Integral occurrence probability: combining cover and relative shoot frequencies based on bounded point-to-plant distances

<table>
<thead>
<tr>
<th>Journal:</th>
<th>Journal of Vegetation Science</th>
</tr>
</thead>
<tbody>
<tr>
<td>Manuscript ID</td>
<td>JVS-RA-03699.R2</td>
</tr>
<tr>
<td>Manuscript Type:</td>
<td>Research article</td>
</tr>
<tr>
<td>Date Submitted by the Author:</td>
<td>n/a</td>
</tr>
<tr>
<td>Complete List of Authors:</td>
<td>Van Calster, Hans; Institute for Forest and Nature Research, Damgaard, Christian; Aarhus University, Department of Terrestrial Ecology</td>
</tr>
<tr>
<td>Keywords:</td>
<td>Plant abundance, Empty-space distance, Inflated beta distribution, Point intercept sampling, Plotless density estimation, Spatial point patterns, Cover, Relative frequency, Integral occurrence probability</td>
</tr>
</tbody>
</table>
Integral occurrence probability: combining cover and relative shoot frequencies based on bounded point-to-plant distances

Hans Van Calster, Christian Damgaard

Van Calster, H. (corresponding author, hans.vancalster@inbo.be

Damgaard, C. (cfd@dmu.dk)

1 Biometry and Quality Assurance, Research Institute for Nature and Forest (INBO), Kliniekstraat 25, 1070 Brussel, BE;

2 Department of Bioscience, Aarhus University, Vejlsøvej 25, 8600 Silkeborg, DK

Abstract

Aims: To introduce a new distance-based field method for (herbaceous, terrestrial) plant species that relates cover to relative shoot frequency as a continuous process of occurrence probabilities and explain how these data can be analyzed.

Methods: We propose to measure shortest distances from a sample of sampling points to the nearest aboveground part of plant species (up to a maximum search distance). We show how, after appropriate transformation of the point-to-plant distances to a 0-1 interval scale, cover as well as relative shoot frequency at any area up to the searched area can be read from the same curve. This leads to the notion of an integral occurrence probability, which we propose as a new species abundance measure. For estimation and regression modelling we make use of the zero-and-one inflated beta distribution. We supply all code required for these analyses.

Results: Simulations of plant distribution patterns showed that the integral occurrence probability is able to differentiate between plant abundance patterns that differed in terms of relative cover, density and type of spatial distribution pattern. It is more sensitive to these differences than either cover or relative shoot frequency alone. The method allows summing occurrence probabilities over species to predict expected species richness as a function of the area searched. Aggregation across species while accounting for overlap in species spatial distributions is a simple matter of taking the minimum among the point-to-plant distances at each sampling point. The latter was demonstrated with data from a field trial in Nardetea grassland.

Conclusions: The method may be a viable alternative for currently employed field methods, such as visual cover estimates, point-intercept sampling and recording the frequency of plant species in equal-area plots. Applications include, but are not limited to, conservation management monitoring and ground-truthing of remote sensing data.
Keywords
Plant abundance; Empty-space distance; Cover; Relative frequency; Integral occurrence probability; Inflated beta distribution; Point intercept sampling; Plotless density estimation; Spatial point patterns

Nomenclature
Lambinon et al. (1998) for vascular plants

Abbreviations
CDF: cumulative distribution function; IOP: integral occurrence probability; PDF: probability density function

Running head
Sampling vegetation with point-to-plant distances

Introduction
Vegetation is characterized by its species composition on the one hand and the relative differences in abundance of the constituting species on the other (Kent & Coker 1992). There are two basic attributes that measure abundance (Morrison et al. 1995): density (i.e. when the interest is in the number of individuals or other types of countable units, e.g. shoots, of a species per unit area) and biomass (which is the product of density and the average weight). Field methods that directly measure these basic attributes are too time-consuming and impractical in applied contexts. Therefore, a plethora of more practical field methods are used by field ecologist to describe plant species abundances (Kent & Coker 1992; Elzinga et al. 1998; Bonham 2013).

