

# Package ‘QsRutils’

December 24, 2020

**Type** Package

**Title** R Functions Useful for Community Ecology

**Version** 0.1.4

**Date** 2020-1-15

**Author** John Quensen

**Maintainer** John Quensen <quensenj@msu.edu>

**Description** Some functions I have written that simplify community analyses by vegan, phyloseq, etc.

**Depends** phyloseq,  
vegan (>= 2.4-6),  
R (>= 3.2.0)

**Imports** ape,  
data.table,  
ggplot2,  
multcompView,  
magrittr,  
reshape2,  
stats

**License** GPL-2

**Encoding** UTF-8

**LazyData** true

**RoxygenNote** 7.0.2

**Suggests** knitr,  
rmarkdown

**VignetteBuilder** knitr

**URL** <https://github.com/jfq3/QsRutils>

## R topics documented:

arc_sine . . . . .	2
asterix . . . . .	3
check_var . . . . .	3
clear_warnings . . . . .	4
comb . . . . .	4
comp_assemble . . . . .	5

comp_comparisons . . . . .	5
comp_make_f_tests . . . . .	6
comp_means_sd . . . . .	6
comp_prepare_otu_table . . . . .	7
comp_prepare_phyloseq . . . . .	8
deg2rad . . . . .	8
generate_password . . . . .	9
get_groups . . . . .	9
get_plot_limits . . . . .	10
goods . . . . .	11
its.root . . . . .	11
log_arc_sine . . . . .	12
make_comparisons . . . . .	12
make_letter_assignments . . . . .	13
merge_2_frames . . . . .	14
ord_labels . . . . .	14
pca_labels . . . . .	15
perm . . . . .	15
plot_df . . . . .	16
prop_filter . . . . .	16
QsRutils . . . . .	17
rad2deg . . . . .	17
rda_labels . . . . .	18
root_phyloseq_tree . . . . .	18
sqrt_arc_sine . . . . .	19
subset_dist . . . . .	19
veganotu . . . . .	20
vegansam . . . . .	20
vegan_stand . . . . .	21

**Index** **22**

---

arc_sine	<i>arc_sine</i>
----------	-----------------

---

**Description**

Arcsine of a Percentage

**Usage**

arc\_sine(x)

**Arguments**

x	A percentage.
---	---------------

**Value**

The arcsine transformation of x.

**Examples**

```
arc_sine(30.1)
```

---

asterix	<i>Indicate Significance with Stars</i>
---------	---

---

**Description**

Indicate Significance with Stars

**Usage**

```
asterix(prob)
```

**Arguments**

prob	p value
------	---------

**Details**

Returns '\*\*\*'; for  $p < 0.001$ , '\*\*' for  $p < 0.01$ , '\*' for  $p < 0.05$ .

**Value**

Character vector of asterisks indicating significance level.

**Examples**

```
asterix(0.039)
```

---

check_var	<i>Check Variance</i>
-----------	-----------------------

---

**Description**

Tests for Heterogeneity of Variances in make\_comparisons Result

**Usage**

```
check_var(otu.pc.transformed, group.vector)
```

**Arguments**

otu.pc.transformed	An OTU matrix of transformed data.
group.vector	A vector of treatments.

**Value**

Prints test results to the console.

**See Also**

make\_comparisons

---

clear_warnings	<i>Clear Warnings</i>
----------------	-----------------------

---

**Description**

Clears all warning messages from the base environment.

**Usage**

```
clear_warnings()
```

**Details**

Sometimes when working in the console R retains a list of warnings such that they keep being reported after the function call which originated them. This function removes them so that they are not a nuisance

**Examples**

```
clear_warnings()
```

---

comb	<i>comb</i>
------	-------------

---

**Description**

Calculates the number of combinations of n things drawn r at a time.

**Usage**

```
comb(n, r, repetition = FALSE)
```

**Arguments**

n	The total number of items.
r	The number of items to be drawn.
repetition	A logical, whether or not repetitions are allowed. FALSE by default.

**Value**

An integer giving the number of ways a set of r items can be drawn from a set of n items.

**Examples**

```
comb(5, 3)
comb(5, 3, repetition = TRUE)
```

---

comp_assemble	<i>Assemble Comparison Parts</i>
---------------	----------------------------------

---

**Description**

Assembles Comparison Data Frame

**Usage**

```
comp_assemble(part1, part2, part3)
```

**Arguments**

part1	Result from comp_means_sd
part2	Result from comp_make_f_tests
part3	Result from comp_comparisons

**Value**

A summary data frame of differential abundances by taxon and treatment.

