Journal of Statistical Software

February 2016, Volume 69, Code Snippet 2. *doi: 10.18637/jss.v069.c02*

cooccur: Probabilistic Species Co-Occurrence Analysis in R

Daniel M. Griffith
Wake Forest University

Joseph A. Veech
Texas State University

Charles J. Marsh
University of Leeds

Abstract

The observation that species may be positively or negatively associated with each other is at least as old as the debate surrounding the nature of community structure which began in the early 1900's with Gleason and Clements. Since then investigating species co-occurrence patterns has taken a central role in understanding the causes and consequences of evolution, history, coexistence mechanisms, competition, and environment for community structure and assembly. This is because co-occurrence among species is a measurable metric in community datasets that, in the context of phylogeny, geography, traits, and environment, can sometimes indicate the degree of competition, displacement, and phylogenetic repulsion as weighed against biotic and environmental effects promoting correlated species distributions. Historically, a multitude of different co-occurrence metrics have been developed and most have depended on data randomization procedures to produce null distributions for significance testing. Here we improve upon and present an R implementation of a recently published model that is metric-free, distribution-free, and randomization-free. The R package, **cooccur**, is highly accessible, easily integrates into common analyses, and handles large datasets with high performance. In the article we develop the package's functionality and demonstrate aspects of co-occurrence analysis using three sample datasets.

Keywords: co-occurrence, species niche, community ecology.

1. Introduction

The analysis of species co-occurrence patterns is a fundamental task for many ecological investigations. Species coexistence, community structure and assembly, and the maintenance of biodiversity are all essentially founded on the ways in which species co-occur with one another. Even the very early Clementsian and Gleasonian perspectives on the organization of plant communities can be put in the context of species co-occurrence (Hoagland and Collins

1997). At some fundamental level, two species are either positively, negatively, or randomly associated with one another. Indeed, recently developed pairwise approaches are intended to classify species pairs as representing positive, negative, and sometimes random associations (Sfenthourakis, Tzanatos, and Giokas 2005; Gotelli and Ulrich 2010; Veech 2013). Furthermore, co-occurrence is a measurable property of a pair of species. The probabilistic model of species co-occurrence (Veech 2013) measures co-occurrence in the most straightforward way as the number of sampling sites where two species co-occur. Observed co-occurrence can be compared to the expected co-occurrence where the latter is the product of the two species' probability of occurrence multiplied by the number of sampling sites: $E(N_{1,2}) = P(1) \times P(2) \times N$. The probabilistic model employs combinatorics to determine the probability that the observed frequency of co-occurrence is significantly large and greater than expected (positive association), significantly small and less than expected (negative association), or not significantly different and approximately equal to expected (random association) (Veech 2013). The probabilistic model is very different from nearly all previous methods for analyzing co-occurrence in that data randomization is not required (Veech 2013). However, because the probabilistic model uses combinatorics, the algorithms can often generate enormous numbers (e.g., 1×10^{50}) when there is a large number of sampling sites (>200) in the dataset. Simply storing such large numbers with precision can be difficult for many computing languages and spreadsheet programs. Therefore, we were motivated to develop a version of the probabilistic model in R, a flexible programming language popular among ecologists, so as to increase the availability of the model as an easy-to-use method for conducting pairwise co-occurrence analyses.

The original combinatorics approach of Veech (2013) can alternatively be cast as a random sampling with replacement scenario and thus represented by the probability mass function of the hypergeometric distribution. This scenario is often illustrated by randomly selecting marbles of two different colors out of an urn. The probability mass function gives the probability of selecting X marbles of a certain color given a particular number of marbles randomly grabbed out of a specified total number of marbles in the urn. For species co-occurrence, the scenario is tweaked slightly such that we calculate the probability of selecting a site (or sample) that has species #1 given that it already has species #2. The probability that the two species co-occur at exactly j number of sites is given by,

$$P_j = \frac{\binom{N_1}{j} \times \binom{N-N_1}{N_2-j}}{\binom{N}{N_2}} \tag{1}$$

For $j = 1$ to N_1 sites (or samples), N_1 = number of sites where species #1 occurs, N_2 = number of sites where species #2 occurs, and N = total number of sites that were surveyed (where both species could occur). The term, $\binom{N_1}{j}$, represents the number of ways of selecting j sites that have species #1 given that there are N_1 such sites in the "population" of all sites. The term $\binom{N-N_1}{N_2-j}$ represents the number of ways of selecting $N_2 - j$ sites that have species #2 but not species #1 given that there are $N - N_1$ such sites. Multiplying these two quantities together (the numerator) gives the total number of ways of selecting j sites that have species #1 and #2. The denominator, $\binom{N}{N_2}$, represents the total number of ways that N_2 number of sites could be obtained out of a total of N sites. Thus the equation is giving the proportion of the N_2 sites that also have species #1 under the condition that the two species co-occur at j sites. We note that this equation (Equation 1) has only three combination terms compared to the five in Equation 1 of Veech (2013) which also requires calculating the product of three combination terms in the numerator and the product of two combination

terms in the denominator. Equation 1 of Veech (2013) involves very large numbers except for the smallest of datasets. The above equation (Equation 1) is mathematically more succinct and this results in much quicker calculation than that of Veech (2013). By default **cooccur** uses this hypergeometric approach.