Field methods based on measurements of distances are well known in the context of density estimation. They fall within the family of plotless sampling techniques and have as main advantage their speed of application (Parker 1979; Engeman et al. 1994; Dobrowski & Murphy 2006). These techniques are used mainly in forestry to measure the density of tree stems (overall, per species, per size class, ...). Other well-known plotless sampling techniques in forestry, such as Bitterlich sampling, allow estimation of basal area, which is the cover by the bases of trees or tussocks, at the ground or at breast height (Gregoire & Valentine 2008; Wilson 2011). These field methods work because individual trees in a forest are easily recognizable, they can be treated as points in the case of density estimation, and the observer is small compared to the size of trees.

In this paper we introduce a distance-based field method that differs in important respects from existing methods, making it suitable for herbaceous plant communities: (i) its focus is on estimation of cover and relative shoot frequency, i.e. we measure the distance to the nearest aboveground part of the plant (henceforth: point-to-plant distances) instead of the distance to where a plant is rooted and (ii) we restrict the search up to a maximum search distance (i.e. it is not a plotless method).
Wilson (2011) defines cover as the proportion of ground that would be covered by a particular species were all other species removed, i.e. the vertical projection of its leaves (etc.) and overlapping leaves. With point-intercept (or pin-point) sampling, a random sample is drawn and it is noted whether a plant species is present or absent from the sampling location (a point). In practice, a pin or rod is often used which is lowered into the vegetation and the species that touch the pin at least once are noted. The number of points where a species is present divided by the total number of sampling points is an estimate of the cover of a species in the area sampled. It is the most unambiguous measure for cover of a plant species, provided the pin cross-sectional area is very small (Wilson 2011). Other techniques to estimate cover are based on visual estimates of cover in quadrats with specific recording scales (e.g. Daubenmire, Braun-Blanquet scale) (Kent & Coker 1992).

Relative frequency is usually estimated from a random sample of circular or square sampling units of fixed size. The presence or absence of the species is noted. The definition of presence depends on the type of frequency estimate. Rooted frequency refers to the situation where a species is noted as present if it is rooted within the sampling unit. It is related to density estimates (Bonham 2013). Cover-based or shoot frequency is the situation where a species is present if the vertical projection of any of the aboveground parts of the species falls within the sampling unit. It is related to biomass. Shoot frequency, when determined with the aid of circular sampling units, and cover, as determined by the point-intercept method, are related. Imagine that the pin, which is lowered into the vegetation, has an increasingly larger cross-sectional area. Only when the cross-sectional area is very small (almost zero), cover and shoot frequency coincide. When the cross-sectional area becomes larger, we obtain an estimate of shoot frequency that must be larger than the estimate of cover (overestimation). This reveals a known drawback of relative frequency: its value depends on the size of the sampling unit. The optimal size (in terms of sampling design considerations) is species dependent due to differences in the average size of individual plants, the rarity and spatial patterning of species (Elzinga et al. 1998; Bonham 2013). A way out is to consider multiple concentric sampling units of increasing size (Böcher 1935; Aberdeen 1958; Morrison et al. 1995; Damgaard 2015b).

Morrison et al. (1995) used simulations and a field study to assess the efficacy of a compounded set of seven nested, concentrically placed square sampling units (subquadrats) compared to square sampling units of fixed size. A subquadrat does not contain the area of the smaller subquadrats, and the cumulative sizes of subquadrats increased geometrically. In one scenario, they calculated a frequency-score, which entailed that a species list needed to be made for each of the subquadrats. In another scenario, an importance-score was calculated that only required noting new species, as the subquadrats were searched from the inner-most to the outer-most subquadrat. The latter had the considerable benefit that no more time was required compared to the traditional technique using sampling units of fixed size. They used rooted frequency in their study. One of their important findings was that these estimators have a more direct relationship with density over a much larger range of density values compared to the traditional technique. They also found that the method was less sensitive to the spatial distribution of the species. The latter is in agreement with Aberdeen (1958) who demonstrated that a minimum of three quadrat sizes are needed if frequency alone is to be used to discriminate between random and non-random distributions.
Damgaard (2015b) refers to Böcher (1935), who described a very similar method, this time using a set of concentrically placed circular sampling units. For each species, Böcher (1935) also noted in which of the sampling units it was first encountered (shoot frequency) when searching from the center outwards (analogous to the importance score of Morrison et al. (1995)). Whereas Böcher (1935) and Morrison et al. (1995) used ordinal scores to discuss the differences in frequency between species, Damgaard (2015b) shows how these data can be statistically modelled and can be used to estimate: (i) the occurrence probability (= relative frequency) in an area equal to the smallest subplot, and (ii) the intra-plot correlation (or within-site spatial aggregation).