---

comp_comparisons	<i>Make Comparisons</i>
------------------	-------------------------

---

**Description**

Calculates the treatment comparison portion of a table comparing relative abundances of each taxon among treatments.

**Usage**

```
comp_comparisons(  
  otu.pc,  
  otu.pc.trans,  
  grps,  
  p.adjust.method = "BH",  
  pool.sd = FALSE  
)
```

**Arguments**

otu.pc	An OTU table of percentages.
otu.pc.trans	An OTU table of transformed data.
grps	A vector of treatment groups for which to make comparisons.
p.adjust.method	Adjustment method for multiple comparisons.
pool.sd	A logical, whether or not to pool standard deviations.

**Value**

A data frame of differences in relative abundances among treatments.

---

comp\_make\_f\_tests      *Make F Tests*

---

**Description**

Calculates omnibus F tests to be included in a table comparing relative abundances of each taxon among treatments.

**Usage**

```
comp_make_f_tests(otu.pc.trans, grps, var.equal = FALSE)
```

**Arguments**

otu.pc.trans	An OTU table of transformed data from comp_prepare_otu_table.
grps	A vector of treatment groups for which to make comparisons.
var.equal	Logical, whether or not to assume variances equal.

**Value**

A data frame of the F-test results.

---

comp\_means\_sd      *Calculate Means and Standard Deviations*

---

**Description**

Calculates means and standard deviation for each taxon to be included in a table comparing relative abundances of each taxon among treatments.

**Usage**

```
comp_means_sd(otu.pc)
```

**Arguments**

otu.pc	An OTU table with data as percentages.
--------	--

**Details**

The OTU table should be created with `comp_prepare_otu_table`.

**Value**

A data frame with means and standard deviations by taxon.

---

`comp_prepare_otu_table`*Prepare OTU Table*

---

**Description**

Make OTU tables for making comparisons of relative abundances among treatments.

**Usage**

```
comp_prepare_otu_table(  
  expt.taxon.pc,  
  grps = "Treatment",  
  transformation = "sqrt_arc_sine"  
)
```

**Arguments**

`expt.taxon.pc` Phyloseq object from `comp_prepare_phyloseq` with percentages in the `otu_table`.

`grps` Factor in sample data for which to make comparisons.

`transformation` Transformation function to use.

**Details**

`transformation` may be "none" or a user-supplied function name in quotation marks or any of the built-in transformations("arc\_sine", "log\_arc\_sine", or "sqrt\_arc\_sine"). The `+sqrt_arc_sine` has generally proven most effective.

**Value**

A list consisting of an OTU table with percentages, an OTU table with transformed data, and a vector of treatment groups.

---

comp\_prepare\_phyloseq *Prepare Phyloseq*

---

### Description

Prepares a phyloseq object for making comparisons of relative abundances among treatments.

### Usage

```
comp_prepare_phyloseq(expt, taxrank = "Phylum", pc.filter = 0.01)
```

### Arguments

expt	Experiment level phyloseq object.
taxrank	Taxonomic rank for which to make comparisons.
pc.filter	Minimum percentage of total counts to include rank in result.

### Details

The `otu_table` in one of the returned objects has been transformed to percentages based on the original phyloseq object supplied. The taxa in both have been filtered to include only OTUs initially present at  $\geq$  `pc.filter` times the original total counts. For both only `taxrank` is included in the `tax_table`.

### Value

A list of two modified experimnt level phyloseq objects

---

deg2rad *deg2rad*

---

### Description

Degrees to radians

### Usage

```
deg2rad(x)
```

### Arguments

x	Angle in degrees
---	------------------

### Value

Angle in radians.

### Examples

```
deg2rad(90)
```



---

generate_password	<i>Generate a Password</i>
-------------------	----------------------------

---

**Description**

Generates a random character string of specified length.

**Usage**

```
generate_password(n, type = "alpha_numeric")
```

**Arguments**

n	Number of characters in password.
type	c("alpha_numeric", "anything_else")

**Details**

If type equals "alpha\_numeric" (the default), only alpha-numeric characters are used to generate the password. If type does not equal "alpha\_numeric" then at least one non-alpha-numeric symbol will be included in the password. In either case, the alpha characters used are both upper and lower case.

**Value**

A character string.

**Examples**

```
generate_password(8)
```

---

get_groups	<i>get_groups</i>
------------	-------------------

---

**Description**

Assign treatment groups based on pairwise t-tests.