1.1. The analysis of species co-occurrence patterns in R

The ecological literature has produced a number of methods for detecting pairs of species that share sites more or less frequently than expected. Many of these methods have been implemented in various programming languages including the statistical language R (R Core Team 2015). For example, the R package **vegan**, which houses a plethora of community ecology analyses, identifies patterns of co-occurrence through comparison of community data to simulated null models of species occurrence (Oksanen *et al.* 2016). The package **vegan** also has functionality for calculating community beta-diversity metrics that, when modified, can also produce species dissimilarity metrics. Notably, if one assumes that species have equal probabilities of occurrence across sites, the "Raup-Crick" dissimilarity index can be applied for species rather than sites and a species dissimilarity matrix is produced that is numerically equivalent to the probability (as calculated in **cooccur**) that the observed frequency of species co-occurrence is greater than expected. In this case the advantage of **cooccur** is the additional calculation of the probability that species co-occur less than expected. However, if species are believed to be less likely to occur in species poor sites then null model approaches might be necessary to account for this. Other null model co-occurrence tests are available in the **spaa** package (Zhang 2013). Aspects of Gotelli's "EcoSim" software are now available in "EcoSimR", a downloadable suite of R scripts; however, much of the functionality currently remains in executable form or in related Fortran scripts (Ulrich 2008; Gotelli and Ellison 2013; Sfenthourakis *et al.* 2005). Additionally, distance based tools for determining pairwise species co-occurrence patterns exist in the phylo-community ecology package **picante** (Kembel *et al.* 2010). Finally, some studies have implemented R versions of their co-occurrence based approaches in their publications, such as Fridley, Vandermast, Kuppinger, Manthey, and Peet (2007) who used co-occurrence to investigate specialist versus generalist species in Eastern North American plant communities. In this article we improve, describe, and evaluate runtime performance of an R implementation of the probabilistic co-occurrence model from (Veech 2013). The package is implemented using the improvements made in Equation 1 and by default calculates co-occurrence probabilities using the hypergeometic distribution (R function `phyper()`).

2. Probabilistic co-occurrence analysis in cooccur

In this article we present the R package **cooccur** for species co-occurrence analysis. This section will describe the installation, functionality, and application of the package. We demonstrate using three community datasets, supplied with the package distribution, which include carabid beetles from Poland (Ulrich and Zalewski 2006; Gotelli and Ulrich 2010), Great Basin rodents (Brown and Kurzius 1987), and Galapagos finches (Sanderson 2000). For a mathematical treatment of the probabilistic model of species co-occurrence see Veech (2013).

	Seymour	Baltra	Isabella	Fernandina	Santiago	Rabida	Pinzon	Santa Cruz	Santa Fe	San Cristobal	Espanola	Floreana	Genovesa	Marchena	Pinta	Darwin	Wolf
G. magnirostris	0	0	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1
G. fortis	1	1	1	1	1	1	1	1	1	1	0	1	0	1	1	0	0
G. fuliginosa	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	0	0
G. difficilis	0	0	1	1	1	0	0	1	0	1	0	1	1	0	1	1	1
G. scandens	1	1	1	0	1	1	1	1	1	1	0	1	0	1	1	0	0
G. conirostris	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0
Ca. psittacula	0	0	1	1	1	1	1	1	1	0	0	1	0	1	1	0	0
Ca. pauper	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
Ca. parvulus	0	0	1	1	1	1	1	1	1	1	0	1	0	0	1	0	0
P. crassirostris	0	0	1	1	1	1	1	1	1	1	0	1	0	1	1	0	0
Ca. pallida	0	0	1	1	1	0	1	1	0	1	0	0	0	0	0	0	0
Ca. heliobates	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0
Ce. olivacea	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Table 1: An example of the type of data used in species co-occurrence analysis. Data are presence-absences of *Geospiza*, *Camarrhynchus*, *Platyspiza*, and *Certhidea* finches on Galapagos Islands from Sanderson 2000. Rows are species and columns are islands.

2.1. Installation

The **cooccur** package is distributed on CRAN (Comprehensive R Archive Network) at url https://CRAN.R-project.org/package=cooccur. Therefore, the package can be accessed simply through the R console. The package has dependencies on other R packages **gmp** (Lucas, Scholz, Boehme, Jasson, and Maechler 2014), **ggplot2** (Wickham 2009), and **reshape** (Wickham 2007) – these will be loaded automatically with **cooccur**.

```
R> install.packages("cooccur")
R> library("cooccur")
```

2.2. Overview

The **cooccur** package centers around the function `cooccur()`. This function accepts community data (e.g., species by site matrix or vice-versa) in the form of a data frame or matrix and returns a list containing pairwise species co-occurrence results. In the probabilistic co-occurrence model, community data is used in presence-absence form and will be converted to occupancies if abundances, cover-classes, or counts (etc.) are supplied (anything not 0 is a presence and coded as "1"). If community data have species names they should be stored in either the row names or column names of the data object—the default is row names. An example of a species by site matrix is shown in Table 1.

The `cooccur()` function returns an object of class `cooccur` which is a list containing summary statistics from the analysis and a data frame of all pairwise species combinations and their

Field name	Field definition
`sp1`	Numeric label giving the identity of species 1, assigned based on the order in the input matrix
`sp2`	Numeric label for species 2
`sp1_inc`	Number of sites (or samples) that have species 1
`sp2_inc`	Number of sites that have species 2
`obs_cooccur`	Observed number of sites having both species
`prob_cooccur`	Probability that both species occur at a site
`exp_cooccur`	Expected number of sites having both species
`p_lt`	Probability that the two species would co-occur at a frequency less than the observed number of co-occurrence sites if the two species were distributed randomly (independently) of one another
`p_gt`	Probability of co-occurrence at a frequency greater than the observed frequency
`sp1_name`	If species names were specified in the community data matrix this field will contain the supplied name of `sp1`
`sp2_name`	The supplied name of `sp2`

Table 2: Definitions for column names presented in the probability table. Records in the probability table each represent one species pairing.

probability of co-occurring more frequently or less frequently than expected by their observed frequency. This pairwise probability table is the primary result of the analysis conducted by the `cooccur()` function and a detailed description of each field returned can be found in Table 2. Objects of class `cooccur` (i.e., the result `cooccur()`) have `print()`, `summary()`, and `plot()` methods defined. Calling `print()` on the `cooccur` object will output a pairwise probability table containing significant species combinations only. Note, to access the entire table with all species pairs the `prob.table()` function should be used or the `$results` element of the `cooccur` object should be accessed. `summary()` will return an analysis-wide count of the number of species combinations classified as positive, negative, or random. `plot()` will create a lower triangle heat map visually indicating the significant positive, negative, and random co-occurrence patterns among all species.

