

# rspear: Calculate SPEAR in R

Eduard Szöcs

December 26, 2012

## Contents

<b>1</b>	<b>Theoretical background</b>	<b>1</b>
<b>2</b>	<b>Install rspear</b>	<b>1</b>
<b>3</b>	<b>Functions in rspear</b>	<b>1</b>
3.1	spear() . . . . .	1
3.1.1	Description . . . . .	1
3.1.2	get_traits(), Trait-database . . . . .	2
3.1.3	Matching taxon names with trait-database . . . . .	3
3.1.4	Output . . . . .	3
<b>4</b>	<b>Usage of rspear</b>	<b>4</b>
4.1	Data requirements . . . . .	4
4.2	Calculation of SPEAR . . . . .	5
4.3	Modifying trait-values . . . . .	7
4.4	Modifying SPEAR-values . . . . .	9
<b>5</b>	<b>Miscellaneous</b>	<b>9</b>
5.1	Citation . . . . .	9
5.2	Problems . . . . .	9

## 1 Theoretical background

For theoretical background please refer to:

- Liess M, von der Ohe PC 2005. Analyzing effects of pesticides on invertebrate communities in streams. Environmental Toxicology and Chemistry. 24, (4):954-965.

For more information please visit:

- <http://www.systemecology.eu/spear/spear-system/>

## 2 Install rspear

rspear can be installed from CRAN:

```
> install.packages('rspear')
```

rspear is developed at github. If you want to install the latest development version from github use:

```
> install.packages("devtools")
> require(devtools)
> install_github("rspear", "EDiLD")
> require(rspear)
```

## 3 Functions in rspear

### 3.1 spear()

#### 3.1.1 Description

The main function in the rspear-package is `spear()`:

```
spear(x, taxa = NULL, abundance = NULL, group = NULL,
      region = "Eurasia", traits = NULL,
      sensitivity = -0.36, generationTime = 0.5, exposed = 1,
      migration = 0, ...)
```

It takes the following arguments:

**x** data.frame; data.frame with abundances in the long format.

**taxa** character; name of column in x, which holds the taxon-names.

**abundance** character; name of column in x, which holds the abundances.

**group** character-vector; names of columns for groupings.

**region** character; default is set to 'Eurasia', which covers trait-data for Finland, United Kingdom, West Siberia and Central Europe. 'Finland', 'United Kingdom', 'West Siberia' are also allowed and traits may vary between different regions.

**traits** NULL or data.frame; If 'NULL' (default) then it is checked if there is a file 'traits.csv' in the working directory and if this file is up-to-date with the database. If there is no such file, it is downloaded from the web-server. Or a trait-table as returned by `spear()`.

**sensitivity, generationTime, exposed, migration** Threshold values for classification into SPEAR. These values should only be changed if there is strong indication that they are different than these defaults! A species is classified to SPEAR if all criteria are met:

- sensitivity > -0.36
- generationTime >= 0.5
- exposed == 1
- migration == 0

... additional arguments passed to `get_traits()`. Currently only 'check' is available. By default the file is checked if up-to-date. See `get_traits()`.

Generally the defaults are appropriate and one must only specify **x**, **taxa**, **abundance** and **group**.

### 3.1.2 `get_traits()`, Trait-database

In order to minimize traffic on server trait-data is saved locally in the **working directory** (ensure that it has be set correctly).

This is done in `spear()`, via the internal function `get_traits()`:

```
get_traits ( check = TRUE)
```

`spear()` checks if there is file 'traits.csv' in the working directory and otherwise downloads the trait-data from the web-server <http://www.systemecology.eu/spear/spear-calculator/> to a file 'traits.csv' into the working directory.

If this file already exists, it is checked if this file is up-to-date with the web-server.

`get_traits ()` takes one argument 'check' which disables checking when 'FALSE'. Check must be disabled when working offline. Check can be passed directly in `spear()`, as in most of the examples here.