The method introduced in this paper can be seen as a continuous analogue of the field methods that estimate relative frequency by concentrically placed sampling units of different size (Morrison et al. 1995; Damgaard 2015b). The method is equally well applicable to other sedentary organisms or any other spatially distributed attributes that can be considered more or less fixed in the timeframe of sampling.

We will first explain the statistical treatment of the data to show that point-to-plant distances allow calculating (i) cover as well as relative frequency and (ii) a new summary statistic, the integral occurrence probability (IOP), which is sensitive to changes in both cover and relative frequency. Next, a simulation study demonstrates the validity of the method. The simulation experiment assesses the dependency of cover, relative frequency and IOP on the spatial distribution characteristics of simulated species. Each simulated species is the result of the type of point pattern (regular, random, clustered), the intensity of the point pattern (~ determining density), and the size of the plants. These three factors combined determine the true cover of a simulated species, i.e. the cover of the species in the areal sampling frame without measurement error. These simulations will also be used to show how predictions of separate species models can be assembled to predict expected species richness as a function of incremental search area. Finally, a field trial in Nardetea grassland was designed to showcase (i) how the method performs in practice and (ii) how point-to-plant distances recorded for individual species can be aggregated across species that belong to the same species groups.

Material and methods

The distribution of bounded point-to-plant distance data

We measure the shortest distance \( r_{ij} \) from a sampling point, \( i \), to the nearest aboveground part of a plant species, \( j \), up to a predefined maximum search distance \( r_{\text{max}} \). This is to be repeated for a set of sampling points (a sample with sample size \( n \)) and all (or a selection of) species. We refer to these measurements as point-to-plant distances, but a more generic name is empty-space distance (Baddeley et al. 2015 (chapter 8)).

The possible values, \( r_{ij} \), range from 0 to \( r_{\text{max}} \). To explain the link between bounded point-to-plant distances and cover or relative shoot frequency, we transform these values as follows (to simplify, we leave out indices \( i \) and \( j \) from the formulae):

\[
    y = \frac{\pi r^2}{\pi r_{\text{max}}^2} = \frac{r^2}{r_{\text{max}}^2}
\]
If the species is present at the sampling point, \( r = 0 \) will be recorded, which remains 0 after transformation. If the species is not observed within the maximum search distance, we set \( r = r_{\text{max}} \) and the transformed value is 1. This transformation reflects the fact that if we measured a distance \( r \), we have searched an incremental area up to \( \pi r^2 \), where the species is absent. The division by \( \pi r_{\text{max}}^2 \) turns this into continuous values between 0 and 1 (exact 0 and 1 included).

The beta distribution is suitable to fit bounded continuous data between 0 and 1, but it does not allow values that are exactly 0 or 1 (Ospina & Ferrari 2010). It is closely related to the binomial distribution for discrete data. The shape of the distribution is very flexible and is governed by two parameters (Bolker 2008). It includes the uniform distribution as a special case and the normal distribution as a limiting case. But it can also be right or left skewed, or peaked at one or both of the boundaries. In a series of recent publications, the beta distribution and its variants appear prominently in the modelling of plant abundance data. For instance for the modelling of pin-point cover data (Chen et al. 2006; Chen, Shiyomi, Hori, et al. 2008; Chen, Shiyomi, Bonham, et al. 2008; Damgaard 2009; Damgaard 2012; Damgaard 2013; Damgaard 2014; Damgaard 2015a), but also for the modelling of visually estimated cover data (Damgaard 2009; Damgaard 2014; Herpigny & Gosselin 2014). A solution to the problem of exact 0 and 1 values is a mixture of a discrete Bernoulli distribution and a continuous beta distribution, which results in a beta distribution with probability masses at 0 and 1 (Ospina & Ferrari 2010).

This zero-and-one inflated beta distribution (Ospina & Ferrari 2010) can be parameterized as follows:

\[
\text{beinf}(y; p_0, p_1, \mu, \Phi) = \begin{cases} 
  p_0 & \text{if } y = 0 \\
  p_1 & \text{if } y = 1 \\
  (1 - p_0 - p_1) \frac{\Gamma(\Phi)}{\Gamma(\mu)\Gamma(1-\mu)\Phi} y^{\mu-1}(1-y)^{(1-\mu)\Phi-1} & \text{if } y \in ]0,1[ 
\end{cases}
\]

The parameter \( 0 < \mu < 1 \) is the mean of the beta part and \( \phi > 0 \) is a precision parameter. For a given mean, larger values of the precision parameter give higher precision (lower variance). This distribution is implemented in the gamlss package in R (Rigby & Stasinopoulos 2005) and in the ZOIB package in R (Liu & Kong 2015). Each package uses a different parametrization, but they can be re-parameterized (see Appendix S1).