We have also created helper functions and additional visualization functions to help users explore, interpret, and conduct further analysis with the results of the probabilistic `cooccur()` model. `pair()` is a function that extracts the significant positive and negative association data for a single species. `pair.attributes()` is a function that will summarize for each species the percent of its associations that are positive, negative, or random. These data can be visually represented using the `pair.profile()` function which will create a ranked (by percent significant associations) bar plot showing the percentage of positive, negative, and random associations for each species. We have also added a function `obs.v.exp()` which plots the observed versus expected number of co-occurrence sites for each species pair. Table 3 contains a description of each function in the package.

Analysis with **cooccur** produces, for all species pairs, exact probabilities of co-occurrence greater than or less than what is observed (for discussion of probability calculations see Section 3. Model performance and development). This analysis is also distribution-free and the results can be interpreted and reported as p-values, without reference to a statistic.

Therefore, given two species in a dataset, a $P(lt) \leq \alpha$ suggests that those two species are negatively associated (where $P(lt) =$ `$p_lt` and $\alpha = 0.05$). The next section will demonstrate how to conduct this analysis in R, summarize the results, extract and interpret the desired results, visualize the results, and prepare output for use in other analyses.

2.3. Example analysis: Finches

For the purpose of leading the user through the analysis workflow of our package we will demonstrate using an example dataset describing finch occurrences in the Galapagos. The aim of the sample analysis is to determine the degree to which communities contain species that are positively, negatively, and randomly associated with one another, investigate the contribution of individual species to these patterns, and to quantify the strength of the positive and negative associations between species pairs. In addition we explore options for visualizing these results, comparing findings among datasets, and facilitating downstream analysis of results from the probabilistic co-occurrence model.

The finches data are presence-absences collected from different islands of the Galapagos; these data are presented as an example of the data format used by our package (see Table 1; Sanderson 2000). The data are available in R as a data frame, which includes species names and site names. Providing species names makes the results easier to interpret and pipeline into downstream analyses, compared using species numbers. Methods for extracting and using site-specific information are not yet implemented, but species names are acceptable in the row names (e.g., Table 1; or column headings, if using a site by species matrix) of the data object. Site names are ignored. The finch dataset can be loaded into R using the `data()` command. Because the finches data are organized with species as rows and sites as columns (i.e., species by site) we can specify `type = "spp_site"` as a parameter to the function `cooccur()` and since we have species names we should specify `spp_names = TRUE`. Lastly, according to their probabilities of co-occurrence some species in the dataset will be expected to share less than one site and it is recommended to filter these pairs from the analysis using `thresh = TRUE`, when the goal is to summarize the most important species associations. This threshold is discussed in more detail in Veech (2013)—however, its purpose is to remove from analysis species that simply do not have sufficient occurrence data.

```
R> data("finches")
R> cooccur.finches <- cooccur(mat = finches, type = "spp_site",
+    thresh = TRUE, spp_names = TRUE)
R> class(cooccur.finches)

[1] "cooccur"
```

The `cooccur()` function produces an output object of class `cooccur` containing all of the results from the co-occurrence analysis. As a first step, the `summary()` method will quickly supply a readout of the total positive, negative, and random species pairs classified by the algorithm. In addition, the function reports on the number of species and sites analyzed, the number of species pairs removed from the analysis by our threshold, and the number of species pairs that were not classifiable due to low statistical power. In calculating the percentage of non-random species associations the unclassified pairs are included in the count of total pairs, whereas those removed by the co-occurrence threshold are not.

```
R> summary(cooccur.finches)

Call:
cooccur(mat = finches, type = "spp_site", thresh = TRUE, spp_names = TRUE)

Of 78 species pair combinations, 14 pairs (17.95%) were removed from the
analysis because expected co-occurrence was < 1 and 64 pairs were analyzed

Cooccurrence Summary:
  Species  Sites  Positive  Negative  Random Unclassifiable Non-random (%)
     13.0   17.0      14.0       1.0    42.0            7.0           23.4
```

This result suggests that most of the classifiable species pairs had 'truly random' associations. The significant non-random associations were mostly positive (14 positive compared to one negative). The seven unclassifiable species pairs are determined by a heuristic criteria that classifies as 'truly random' species pairs that do not differ significantly from their expected number of co-occurrences and deviate by less than 10 % of the total number of sites–the remainder are deemed unclassifiable. Currently, all of our subsequent visualizations treat these unclassifiable pairs as random in order to highlight the positive and negative associations. The value of 10 % is suggested based on a power analysis conducted in Veech (2013) but it can be modified by specifying a proportion to the `true_rand_classifier` parameter in `cooccur()` (i.e., the default is 0.1 and a more stringent value would be 0.05). The analysis also removed 14 species pairs because we used a threshold (i.e., `thresh = TRUE`) to filter from the results any species pairs that are expected to share less than 1 site.

Our next goal should be to inspect the pairwise results. A list of only significant species combinations can be obtained using the `print` method; however, to obtain the complete set of species pairs analyzed, use `prob_table()` or access the `$results` element of the `cooccur` object to retrieve the species pairs and their probabilities. See Table 2 for a description of the fields in the results table. In R this will give a warning to remind the user that because they applied a threshold the table does not represent all possible combinations of species. Below, we show the first six species pairs—a look at `p_lt` and `p_gt` shows that none of these species pairs were significantly associated, negatively or positively. For a given species pair, these two values represent the probabilities that those species could co-occur less than or greater than what is observed in our data, respectively. They can be interpreted as p-values, thus indicating significance levels for negative and positive co-occurrence patterns.

```
R> prob.table(cooccur.finches)

sp1 sp2 sp1_inc sp2_inc obs_cooccur prob_cooccur exp_cooccur    p_lt    p_gt
  1   2      14      13          11        0.630        10.7 0.87941 0.57941
  1   3      14      14          11        0.678        11.5 0.53529 1.00000
  1   4      14      10          10        0.484         8.2 1.00000 0.05147
  1   5      14      12          10        0.581         9.9 0.80882 0.67647
  1   6      14       2           1        0.097         1.6 0.33088 0.97794
  1   7      14      10          10        0.484         8.2 1.00000 0.05147

Warning message:
```