### 3.1.3 Matching taxon names with trait-database

`spear()` matches the taxon-names with the trait table using direct and approximate string matching. It is very likely that the matches are not always 100% (direct) (for example spelling errors, see example in section 'Usage of rspear'). `spear()` then tries to find the species with nearest match (approximate match). However this approximate match must not always be appropriate! When there are non-direct matches a warning is printed and the user is called to check the match:

Warning message:

Non-direct taxon matches!

Check trait table if match is appropriate!!

If `spear` cannot find an appropriate match in the trait table, the SPEAR-value for this species is set to '0' and a warning printed:

```
Warning message:
There were unmatched species:
xxxxxxxxxx
Set SPEAR to 0
```

### 3.1.4 Output

`spear()` returns a list of two elements ('spear' and 'traits):

**spear** a data.frame with the SPEAR-values for every combination of the grouping variables

**traits** a data.frame with the following columns:

**region, exposed, generationTime, sensitivity, migration** species traits used to classify species into SPEAR.

**SPEAR** Classification of species into SPEAR.

**taxa\_data** taxon names as in x.

**taxa\_matched** matched taxon-names in traits-database.

**match\_val** goodness of match. '-1' indicates a direct match, 'NA' indicates a failed match. Values between 0 and 0.5 indicate an approximate match (smaller values - better match)

## 4 Usage of rspear

The usage of the rspear-package is explained on fictitious example data. The example data is shipped with the package:

```
> data(spear_example)
> head(spear_example)
```

	Taxon	Abundance	Year	Site
1	Baetis	1	2007	Sample Point A
2	Baetis rhodani	1	2007	Sample Point A
3	Baetis rodani	1	2007	Sample Point A
4	xxxxxxxxxx	1	2007	Sample Point A
5	Baetis sp.	1	2007	Sample Point A
6	Athericidae	2	2007	Sample Point A

Description of the data-set:

**Taxon** The taxon names. There are spelling errors in the taxon names ('*Baetis rodani*'), Baetis is listed as '*Baetis sp.*' and '*Baetis*' and there is weird species named '*xxxxxxxxxx*'.

**Abundance** Abundances

**Year** There is data from different year, so this is a grouping variable (we want SPEAR-values per year)

**Site** Data has been sampled at four sites, so this is also a group-variable

## 4.1 Data requirements

Like the web application rspear requires data in the long format (see `spear_example` from above). Ecologists often organized their data in wide format (eg. species x samples matrix). This data must be transformed into the long-format, eg using the `melt()` function from the `reshape2` package.

For example if we have a columns for every species (wide-format):

```
> df_wide
```

	Site	Year	Athericidae	Baetis	Baetis fuscatus	Baetis rhodani
1	Sample Point A	2007	2	1	15	1
2	Sample Point A	2008	15	0	0	0
3	Sample Point B	2007	0	0	0	0
4	Sample Point B	2008	0	0	0	0
5	Sample Point C	2007	4	0	3	0
6	Sample Point C	2008	0	0	0	0
7	Sample Point D	2007	0	0	0	0
8	Sample Point D	2008	5	0	0	0

We can transform it to the long format using `melt` from the `reshape2`-package:

```
> require(reshape2)
> df_long <- melt(df_wide, id = c("Site", "Year"))
> head(df_long)
```

	Site	Year	variable	value
1	Sample Point A	2007	Athericidae	2
2	Sample Point A	2008	Athericidae	15
3	Sample Point B	2007	Athericidae	0
4	Sample Point B	2008	Athericidae	0
5	Sample Point C	2007	Athericidae	4
6	Sample Point C	2008	Athericidae	0

## 4.2 Calculation of SPEAR

When we have the data in the long-format we can use `spear()` to calculate the `spear` values. First argument is our data in the long format (`spear_example`), then we must specify the columns coding for taxon-names, abundances and grouping variables:

```
> sp <- spear(spear_example,
+             taxa = "Taxon", abundance = "Abundance", group = c("Year", "Site"))
```