We can fit the distribution to the transformed point-to-plant distances for each species separately. To see the similarity with the point-intercept method, we note that \( p_0 \) equals the proportion of \( r = 0 \) (the number of times a species was measured with distance 0 divided by the total number of sampling points), which is an estimate of cover like in point-intercept methods.

To see the similarity with shoot frequency, note that the proportion of ones \( (p_1) \) equals the proportion of sampling points where the species was not found within a circle with radius \( r_{\text{max}} \) centered at the point. One minus this proportion is thus an estimate of the relative (shoot) frequency in these circles. Now we can ask, what is the relative frequency in circular plots with \( 0 < r < r_{\text{max}} \)? This piece of information is given by the parameters that govern...
the shape of the beta part of the distribution. Figure 1 gives an example of the PDF and the CDF for a zero-and-one inflated beta distribution with \( p_0 = 0.31, p_1 = 0.06, \mu = 0.2 \) and \( \phi = 10.1 \). We need the CDF to get the predicted shoot frequency at any value of \( r \).

**Figure 1:** Example of a PDF and CDF for a zero-and-one inflated beta distribution with \( \mu = 0.2, \phi = 10.11, p_0 = 0.31 \) (point mass probability at 0 = cover) and \( p_1 = 0.063 \) (point mass probability at 1 = absence probability, i.e. the probability that the species is not observed at \( r < r_{\text{max}} \)) (full lines). When \( r^2/r^2_{\text{max}} = 1 \) (the maximum possible value), y-axis is 1 (by definition of the CDF). This value has no specific biological meaning, which is why we do not show the jump to 1. When \( r^2/r^2_{\text{max}} \) approaches 1, the y-axis of the CDF is \( 1 - p_1 \) or, the relative frequency obtained when sampling units are circles of radius (almost) \( r_{\text{max}} \). The dashed line and grey 95% confidence bounds, obtained via bootstrapping, give the CDF for one realisation of sample size 100. The point estimates and 95% error bars (dark grey) give the corresponding estimates if presence - absence of the species had been determined in a series of circular plot sizes. The figure demonstrates that cover (the value along the y-axis where the x-axis is 0) and relative shoot frequency are related, as a continuous process of occurrence probabilities.

If we let \( x \) take on values \( \in [0,1] \), the CDF of the mixture distribution can be expressed as follows:

\[
P(y = \frac{r^2}{r^2_{\text{max}}} \leq x) = p_0 + (1 - p_0 - p_1) \int_0^x f_y(t) \, dt
\]

The integral represents the CDF for the continuous part of the data that is beta-distributed and \( f_y(\cdot) \) is the PDF for the beta-distribution. Thus, for \( x = 0 \), this is the partial area under the PDF curve at the probability mass at 0. This means that the CDF at \( x = 0 \) equals \( p_0 \) \( (f_y(0) = 0) \). Similarly, evaluating the CDF when \( x \to 1 \) equals \( 1 - p_1 \) (the partial area under the PDF when \( x \) approaches one). These observations imply that cover and relative shoot frequency can be recovered from the CDF. The CDF predicts the relative shoot frequency at
a particular value for $r$, or, in other words, it indicates how large a circular sampling unit needs to be in order to observe the species with the corresponding occurrence probability. These observations make it clear that both cover and relative frequency are occurrence probabilities, thus, the curve can be read as follows:

- if $r^2 / r_{\text{max}}^2 = 0$, y-axis = an estimate for cover (occurrence probability in a point)
- if $0 < r^2 / r_{\text{max}}^2 < 1$, y-axis = relative (shoot) frequency (= occurrence probability) when the sampling unit is a circle with radius $r$.