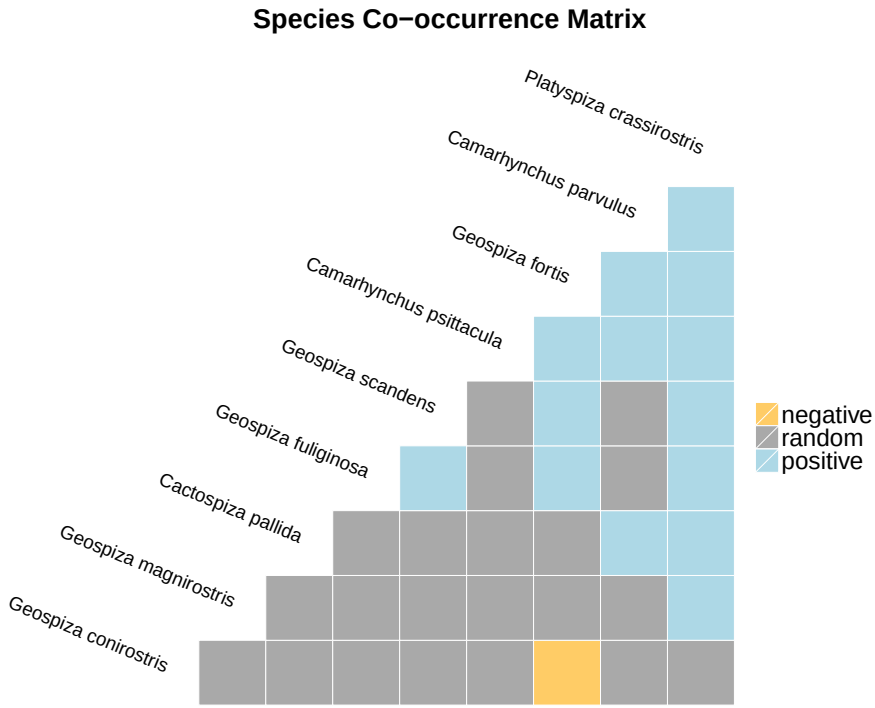

Figure 1: **ggplot2** heat map showing the positive and negative species associations determined by the probabilistic co-occurrence model for Galapagos finches. Species names are positioned to indicate the columns and rows that represent their pairwise relationships with other species.

```
In prob.table(cooccur.finches) :
  The co-occurrence model was run using 'thresh = TRUE.' The probability
  table may not include all species pairs
```

To assist in the interpretation and exploration of these large tables, use the `plot()` method on the results object. This will produce a visualization of all of the pairwise combinations of species and their co-occurrence signs (positive or negative) using a **ggplot2** heatmap. The plot trims out any species that do not have any significant negative or positive associations and orders the remaining species starting from those with the most negative interactions to those with the most positive interactions (left to right; Figure 1).

```
R> plot(cooccur.finches)
```

The probabilistic analysis finds mostly positive co-occurrence patterns among the finches in this dataset. From the finches heatmap it looks like *Geospiza fortis* has some interesting negative and positive associations with species so we will extract the results for this species. The `pair()` function can be used to specifiy a specific species, by name or number, to inspect—by default only significant results are shown, but if `all = TRUE` then all results will be shown.

```
R> pair(mod = cooccur.finches, "Geospiza fortis")
```

```
Species:
[1] "Geospiza fortis"
with 6 associations

             sp2 sp2_inc obs_cooccur prob_cooccur exp_cooccur    p_lt    p_gt
   G. fuliginosa      14          13        0.630        10.7 1.00000 0.00588
     G. scandens      12          12        0.540         9.2 1.00000 0.00210
  G. conirostris       2           0        0.090         1.5 0.04412 1.00000
   C. psittacula      10          10        0.450         7.6 1.00000 0.01471
     C. parvulus      10          10        0.450         7.6 1.00000 0.01471
P. crassirostris      11          11        0.495         8.4 1.00000 0.00630
```

To understand each species' individual contribution to the positive and negative species associations we need to create a pairing profile. The function `pair.attributes()` produces a table of the percentage of each species total pairings that were classified as positive, negative, and random (columns with prefix "num" are counts). Because the primary goal of this summary approach is to weight the degree of significant interactions (i.e., compare the numbers of positive versus negative associations), this version of the function treats unclassifiable pairings as random. These same results can be visualized across all species by using the function `pair.profile()` to create a box plot of these percentages. This plot will show the percent of species pairs that were positive, negative, and random for all species. This plot can easily communicate whether or not species tend to have mostly negative or mostly positive interactions. It will also suggest whether these interactions are evenly distributed among the species as opposed to being clustered in a few species (Figure 2). This pairing summary and the pairwise probability table can be used in downstream analyses (e.g., combination with phylogenetic data or correlation with trait and resource use differences). Obtaining effect sizes for use in these analyses is described in the next section.

```
R> pair.attributes(cooccur.finches)

  pos   neg   rand num_pos num_neg num_rand                 sppname
 9.09  0.00  90.91       1       0       10    Geospiza magnirostris
45.45  9.09  45.45       5       1        5          Geospiza fortis
27.27  0.00  72.73       3       0        8       Geospiza fuliginosa
 0.00  0.00 100.00       0       0       11       Geospiza difficilis
27.27  0.00  72.73       3       0        8         Geospiza scandens
 0.00 11.11  88.89       0       1        8      Geospiza conirostris

R> pair.profile(cooccur.finches)
```

Effect sizes can also be calculated from co-occurrence analyses; they allow for comparisons among studies and methods as well as providing a quantitative measurement of co-occurrence for use in downstream analyses. In the context of the probabilistic co-occurrence analysis from Veech (2013) effect sizes are the differences between expected and observed frequency of co-occurrence. These values can be standardized by dividing these differences by the number of sampling sites in the dataset. In standardized form, these values are bounded from -1 to 1, with positive values indicating positive associations and negative values indication negative