**Usage**

```
get_groups(ptt.rslt, alpha = 0.05, rm.subset = FALSE)
```

**Arguments**

ptt.rslt	Result from the stats function pairwise.t.test.
alpha	Confidence level.
rm.subset	A logical; remove group subsets if true.

## Details

This function aids in making letter assignments as to which treatments are significantly different. Also returns a square matrix of alpha values for all pairwise differences. This square matrix can serve as input to the multcompLetters function of the multcompView package which provides letter assignments. If rm.subset is FALSE, then groups such as A,B and A, B, C may be reported. This is redundant in the sense the A, B is a subset of A, B, C. In this case if rm.subset is FALSE, the group A, B is not reported.

## Value

A list consisting of groups of treatment groups that are not significantly different and a matrix of p values.

## See Also

make\_letter\_assignments

---

get\_plot\_limits      *Get ggplot Plot Limits*

---

## Description

Gets the ranges for the width and height of a ggplot panel.

## Usage

```
get_plot_limits(plot)
```

## Arguments

plot                  A plot created with ggplot2

## Value

A list: xmin, xmax, ymin, ymax

## Examples

```
library(ggplot2)
data(iris)
plt <- ggplot(data=iris, aes(x=Species, y=Petal.Length)) + geom_boxplot()
get_plot_limits(plt)
```

---

goods *Calculate Good's Coverage*

---

### Description

Calculates Good's coverage from a community data matrix with samples as rows and OTUs as columns.

### Usage

goods(com)

### Arguments

com a vegan compatible community data matrix.

### Value

A table with the headings number of singletons, number of sequences, and Good's coverage for each sample in rows.

### References

Good, I. J. 1953. The Population Frequencies of Species and the Estimation of Population Parameters. *Biometrika* 40:237-264.

---

its.root *An Experiment Level phyloseq Object*

---

### Description

Based on ITS2 sequences amplified from corn roots.

### Usage

its.root

### Format

A phyloseq object with otu\_table, sample\_data and tax\_table. The sample\_data variables are:

**P** Phosphorous level, H or L

**Genotype** One of three: 2, 3, and C

**Label** A code for treatments: 2HR, 2LR, 3HR, 3LR, CHR, CLR

---

log_arc_sine	<i>log_arc_sine</i>
--------------	---------------------

---

**Description**

Log of the arc-sine Transformation of a Percentage

**Usage**

```
log_arc_sine(x)
```

**Arguments**

x                    A percentage.

**Value**

The common logarithm of the arcsine transformation of x.

**Examples**

```
log_arc_sine(x = 30.1)
```

---

make_comparisons	<i>Make Multiple Comparisons on Transformed Data</i>
------------------	--

---

**Description**

Makes multiple comparisons of the relative abundances of taxa between treatment groups using the pairwise.t.test. Data may be transformed by a user supplied function. Three are included in this package.

**Usage**

```
make_comparisons(  
  expt,  
  taxrank = "Phylum",  
  grps = "Treatment",  
  transformation = "none",  
  pc.filter = 0.01,  
  p.adjust.method = "BH",  
  pool.sd = FALSE  
)
```

**Arguments**

expt	Experiment level phyloseq object.
taxrank	Rank for which to make comparisons.
grps	Factor in sample data for which to make comparisons.
transformation	Transformation function to use.
pc.filter	Minimum percentage of total counts to include rank in result.
p.adjust.method	Adjustment method for multiple comparisons.
pool.sd	Logical, whether or not to pool standard deviations.

**Details**

transformation may be "none" or a user-supplied function name in quotation marks or any of the built-in transformations ("arc\_sine", "log\_arc\_sine", or "sqrt\_arc\_sine"). The "sqrt\_arc\_sine" has generally proven most effective.

**Value**

A data frame with taxa as rows and results in columns.

**See Also**

arc\_sine, log\_arc\_sine, sqrt\_arc\_sine, check\_var

---

make\_letter\_assignments

*Make Letter Assignments*

---

**Description**

Makes letter assignments for treatment groups that are not significantly different.

**Usage**

```
make_letter_assignments(ptt.rslt, significance = 0.05)
```

**Arguments**

ptt.rslt	Output from the pairwise.t.test function.
significance	Alpha level to be declared a significant difference.

**Details**

Letter assignments are made using Piepho's algorithm.

**Value**

Lists of letter assignments.