Based on the above example with known true values for the parameters, we now turn to the case of obtaining inferences from a random sample from this underlying true distribution. In this case, confidence bounds around the CDF curve reflect the uncertainty due to sampling. One approach to construct confidence bounds is bootstrapping. We show this in Figure 1 for a sample of size $n = 100$ from the zero-and-one inflated beta. To illustrate the interpretation of the CDF curve, suppose that instead of measuring the distance from the set of sampling points, we determine the presence or absence of the species in a set of circular sampling units with radius $r$, centered at the same set of sampling points. We repeat this for a range of radii from 0 to $r_{\text{max}}$. From these sets of presence - absence data, we calculate the probability of occurrence (relative frequency) with a logistic regression. The results are also indicated in Figure 1. The fitted values and 95% confidence intervals follow the fitted CDF curve and clearly demonstrate that our interpretation holds.

A simple example illustrating with R-code how the zero-and-one inflated beta distribution can be fit to transformed point-to-plant distance data is given in Appendix S1.

**Integral occurrence probability: a new measure that combines the concepts of cover and relative frequency**

The CDF function has all the information that relates cover and relative frequency to $r$ ($< r_{\text{max}}$). Both aspects are important: high relative frequency can coincide with both low and high cover.

For a species with 100% cover (the probability is one in the point mass at zero), the CDF curve would be a horizontal line where the y-axis equals one. The area underneath the CDF would also equal one in that case. In general, the area under the curve (but excluding the point mass probability at one) is a value between zero and one and is a good summary measure (point estimate) for a given species because it integrates information on both cover and relative frequency at a range of $r$ values.

It can be shown that the area underneath the CDF equals one minus the mean of $\text{beinf}(y; p_0, p_1, \mu, \phi)$ (see Appendix S1 for the derivation). Because cover and relative frequency are occurrence probabilities depicted by the CDF curve, and because we measure the area under the curve, we propose to call our new summary measure *integral occurrence probability*. The IOP can be calculated directly from the parameters of the distribution (Ospina & Ferrari 2010) as one minus the mean, $E(y)$:

$$1 - E(y) = 1 - (p_1 + (1 - p_0 - p_1)\mu)$$
An estimate of the variance for this point estimate is:

$$\text{Var}(y) = \frac{p_0p_1}{p_0 + p_1} + (1 - p_0 - p_1) \frac{\mu(1 - \mu)}{\phi + 1} + (p_0 + p_1)(1 - p_0 - p_1)(\frac{p_1}{p_0 + p_1} - \mu)^2$$

This variance should, however, not be used to construct confidence intervals around the mean as would be done for normally distributed data. Instead, a parametric bootstrapping procedure is recommended.

The IOP can also be calculated directly from the arithmetic sample mean:

$$1 - E(y) = E(1 - y) = \frac{\sum_{i=1}^{n}(1 - \frac{r^2}{r_{\text{max}}^2})}{n}$$

which shows that the summary measure equals the mean relative annulus area. Similarly, the sample variance can be calculated, but should again not be used to construct confidence intervals. Instead, a non-parametric bootstrapping procedure can be applied.

Thus, to estimate IOP and to directly predict IOP in regression modelling, it is best to transform the bounded point-to-plant distances to relative annulus areas $(1 - y)$. Then, a point-to-plant distance of zero, has transformed value 1, and, absence of a species, is zero after transformation. Because of the symmetric role of the parameters in $y$, the mapping of the parameters from $y$ to $1 - y$ is as follows:

$$\mu_y = 1 - \mu_{1-y}, \phi_y = \phi_{1-y}, p_{0,y} = p_{0,1-y} \text{ and } p_{1,y} = p_{0,1-y}.$$ For the CDF, the following holds:

$$CDF_y = 1 - CDF_{1-y}.$$ 

**Simulation strategy and calculation of point-to-plant distances**

We simulated species point patterns with varying point intensity (50, 100 and 500 points per unit area) and type of spatial distribution in the unit square (random, regular and clustered). Each combination was replicated five times (45 point patterns). Next, we transformed each point in a given pattern into a disc of fixed size (radius equal to 0.005, 0.015 or 0.03 units), resulting in a total of 135 representations of the spatial distribution of species cover in the unit square. Three examples are given in Figure 2.
Figure 2: Examples of simulations based on point patterns with varying point pattern type, intensity and disc radius. The point locations and maximum search distance around each of the points on an 8 x 8 grid are indicated in grey.

We overlaid the spatial distribution patterns with a systematic grid sample of 8 x 8 points in the unit square. To avoid edge effects, we fixed the starting position of the grid at \((x = 1/16, y = 1/16)\). These 64 points constituted the sampling locations from which we measured the point-to-plant distances. The maximum search distance was set at \(r_{\text{max}} = 0.04\) units (again avoiding edge effects as \(r_{\text{max}} < 1/16\)).