Column names can be entered as characters or (less error-prone) using names():

```
> names(spear_example)

[1] "Taxon"      "Abundance" "Year"      "Site"

> sp <- spear(spear_example ,
+           taxa = names(spear_example)[1], abundance = names(spear_example)[2],
+           group = names(spear_example)[3:4],
+           check = FALSE)
```

Here we can take advantage of the defaults:

**region = "Eurasia"** subsummarises Finland, UK , West Siberia and Central Europe

**traits = NULL** Will check if we have a local file of the trait-data and otherwise download it.

However we are warned, that there have been approximate matches and even no matches with the trait-table:

Warning messages:

```
1: In spear(spear_example , taxa = names(spear_example)[1] ,
:
:
There were unmatched species:
```

```
xxxxxxxxxx
```

```
Set SPEAR to 0.
```

```
2: In spear(spear_example , taxa = names(spear_example)[1] ,
:
:
Non-direct taxon matches!
```

```
Check trait table if match is appropriate!!
```

We can check the matches looking at the trait-table returned by spear():

```
> head(sp$traits)
```

	taxa_data	taxa_matched	match_val	region	exposed	generationTime
18	xxxxxxxxxx	<NA>	NA	<NA>	NA	NA
17	Baetis rodani	Baetis rhodani	0.1	Eurasia	0	0.50000
1	Baetis	Baetis	-1.0	Eurasia	1	0.64564
2	Baetis rhodani	Baetis rhodani	-1.0	Eurasia	0	0.50000
3	Baetis sp.	Baetis sp.	-1.0	Eurasia	1	0.50000
4	Athericidae	Athericidae	-1.0	Eurasia	1	0.68750
	sensitivity migration SPEAR					
18	NA	NA	0			
17	0.02159	0	0			
1	0.02159	0	1			
2	0.02159	0	0			
3	0.02159	0	1			
4	-0.35000	0	1			

Looking at `match_val` we see that there has been no match (`match_val == NA`) for taxon 'xxxxxxx' and 'Baetis rodani' has been matched approximately (`match_val = 0.1`) with 'Baetis rhodani'. All other taxa have been matched directly (`match_val = -1`). The SPEAR-value for 'xxxxxxx' has been set to '0'. We could check and clean our input data, but the matches are appropriate in this case.

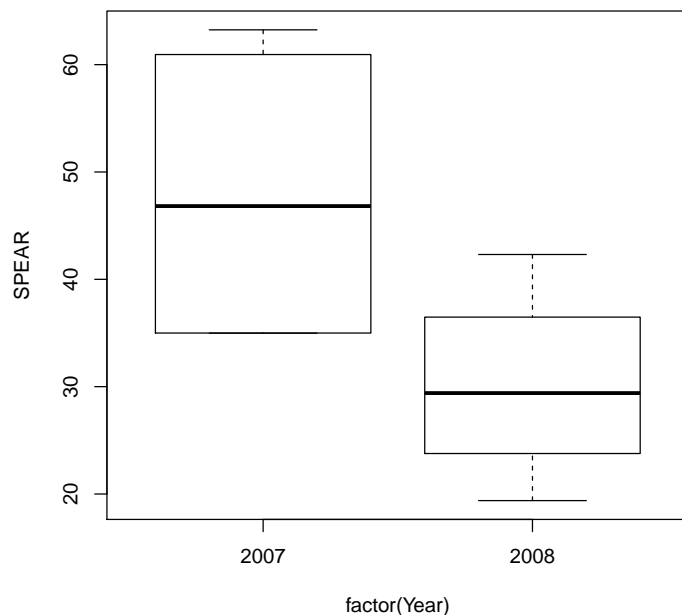
Now let's take a look at the SPEAR-values:

```
> sp$spear
```

	Year	Site	SPEAR
1	2007	Sample Point A	35.00612
2	2007	Sample Point B	63.24266
3	2007	Sample Point C	34.98550
4	2007	Sample Point D	58.64163
5	2008	Sample Point A	42.31371
6	2008	Sample Point B	19.38471
7	2008	Sample Point C	28.15862
8	2008	Sample Point D	30.64599

`spear()` return the result in a `data.frame` which can be directly used for further analysis (plotting, hypothesis testing, etc):

```
> spear_df <- sp$spear
> plot(SPEAR ~ factor(Year), data = spear_df)
```



### 4.3 Modifying trait-values

To modify trait-values we can use the trait-table returned by `spear()`:

First we make a copy of the returned trait-table and then we can modify this table.