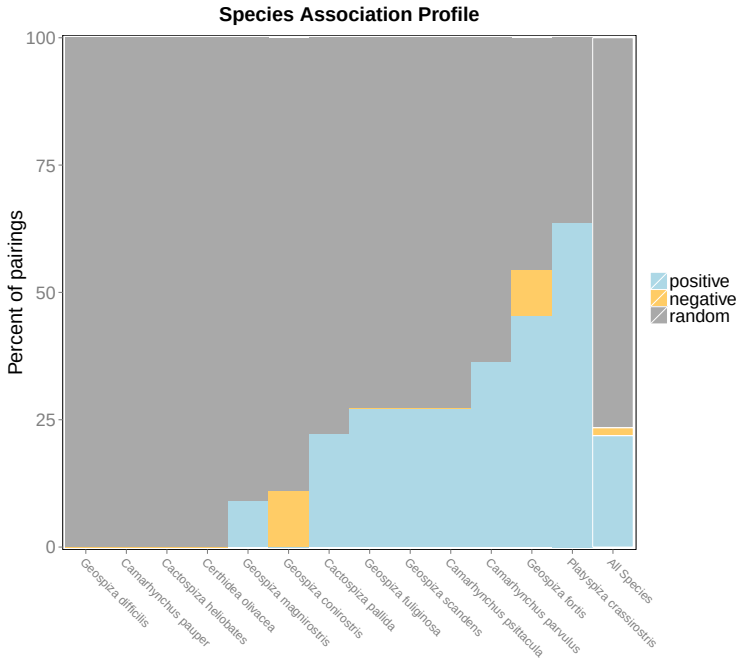

Figure 2: Boxplot showing the percent of total pairings for each species that are positive, negative, or random. Species are ordered by increasing number of total associations. The right-most bar, outlined in white, represents the assemlage-wide percentages.

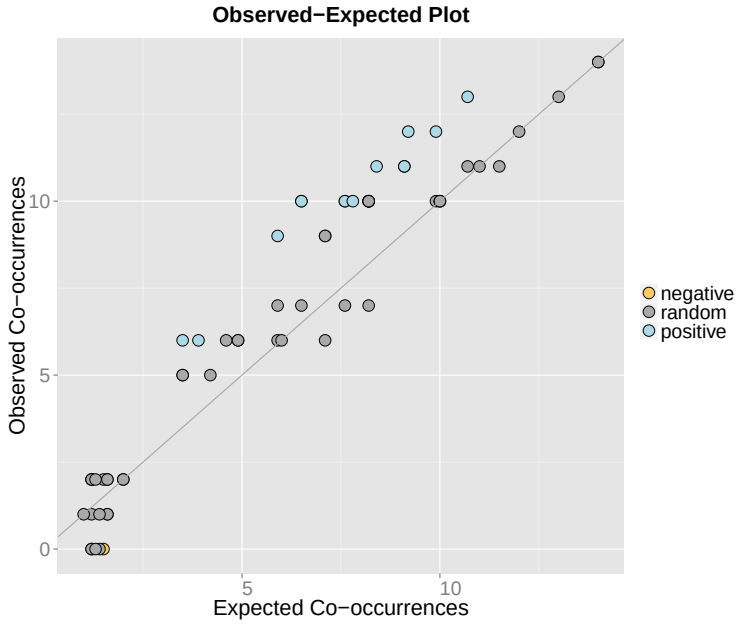

Figure 3: Observed versus expected co-occurrence scatter plot. Each species pair in the analysis is represented by a point colored based on whether it was classified as postive, negative, or random.

associations. The function `effect.sizes()` can be used to extract these effects sizes from a `cooccur` object, standardize them if desired (`standardized = TRUE`), and return them as either a pairwise table or as a species by species matrix (`matrix = TRUE`; perhaps for comparison to a trait distance matrix). However, in the case of the finches dataset, we have conducted our analysis using a threshold that removes species combinations not expected to co-occur in more than 1 site. To avoid running the probability calculations again (which can be time consuming for larger datasets) for the entire dataset and then using `effect.sizes()`, we can simply specify `only_effects = TRUE` in `cooccur()` which will bypass the probability calculations and quickly return effects sizes. Make sure to specify `thresh = FALSE` if all combinations are desired.

```
R> cooccur(mat = finches, type = "spp_site", thresh = FALSE,
+    spp_names = TRUE, only_effects = TRUE, eff_standard = TRUE,
+    eff_matrix = TRUE)
```

```
       1     2     3     4     5     6     7     8     9    10    11    12
2   0.02
3  -0.03  0.14
4   0.11 -0.04 -0.07
5   0.01  0.16  0.12 -0.06
6  -0.04 -0.09 -0.04 -0.01 -0.08
7   0.11  0.14  0.11  0.01  0.11 -0.07
8   0.01  0.01  0.01  0.02  0.02 -0.01  0.02
9   0.11  0.14  0.11  0.06  0.11 -0.07  0.18  0.02
10  0.11  0.15  0.11  0.03  0.13 -0.08  0.21  0.02  0.21
11  0.06  0.08  0.06  0.09  0.05 -0.04  0.09 -0.02  0.15  0.12
12  0.02  0.03  0.02  0.05 -0.02 -0.01  0.05 -0.01  0.05  0.04  0.08
13  0.00  0.00  0.00  0.00  0.00  0.00  0.00  0.00  0.00  0.00  0.00  0.00
```

To inspect the degree to which finch species pairs deviate from their expected co-occurrence levels, plot the observed values against the expected value as a visual diagnostic. This can be done using the code below (Figure 3). The probability calculations are based on the number of sites and the individual frequencies of occurrence and co-occurrence for each species pair. Therefore the conditions determining statistical power change with sample size and it is valuable to examine effect sizes for species pairs regardless of statistical significance. A detailed discussion of power, Type I and II error rates, and a comparison with other methods can be found in Veech (2013).

```
R> obs.v.exp(cooccur.finches)
```

In this plot one can clearly see that there are few species pairs in this dataset that exhibit fewer than expected co-occurrences. The pairs that are less than expected, including one pair classified as having a negative association, are largely clustered towards having low expected co-occurrences in the first place. This is an interesting result given that previous analyses of these data have often revealed negative associations Sanderson (2000). Our analysis of the finch dataset using the probabilistic model of species co-occurrence reveals primarily positive species associations. Also, these results are presented in a form that makes them easy to pipeline into downstream analyses and compare to results using other methods.