The R package `spatstat` was used for the simulation experiment (Baddeley et al. 2015). Technical details and R-code to reproduce the spatial distributions and calculation of point-to-plant distances are given in Appendix S2. Extra figures that display what the patterns look like are given in Appendix S3 (Figs. S1-S3).

**Field trial**

The method was tested in practice in Nardetea grassland vegetation. The grasslands were situated in a part of the Natural Reserve Klein Schietveld (Latitude: 51.339342°, Longitude: 4.502535°). We delimited three parcels of 50 m by 100 m in which we sampled 20 sampling points per parcel on a grid (spacing approximately 15 m by 20 m). At each sampling point, a metal rod with diameter 0.005 m was lowered perpendicularly into the vegetation at a fixed distance and direction from a temporary marker. Each species that touched the rod was noted a distance of zero. Next, we searched in outward direction for other species and recorded the distance with a folding rule to their closest aboveground part. The maximum search distance was set at 0.5 m. Measurements were recorded into a spreadsheet on a smartphone, which also logged a timestamp for each record. Each parcel was also surveyed to have a near complete species list.

The species recorded in the field trial were categorised into forbs, graminoids and Ericaceae. The point-to-plant distances were aggregated across species that belonged to the same species group by calculation of the minimum among their point-to-plant distances at each sampling point.
We used the `gamlss` package in R to fit the zero-and-one inflated beta distribution to the distance data after transformation to relative annulus areas \((1 - y)\). Generalized Additive Models for Location, Scale and Shape (GAMLSS) is a general framework for fitting regression type models (Rigby & Stasinopoulos 2005). It can be seen as an extension of the generalized linear modelling and generalized additive modelling framework from the class of exponential distributions to any class of distribution functions. In our application of the framework, the `gamlss` package simply fits a GLM by maximum likelihood estimation.

For the simulation experiment, separate intercept-only models were fitted for each of the 135 simulated species. The mean and variance of the relative annulus area and estimates for cover and relative frequency at \(r_{\text{max}}\) were calculated from the fitted parameters. We also calculated the arithmetic mean of \(1 - y\) (the IOP calculated directly from the data) and its variance to see if the modelled mean-variance relationship of the zero-and-one inflated beta distribution was suitable for the data.

Because cover, IOP and relative frequency at \(r_{\text{max}}\) are again on a 0 to 1 scale, `gamlss` models assuming a zero-and-one inflated beta distribution for the responses were fit with intensity (three levels), distribution type (uniform, clustered, regular), disc radius (three levels) and all two-way and the three-way interaction as explanatory variables.

For each simulated species, the CDF curve was calculated. We summed the CDF curves for the five replicates in each treatment group to obtain expected species richness as a function of relative plot area (expected species richness = the sum of occurrence probabilities over species).

For the illustrative field trial, we simplified the design and ignored that the data were from three parcels. Instead of fitting zero-and-one inflated beta distributions, we simply calculated for each species and species group the arithmetic mean of point-to-plant distances transformed to relative annulus areas (i.e. the IOP), the arithmetic mean of point-to-plant distances transformed to presence-absence data at \(r = 0\) (i.e. the cover) and the arithmetic mean of point-to-plant distances transformed to presence-absence data in a circle plot with \(r = r_{\text{max}} = 0.5\) m (i.e. relative frequency in largest circle plot). Non-parametric bootstrapping of the mean was used to construct 95% confidence intervals.

### Results

#### Mean-variance relation of IOP

Estimation of the mean relative annulus area (i.e. the IOP) and its variance based on the zero-and-one inflated beta yielded almost exactly the same values as direct estimation of the arithmetic mean and sample variance of \(1 - y\) (a regression through the origin had slope and \(R^2\) close to 1 in both cases).

Depending on the value of the precision parameter \(\phi\), we observed a reduction in variance compared to the expected variance from a Bernoulli distribution with the same mean. This reduction was stronger when the size of discs was smaller (Figure 3).
Figure 3: Observed mean - variance relationship (points) compared to expected variance of a Bernoulli random variable (line). Each circle represents one of the 135 simulated species. The size of the circle is proportional to the size of individual plants or clumps of plants (disc radius).

