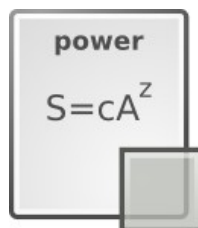
mmSAR handle SAR data sets as list-objects. A data set is a list of 2 elements :

- **\$name** : a character string specifying the name of the data set
- **\$data** : a R data.frame object with 2 columns :
 - **\$a** : a numeric vector of areas
 - **\$s** : a numeric vector of species richness

mmSAR provide some example data sets, for example to load a species-area dataset describing the plants of the Galapagos Islands ([Preston, 1962](#)) use :

```
load(data.galap)
```

Models



Currently, mmSAR handles 8 non linear SAR models such as the power SAR model and others :

Name	Formula	Parameters	Shape	Asymptotic
power	$S = cA^z$	2	convex	no
exponential	$S = c + z\log(A)$	2	convex	no
negative exponential	$S = c(1 - \exp(-zA))$	2	convex	yes
Monod	$S = (cA) / (z + A)$	2	convex	yes
rational function	$S = (c + zA) / (1 + fA)$	3	convex	yes
logistic	$S = c / (1 + \exp(-zA+f))$	3	sigmoid	yes
Lomolino	$S = c / 1 + (z^{\log(f/A)})$	3	sigmoid	yes
cumulative Weibull	$S = c(1 - \exp(-zA^f))$	3	sigmoid	yes

mmSAR handle SAR models as list-objects. A model is a list of 8 elements (examples are given for the exponential model) :

- **\$name** : a character string specifying the name of the model (ex: "expo")
- **\$formula** : an R expression with named parameters (ex: `expression(s == z * log(a) + c)`)
- **\$paramnumber** : a numeric specifying the numbers of parameters in the function (ex: 2)
- **\$paramnames** : a vector of character string of length \$paramnumber specifying the parameters names, as in \$formula (ex: `c("c","z")`)
- **\$parLim** : a vector of character string of length \$paramnumber specifying the

parameters limits from 'R' for $(-\infty; +\infty)$, 'Rplus' for $[0; +\infty)$ or 'unif' for $[0; 1]$ (ex: `c("R","Rplus")`)

- **\$fun** : an R function corresponding to the model function (ex : `function(par,data) {if(length(data)>1) d=data[[1]] else d=data; s = par[2] * log(d) + par[1]; names(s)=c("s.expo"); as.vector(s)}`)
- **\$rssfun** : an R function corresponding to the model Residual Sum of Squares function (ex : `function(par,data,opt){if(opt)par=backLink(par,expo$parLim) ; sum((data[[2]] - (par[2] * log(data[[1]])) + par[1]) ^2) }`)
- **\$init** : an R function corresponding to an initial values calculation for the fitting algorithm, this is a model specific function as \$fun and \$rssfun (ex: `function(data) {semilog.data = data.frame(log(data[[1]]),data[[2]]); names(semilog.data)=c("a","s") ; par=lm(s~a,semilog.data)$coefficients ; names(par)=c("c","z") ; par }`)

To load a model, for example the exponential model use :

`load(expo)`

A more updated documentation will always be available online at <http://mmsar.r-forge.r-project.org/>