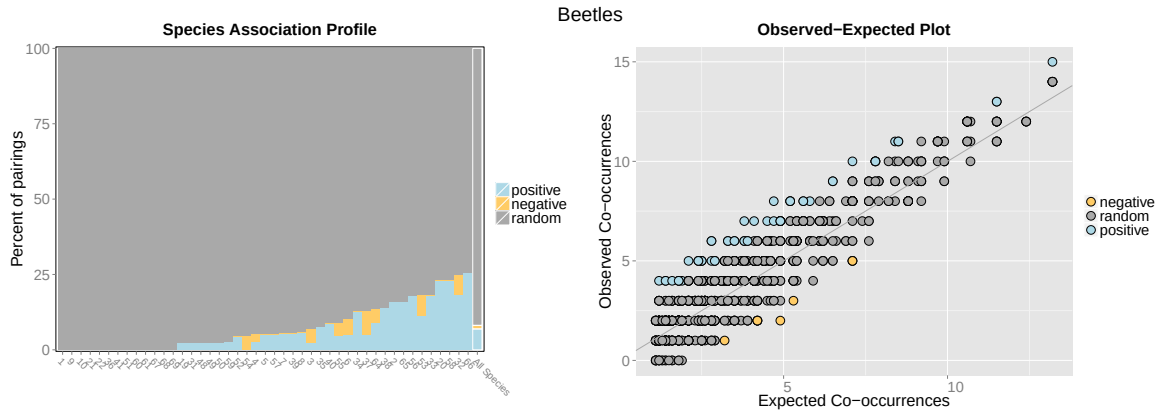

Figure 4: Combined species association profile and observed versus expected plot for the `beetles` dataset. See text and previous figures for details.

2.4. Further examples: Beetles and rodents

For purposes of comparison we have provided two other datasets that can be analyzed in the same way as the finches dataset. The beetles dataset contains occurrence information on 71 species of beetles that were sampled across 17 different sites (Ulrich and Zalewski 2006; Gotelli and Ulrich 2010). The rodents dataset comes from Brown and Kurzius (1987) who present rodent occurrences in different North American desert regions, including the Great Basin (used here) with 16 species across 39 sites. Both datasets are similarly structured as species by site matrices. In analyzing these data we will make our goal to enumerate every possible species combination in the original datasets. Therefore we do not want to apply the default threshold (expected co-occurrences ≥ 1, see Table 2) which would remove species pairs from the analysis. The reason for this would be to facilitate merging pairwise data with other data sources as well as comparisions with randomization techniques and integrating these results into studies of macroecology and community ecology.

```
data("rodents")
data("beetles")
R> cooc_mod <- lapply(list(beetles, rodents),
+    FUN = function(x) cooccur(mat = x, thresh = FALSE))
R> cooccur.beetles <- cooc_mod[[1]]
R> cooccur.rodents <- cooc_mod[[2]]
```

Finally, it is informative to compare the pairs profile and observed-expected plots for these test datasets. Figure 4 shows these plots for the beetles dataset whereas Figure 5 shows them for the rodents (using **gridExtra** from Auguie 2015).

```
R> library("gridExtra")
R> grid.arrange(pair.profile(cooccur.beetles), obs_v_exp(cooccur.beetles),
+    ncol = 2,main = textGrob("Beetles", gp = gpar(cex = 2), just = "top",
+    vjust = 0.75))
R> grid.arrange(pair.profile(cooccur.rodents), obs_v_exp(cooccur.rodents),
+    ncol = 2,main = textGrob("Rodents", gp = gpar(cex = 2), just = "top",
+    vjust = 0.75))
```

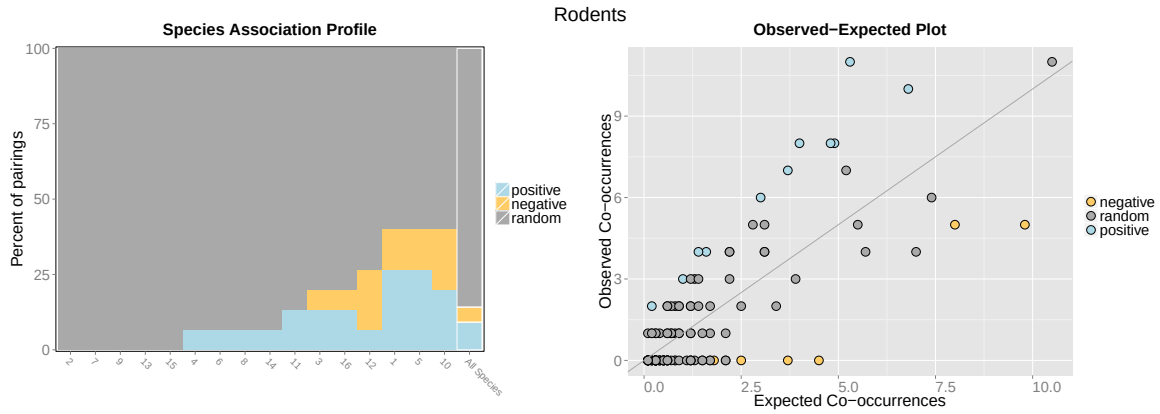

Figure 5: Combined species association profile and observed versus expected plot for the `rodents` dataset. See text and previous figures for details.

The results for these two analyses show more negative relationships than does the finches dataset. The beetles dataset has more pairwise combinations and fewer negative interactions than the rodents but the rodents data seem to have some particularly large effect sizes. The three datasets used here as examples are available in the R package and the code in this manual can be used to recreate these analyses and then to analyze other datasets. The next section describes model performance, justification for aspects of our software implementation, and discussion regarding estimates of runtime.