Sensitivity of cover, IOP and relative frequency to simulation settings

Figure 4 displays the effects of the factors (type, intensity, disc radius) that determined the spatial distribution patterns for three types of response variables (cover, relative frequency and IOP) that can be derived from point-to-plant distance data. It illustrates that the IOP still discriminates between situations, where either cover or relative frequency alone do not.

When cover is low due to small size of the individual plants, both the IOP and relative frequency in circle plots with fixed size will still be able to detect differences between species due to differences in point pattern intensity. On the other hand, the same relative frequency can hide differences that are recovered by the IOP and cover values. For instance, consider randomly distributed species with highest point pattern intensity (500). They all have relative frequency in the largest circle between 0.95 and 1.00, while cover and the IOP range from 0.03 to 0.84 and from 0.68 to 0.99, respectively. Given a point pattern intensity, clustered distributions will tend to have lower values compared to regular or random patterns.
Figure 4: Predicted values (+/− 95% confidence intervals) for three types of summary measures that can be calculated from point-to-plant distances. Each label along the x-axis indicates, respectively, the type of spatial point pattern, the point pattern intensity and the disc radius. The labels are sorted following the predicted mean value of the integral occurrence probability. For each measure, an inflated beta-regression model was fitted with all main effects and two- and three-way interactions.

Scaling of relative frequency and expected species richness with relative area

The CDF curves for the simulated species ranged from an almost flat line, to steep lines that start at 0 and level off at 1 (Figure 5, see also Appendix S5 for all simulated species). The relative frequency at $r_{\text{max}}$ was between 0.06 and 1. The estimated cover (at $r = 0$) ranged from 0 to 0.84. Relative frequency tended to increase at a slower rate when the disc radius (size of a plant) was largest and for clustered patterns. The regular patterns leveled off quicker. The completely spatial random patterns held an intermediate position between clustered and regular. These trade-offs implied that the same IOP could arise from different combinations of intensity, disc size and type of spatial patterning.

Summed CDF-curves showed how expected species richness increased with (relative) circular plot area (Figure 5). The sum was taken over the CDF-curves from replicated species within the same experimental group, but could be calculated for all species or any species group. Additional results are in Appendix S6 and show that the asymptote (i.e. equal to 5, because of 5 replications per factor combination) was reached earlier for simulated species that were regularly distributed, had higher point pattern intensity and larger (disc) size.
Figure 5: Examples of individual (left) and summed (right) CDF-curves for simulated species that belong to contrasting combinations of point pattern type, intensity and size (radius of disc).

Field trial

A total of 62 species were recorded within 0.5 m from the 60 sampling points. An additional 20 species were found when searching the 1.5 hectares in which the 60 sampling points were situated. If only the species that touched the rod were recorded (as in point-intercept sampling), only 22 species would have been recorded. Based on detailed timings in one parcel, recording and measuring took on average 5.4 minutes per sampling point (range: 1 - 16 minutes) when one person measured and the other took care of data input.

Figure 6 shows that graminoids clearly dominate in terms of cover (≈ 90%), with individual graminoid species reaching cover values up to 50%. Forbs and Ericaceae as a group have much lower cover (≈ 12%). Individual forb species always have low cover, yet can reach fairly high relative frequencies and this is also reflected by the IOP. In terms of relative frequency in the largest circle, there is little difference between forbs and graminoids as a group. The IOP, however, still discriminates between these groups, because it integrates both cover and relative frequency aspects. The overall pattern found with the field data resembles in many ways the patterns found with the simulated species (see Figure 4).
Figure 6: Estimates for cover, integral occurrence probability and relative frequency in a circle with radius 0.5 m for species (grey) and species groups (black). The 95% confidence intervals are calculated by non-parametric bootstrapping. Species are sorted following increasing integral occurrence probability. The light grey circles in the background represent the raw point-to-plant distance data transformed to relative annulus areas. Their size is proportional to the number of cases with the same relative annulus area. Only species with at least three point-to-plant distances smaller than 0.5 m are shown.

Discussion

General findings and applications

Simple measurements of point-to-plant distances allowed us to estimate cover as well as relative shoot frequency for any circular plot size up to the maximum search distance. A key insight was to properly transform the raw point-to-plant distances so that they reflect the relative area searched (or its complement). We have shown that cover and relative frequency are conceptually and mathematically related. Furthermore, point-to-plant distances allow the estimation of an integral occurrence probability (analogous to the calculation of the mean relative annulus area), which is a new measure that combines both attributes of plant abundance.