## References

Piepho, H. P. 2004. An algorithm for a letter-based representation of all-pairwise comparisons. *Journal of Computational and Graphical Statistics* **13**:456-466.

---

merge_2_frames	<i>Merge Two Data Frames</i>
----------------	------------------------------

---

## Description

Merge two data frames by their row names.

## Usage

```
merge_2_frames(one, two)
```

## Arguments

one	A data frame.
two	A second data frame.

## Details

Merges data frames by common row names. This function differs from `merge.data.frames` in that the merged data frame returned has row names and not a new column of the row names.

## Value

A merged data frame.

---

ord_labels	<i>Make Ordination Axis Labels</i>
------------	------------------------------------

---

## Description

Makes ordination axis labels that include, if appropriate, the % total variance explained by each axis.

## Usage

```
ord_labels(ord)
```

## Arguments

ord	A vegan ordination object.
-----	----------------------------

## Details

If there are no eigenvalues in `ord`, or if any eigenvalues are less than 0, each element of the vector returned has the form "DIMn" where N is the axis number. Otherwise, each element of the vector returned has the form "AxisN xx.x%" where "Axis" is taken from the vector of eigenvalues in `ord` if they are named or simply "DIM" if they are not, N is the number of the axis, and xx.x is the % of total variance explained by the axis.

**Value**

A character vector, each element of which can be used to label the corresponding axis of an ordination plot.

---

pca_labels	<i>Make PCA Axis Labels</i>
------------	-----------------------------

---

**Description**

Makes PCA axis labels that include the

**Usage**

```
pca_labels(pca)
```

**Arguments**

pca                    Object containig the results of vegan's rda function.

**Details**

Each element of the vector returned has the form "PCAn xx.x"

**Value**

A character vector, each element of which can be used to label the corresponding axis of a PCA plot.

---

perm	<i>Permutations</i>
------	---------------------

---

**Description**

Retuns the number of permutaions of n things taken r at a time.

**Usage**

```
perm(n, r, repetition = FALSE)
```

**Arguments**

n                    Total number of items.  
r                    Number of items drawn.  
repetition        A logical, whether or not repetitions are allowed. FALSE by default.

**Value**

An integer giving how many ways m things can be drawn n at a time.

**Examples**

```
perm(10, 5)
perm(10, 5, repetition = TRUE)
```

---

plot_df	<i>A Data File in Long Format</i>
---------	-----------------------------------

---

**Description**

Used in Case 3 of the vignette `make_comparisons`

**Usage**

```
plot_df
```

**Format**

A data file in long format used for a ggplot. The `sample_data` variables are:

**Treatment** A code for genotype (2, 3, or C), P level (H or L) and sample type (R)

**Family** One of the families in Gigasporaceae

**Percent** Percent of total counts for family and treatment combination.

---

prop_filter	<i>Filter OTUs by Abundance</i>
-------------	---------------------------------

---

**Description**

Allows subsetting of a phyloseq object according to the relative abundance of OTUs in a minimal number of samples. Returns a logical vector of OTUs that are at least `n%` of the sequences in at least `m` samples.

**Usage**

```
prop_filter(x, n, m)
```

**Arguments**

<code>x</code>	A phyloseq object.
<code>n</code>	Minimum percentage to keep OTU.
<code>m</code>	Minimum number of samples.

**Details**

The functions creates a logical vector to be used in subsetting a phyloseq object according to the relative abundance of OTUs in a given number of samples. For example, if `n = 1` and `m = 2`, then the OTUs to be kept must represent at least 1% of the sequences in at least 2 samples. The vector is then used as an argument to the phyloseq object `'prune_taxa'`.



**Value**

A logical vector of OTUs to keep.

---

QsRutils

*QsRutils: R Functions Useful for Community Ecology*

---

**Description**

The QsRutils package contains functions I have written to make some aspects of using phyloseq and vegan simpler. I originally called the package MyRutils, but that does not make much sense if I am posting it publically!

---

rad2deg

*rad2deg*

---

**Description**

Radians to degrees

**Usage**

rad2deg(x)

**Arguments**

x                      Angle in radians

**Value**

Angle in degrees.

**Examples**

```
rad2deg(pi * 0.5)
```

---

rda_labels	<i>Make RDA Axis Labels</i>
------------	-----------------------------

---

**Description**

Makes RDA axis labels that include the

**Usage**

```
rda_labels(rda)
```

**Arguments**

rda                    Object that contains CCA result from vegan's rda function.