3. Model performance and development

3.1. Model performance and run-time

The probabilistic model of co-occurrence relies on combinatorics to produce exact probabilities. To analyze reasonably sized ecological datasets it must handle very large integers with high precision–this is especially true when using Equation 1 of Veech (2013) compared to the hypergeometric approach using the improved Equation 1. By default **cooccur** uses the faster hypergeometric approach (`cooccur(prob = "hyper")`) but the original, slower approach is still available in the `cooccur()` function by specifying `prob = "comb"`. In order to satisfy the need for calculating exact probabilities and simultaneously analyzing large datasets with the original approach we used the package **gmp** to implement all combinatorics algebra. **gmp** is used to access the GNU Multiple Precision Arithmetic libraries and perform arbitrary-precision operations. Arbitrary-precision refers to the property that the length of integers are only limited by the RAM on the computer running the software. This is a fundamental improvement in the quality of the probability calculations compared to using the **base** R combinatorics implementations but comes at a significant cost to processing speed. The tradeoff is such that above an approximately 3500 site dataset the analysis is no longer feasible with the **base** installation function `choose()` because it cannot store integers of adequate length. We use the package **gmp** because it provides for the analysis of the maximum number of sample sites allowable by the user's computer memory and also returns exact calculations of p-values across the entire range of possible sample sizes.

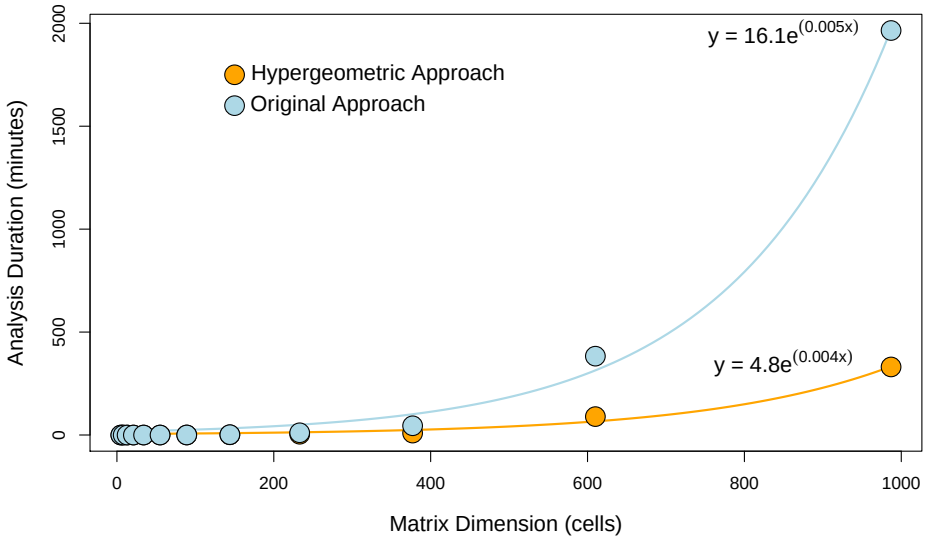

Figure 6: Runtime for the `cooccur()` function across a range of input data dimensions (i.e., number of row and columns, or species and sites) for square matrices randomly assigned occupancy patterns that maximize computational requirements. The software was run on a 64 bit system with 16 GB RAM and 2.70 GHz CPU. The regression equation can be used to estimate the worst case runtime for a potential analysis.

The default approach, using the hypergeometric distribution to calculate probablities, is far less computationally demanding than the origninal approach. To provide the reader with a relevant evaluation of the runtime of the `cooccur()` function we conducted analyses, using both approaches, with simulated data matrices of increasing dimensions. We used square matrices where each species' site occupancy was assigned individually. The limiting step in the original analysis is the function `chooseZ()` which is the arbitrary precision implementation of the binomial coefficient operation $C(n, k)$, or the number of ways to choose k elements out a set of n elements. We assigned species occupancy with a probability of 0.5 because $k = 0.5n$ maximizes $C(n, k)$ and therefore the computational strain on `chooseZ()`. Matrices with dimensions following the Fibonacci sequence from 5 x 5 to 987 x 987 were created in this manner to represent worst-case runtimes for datasets of increasing sample size. Runtime is plotted against matrix dimension in Figure 6. Runtime increases exponentially with the size of the dataset but is much faster with the implementation using the hypergeometric distribution. Figure 6 can be used to project runtimes for large datasets.

3.2. Future extension and developement of the cooccur package

Like most other co-occurrence analyses in ecology, the probabilistic model of (Veech 2013) focuses on pairwise comparisons of species. As such, many common analyses can be done using the results of the the `cooccur()` model. For example, recent papers have tested for differences in co-occurrence patterns among invasive and native plant species (Carboni, Münkemüller, Gallien, Lavergne, Acosta, and Thuiller 2013), fire tolerant and intolerant woody plants (Silva

and Batalha 2010; Cardillo 2012), and community assembly processes in Neotropical birds (Gómez, Bravo, Brumfield, Tello, and Cadena 2010) using primarily randomization (null model) based approaches. Future extensions to our software could assist users with interfacing the randomization-free results of the probabilistic co-occurrence model with trait, community, and phylogenetic data. Similarly, extensions could also provide a framework for hypothesis testing within these types of studies. Furthermore, we hope to build on the pairwise model to allow for detection of co-occurrences of groups of species. We are also investigating ways of redefining the total set of sampling sites (in a dataset) to take into account that sites may not all be equiprobable in having each species. In addition, the graphical display options and analytical tools in **cooccur** could be made compatible with the outputs of other R packages to extend our visualizations and analyses to other approaches that don't often include means to directly inspect co-occurrence results. Finally, we intend on updating and maintaining these tools and are happy to correspond with users.

References

Auguie B (2015). ***gridExtra**: Functions in Grid Graphics*. R package version 2.0.0, URL https://CRAN.R-project.org/package=gridExtra.

Brown JH, Kurzius MA (1987). "Composition of Desert Rodent Faunas: Combinations of Coexisting Species." *Annales Zoologici Fennici*, **24**, 227–237.

Carboni M, Münkemüller T, Gallien L, Lavergne S, Acosta A, Thuiller W (2013). "Darwin's Naturalization Hypothesis: Scale Matters in Coastal Plant Communities." *Ecography*, **36**, 560–568. doi:10.1111/j.1600-0587.2012.07479.x.