For example we set that *Baetis rhodani* is exposed to pesticides:

```
> traits_modi <- sp$traits
> traits_modi[traits_modi$taxa_matched %in% "Baetis rhodani", "exposed"] <- c(1,1)
> head(traits_modi)
```

	taxa_data	taxa_matched	match_val	region	exposed	generationTime
18	xxxxxxxxx	<NA>	NA	<NA>	NA	NA
17	Baetis rodani	Baetis rhodani	0.1	Eurasia	1	0.50000
1	Baetis	Baetis	-1.0	Eurasia	1	0.64564
2	Baetis rhodani	Baetis rhodani	-1.0	Eurasia	1	0.50000
3	Baetis sp.	Baetis sp.	-1.0	Eurasia	1	0.50000
4	Athericidae	Athericidae	-1.0	Eurasia	1	0.68750
	sensitivity	migration	SPEAR			
18	NA	NA	0			
17	0.02159	0	0			
1	0.02159	0	1			



2	0.02159	0	0
3	0.02159	0	1
4	-0.35000	0	1

Note that when selecting *Baetis rhodani* from the trait-table, we have to change two values, because of the spelling error in the data.

This modified trait-table can then be supplied to the 'traits'-Argument of `spear()`:

```
> sp_modi <- spear(spear_example ,
+                 taxa = names(spear_example)[1], abundance = names(spear_example)[2],
+                 group = names(spear_example)[3:4],
+                 traits = traits_modi,
+                 check = FALSE)
> head(sp_modi$spear)
```

	Year	Site	SPEAR
1	2007	Sample Point A	43.16134
2	2007	Sample Point B	63.24266
3	2007	Sample Point C	34.98550
4	2007	Sample Point D	58.64163
5	2008	Sample Point A	42.31371
6	2008	Sample Point B	19.38471

```
> head(sp_modi$traits)
```

	taxa_data	taxa_matched	match_val	region	exposed	generationTime
18	xxxxxxxxx	<NA>	NA	<NA>	NA	NA
17	Baetis rodani	Baetis rhodani	0.1	Eurasia	1	0.50000
1	Baetis	Baetis	-1.0	Eurasia	1	0.64564
2	Baetis rhodani	Baetis rhodani	-1.0	Eurasia	1	0.50000
3	Baetis sp.	Baetis sp.	-1.0	Eurasia	1	0.50000
4	Athericidae	Athericidae	-1.0	Eurasia	1	0.68750

	sensitivity	migration	SPEAR
18	NA	NA	0
17	0.02159	0	1
1	0.02159	0	1
2	0.02159	0	1
3	0.02159	0	1
4	-0.35000	0	1

Note that the SPEAR-values have changed, since *Baetis rhodani* is now classified as Species At Risk.

#### 4.4 Modifying SPEAR-values

*This is currently not supported.*

However it can be done indirectly by modifying the trait-values (see previous section, most easily migration)

## 5 Miscellaneous

### 5.1 Citation

```
> citation("rspear")
```

To cite rspear in publications, please use:

Liess M, Von der Ohe P, 2005. Analyzing effects of pesticides on invertebrate communities in streams. Environmental Toxicology and Chemistry, 24, 954-965.

A BibTeX entry for LaTeX users is

```
@Article{,
  author = {{Liess} and {Matthias} and Von der Ohe and {Peter}},
  title = { Analyzing effects of pesticides on invertebrate communities in streams},
  journal = {Environmental Toxicology and Chemistry},
  year = {2005},
  number = {24},
  pages = {954--965},
}
```

### 5.2 Problems

rspear is developed in github.

Any problems/bugs/feature-request may be reported at github or to the author (Eduard Szöcs, szoe8822@uni-landau.de) directly.