**Details**

Each element of the vector returned has the form "RDAn xx.x"

**Value**

A character vector, each element of which can be used to label the corresponding axis of an RDA plot.

---

root_phyloseq_tree	<i>Root Tree in phyloseq Object</i>
--------------------	-------------------------------------

---

**Description**

Roots an unrooted tree in a phyloseq object

**Usage**

```
root_phyloseq_tree(phylo)
```

**Arguments**

phylo                    A phyloseq object containing an unrooted tree

**Details**

The tree is rooted by the longest terminal branch.

**Value**

The same phyloseq object with a rooted tree

**Examples**

```
## Not run:  
expt.rooted <- root_phyloseq_tree(expt.unrooted)  
  
## End(Not run)
```

---

sqrt_arc_sine	<i>sqrt_arc_sine</i>
---------------	----------------------

---

**Description**

Square Root of the arc-sine of a Percentage

**Usage**

```
sqrt_arc_sine(x)
```

**Arguments**

x	A percentage.
---	---------------

**Value**

The square root of the arcsine transformation of x.

**Examples**

```
sqrt_arc_sine(30.1)
```

---

subset_dist	<i>Subset Distance Matrix</i>
-------------	-------------------------------

---

**Description**

Subsets a distance matrix.

**Usage**

```
subset_dist(physeq, d.matrix)
```

**Arguments**

physeq	An experiment level phyloseq object.
d.matrix	A distance matrix.

**Details**

Some distance matrices take a long time to calculate for large data sets. This is especially true of unifrac and generalized unifrac distances calculated by GUniFrac. If distances are first calculated from data in a large experiment level phyloseq object and then it is desired to perform PERMANOVA (with adonis) on a subset of that object, this function provides a means of sub-setting the distance matrix so that it does not have to be calculated again for the subset data. The arguments are the distance matrix for the original phyloseq object and the smaller phyloseq object subset from the original.

**Value**

A distance matrix of smaller dimensions.

**References**

Chen J, Bittinger K, Charlson ES et al. (2012) Associating microbiome composition with environmental covariates using generalized UniFrac distances. *Bioinformatics*, 28, 2106-2113.

---

veganotu	<i>Extract Vegan OTU Table</i>
----------	--------------------------------

---

**Description**

Extracts a vegan compatible OTU table from a phyloseq object.

**Usage**

```
veganotu(physeq)
```

**Arguments**

physeq            A phyloseq object containing at least an OTU table.

**Value**

A matrix with samples in rows and OTUs in columns.

---

vegansam	<i>Extract Sample Data Table</i>
----------	----------------------------------

---

**Description**

Extracts a sample data table from a phyloseq object.

**Usage**

```
vegansam(physeq)
```

**Arguments**

physeq            A phyloseq object containing sample\_data.

**Value**

A data frame with samples in rows and factors and/or variables in columns.

---

vegan_stand	<i>Standardize a Phyloseq OTU Table</i>
-------------	---

---

**Description**

Applies any vegan decostand standardization method to a phyloseq OTU table.

**Usage**

```
vegan_stand(physeq, method = "hellinger", ...)
```

**Arguments**

physeq	A phyloseq object containing at least an OTU table.
method	A method from vegan's decostand function.
...	Other parameters passed to vegan's decostand function.

**Value**

Returns a phyloseq object with transformed OTU table.

# Index

- \* **datasets**
  - its.root, 11
  - plot\_df, 16
- arc\_sine, 2
- asterix, 3
  
- check\_var, 3
- clear\_warnings, 4
- comb, 4
- comp\_assemble, 5
- comp\_comparisons, 5
- comp\_make\_f\_tests, 6
- comp\_means\_sd, 6
- comp\_prepare\_otu\_table, 7
- comp\_prepare\_phyloseq, 8
  
- deg2rad, 8
  
- generate\_password, 9
- get\_groups, 9
- get\_plot\_limits, 10
- goods, 11
  
- its.root, 11
  
- log\_arc\_sine, 12
  
- make\_comparisons, 12
- make\_letter\_assignments, 13
- merge\_2\_frames, 14
  
- ord\_labels, 14
  
- pca\_labels, 15
- perm, 15
- plot\_df, 16
- prop\_filter, 16
  
- QsRutils, 17
  
- rad2deg, 17
- rda\_labels, 18
- root\_phyloseq\_tree, 18
  
- sqrt\_arc\_sine, 19
  
- subset\_dist, 19
  
- vegan\_stand, 21
- veganotu, 20
- vegansam, 20