Cardillo M (2012). "The Phylogenetic Signal of Species Co-Occurrence in High-Diversity Shrublands: Different Patterns for Fire-Killed and Fire-Resistant Species." *BMC Ecology*, **12**, 21. doi:10.1186/1472-6785-12-21.

Fridley JD, Vandermast DB, Kuppinger DM, Manthey M, Peet RK (2007). "Co-Occurrence Based Assessment of Habitat Generalists and Specialists: A New Approach for the Measurement of Niche Width." *Journal of Ecology*, **95**, 707–722. doi:10.1111/j.1365-2745.2007.01236.x.

Gómez JP, Bravo GA, Brumfield RT, Tello JG, Cadena CD (2010). "A Phylogenetic Approach to Disentangling the Role of Competition and Habitat Filtering in Community Assembly of Neotropical Forest Birds." *Journal of Animal Ecology*, **79**, 1181–1192. doi:10.1111/j.1365-2656.2010.01725.x.

Gotelli NJ, Ellison AM (2013). **EcoSimR**. *Version 1.00*. Burlington, VT. URL http://www.uvm.edu/~ngotelli/EcoSim/EcoSim.html.

Gotelli NJ, Ulrich W (2010). "The Empirical Bayes Approach as a Tool to Identify Non-Random Species Associations." *Oecologia*, **162**, 463–477. doi:10.1007/s00442-009-1474-y.

Hoagland BW, Collins SL (1997). "Gradient Models, Gradient Analysis, and Hierarchical Structure in Plant Communities." *Oikos*, **78**, 23–30. doi:10.2307/3545796.

Kembel SW, Cowan PD, Helmus MR, Cornwell WK, Morlon H, Ackerly DD, Blomberg SP, Webb CO (2010). "**picante**: R Tools for Integrating Phylogenies and Ecology." *Bioinformatics*, **26**, 1463. doi:10.1093/bioinformatics/btq166.

Lucas A, Scholz I, Boehme R, Jasson S, Maechler M (2014). ***gmp**: Multiple Precision Arithmetic*. R package version 0.5-12, URL https://CRAN.R-project.org/package=gmp.

Oksanen J, Blanchet FG, Kindt R, Legendre P, Minchin PR, O'Hara RB, Simpson GL, Solymos P, Stevens MHH, Wagner H (2016). ***vegan**: Community Ecology Package*. R package version 2.3-3, URL https://CRAN.R-project.org/package=vegan.

R Core Team (2015). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. URL https://www.R-project.org/.

Sanderson JG (2000). "Testing Ecological Patterns: A Well-Known Algorithm from Computer Science Aids the Evaulation of Species Distributions." *American Scientist*, **88**, pp. 332–339.

Sfenthourakis S, Tzanatos E, Giokas S (2005). "Species Co-Occurrence: The Case of Congeneric Species and a Causal Approach to Patterns of Species Association." *Global Ecology and Biogeography*, **15**, 39–49. doi:10.1111/j.1466-822x.2005.00192.x.

Silva IA, Batalha MA (2010). "Woody Plant Species Co-Occurrence in Brazilian Savannas under Different Fire Frequencies." *Acta Oecologica*, **36**, 85–91. doi:10.1016/j.actao.2009.10.004.

Ulrich W (2008). `Pairs` – *A Fortran Program for Studying Pair-Wise Species Associations in Ecological Matrices*. Torun, Poland. URL ftp://raksti.daba.lv/pub/GIS/datu_analiize/UlrichW/PairsManual.pdf.

Ulrich W, Zalewski M (2006). "Abundance and Co-Occurrence Patterns of Core and Satellite Species of Ground Beetles on Small Lake Islands." *Oikos*, **114**, 338–348. doi:10.1111/j.2006.0030-1299.14773.x.

Veech JA (2013). "A Probabilistic Model for Analysing Species Co-Occurrence: Probabilistic Model." *Global Ecology and Biogeography*, **22**, 252–260. doi:10.1111/j.1466-8238.2012.00789.x.

Wickham H (2007). "Reshaping Data with the **reshape** Package." *Journal of Statistical Software*, **21**(12), 1–20. doi:10.18637/jss.v021.i12.

Wickham H (2009). ***ggplot2**: Elegant Graphics for Data Analysis*. Springer-Verlag, New York.

Zhang J (2013). ***spaa**: SPecies Association Analysis*. R package version 0.2.1, URL https://CRAN.R-project.org/package=spaa.

A. Functions and methods for class cooccur

Function	Description
cooccur	This function takes a community dataset (data frame or matrix) of species by site presence-absence data and classifies species pairs as having positive, negative, and random associations based on the probabilistic model of specie co-occurrence from Veech (2013). It produces an object of class cooccur.
effect.sizes	Calculate standardized and raw effect sizes from an object of class cooccur.
obs.v.exp	Plot the observed number of co-occurrences versus the number expected from the probability analysis in a cooccur object.
pair	Extracts results for a single species from a cooccur object.
pair.attributes	Summarizes the positive, negative, and random interactions for each species in an cooccur analysis.
pair.profile	Plots a bar plot for visualizing the associations of each individual species from a cooccur object.
plot.cooccur	Heatmap visualization of the pairwise species associations revealed by a cooccur analysis.
print.cooccur	Returns a table of analysis results for all significant pairwise interactions found in a cooccur object.
prob.table	Returns a results table for all analyzed species pairs in a cooccur object.
summary.cooccur	Presents a count of positive, negative, random, and unclassified pairwise comparisons from a cooccur object.

Table 3: Definitions for functions included in the **cooccur** package. Besides function cooccur(), all functions take a cooccur object as input.

Affiliation:

Daniel M. Griffith
Department of Biology
Wake Forest University
NC, 27109, United States of America
E-mail: griffith.dan@gmail.com
URL: http://danielmgriffith.wordpress.com/

Journal of Statistical Software http://www.jstatsoft.org/
published by the Foundation for Open Access Statistics http://www.foastat.org/
February 2016, Volume 69, Code Snippet 2 *Submitted:* 2014-03-28
doi:10.18637/jss.v069.c02 *Accepted:* 2015-02-04