When cover is low, pin-point sampling does not differentiate between species with widely differing density and distribution patterns. On the other hand, determining relative frequency in a set of circular plots of a certain radius will not differentiate between species with widely differing cover values. The IOP is better able to differentiate between both...
extremes (Figure 4 and Figure 6).

For instance, Luzula campestris and Agrostis were about equally frequent within a distance of 0.5 m from the sampling points (respectively, 37 and 38 presences, 23 and 22 absences). On the other hand, Luzula campestris had much lower cover: the rod hit the species only 6 out of 60 times, whereas 22 out of 60 hits were recorded for Agrostis. Because the IOP combines cover and relative frequency at all distances smaller than the maximum search distance, a slightly lower IOP for Luzula campestris was found. The average distance was 0.09 m for Luzula campestris and 0.04 m for Agrostis when present. Transforming the distance values to relative annulus areas gives relatively more weight to short distances (they are compressed close to 1). Thus if the species are found in close vicinity of the sampling point, the difference between IOP and relative frequency in the largest circle will likely be small. Even for species with the same cover and relative frequency in the largest circle, the IOP can still differ. Achillea millefolium and Rumex acetosella had the same cover (1 hit in 60 sampling points) and were about equally likely to be recorded within 0.5 m from the sampling point (respectively 19 and 21 times). Still, because Achillea millefolium was at an average point-to-plant distance of 0.18 m whereas Rumex acetosella was on average at 0.11 m, the IOP for Rumex acetosella was higher.

The relative annulus area can be viewed as a continuous value that serves as the equivalent of how Böcher (1935) and Morrison et al. (1995) recorded species occurrences in a series of nested plots of increasing size: the highest rank was given to occurrence in the inner-most plot and lowest rank to first occurrence in the outer-most. The rank value was made relative in Morrison et al. (1995) through division by the number of nested plots and called importance-score.

The idea of a measure that combines different aspects of plant abundance is not new. In forest communities, the Importance Value developed by Curtis & McIntosh (1951) is a sum of relative dominance, relative density and relative frequency for each tree species. Curtis & McIntosh (1951) write that the Importance Value is an excellent indication of the importance of a species within a stand, since it is sensitive to apparent contagion or exceptional basal area.

Inference for a species group (Figure 6), is simply a matter of taking the minimum values among the point-to-plant distances of the species belonging to the species group at each sampling location. In this way, the estimated cover, relative frequency and IOP for the species group properly accounts for spatial overlap between the constituent species. This is in sharp contrast to the aggregation of visually estimated cover values by addition, which is unrelated to the cover of the species group as a whole when there is leaf overlap between species (Wilson 2011; see also Fischer 2015).

Our method may be very useful, for instance, in monitoring conservation or restoration management, where sensitive measures are needed that react when plants change in cover, relative frequency or both. For instance, Matthews & Whittaker (2015) review the usefulness of species abundance distributions in the context of biodiversity management. They, McGill et al. (2007) and Anderson et al. (2012) also discuss using measures other than abundance (sensu stricto i.e. counts, density), and we propose that our new measure may fit well in this scheme. Thus, simple exploratory data analysis techniques (e.g. histograms, empirical cumulative distributions or rank-abundance sensu lato plots based on the IOP)
may be useful to track changes over time, especially if the species pool can be deconstructed into indicative species groups.

Another application may be in ground-truthing of remote sensing data. In remote sensing, aggregating pixel data into sets of increasingly larger areal units of analysis results in different data values and inferences (Cressie 1996; Jelinski & Wu 1996; Wu & Li 2009; Feilhauer et al. 2014). At each resolution, a different ground-truth value should correspond. Our method, if focused on the species visible from a birds-eye perspective, may be able to deliver these scale-dependent ground-truth values.

**The zero-and-one inflated beta distribution and regression modelling**

If the purpose is solely to rank the species from low cover/shoot frequency to high cover/shoot frequency, there is no need to fit the distribution and it suffices to just calculate the arithmetic mean of \(1 - r^2/r_{\text{max}}^2\) (a result which we confirmed in the first simulation study and used in the analysis for the field trial). Non-parametric bootstrapping can then be used to construct confidence intervals around the mean. The fact that the modelled mean and variances matched the arithmetic mean and variances closely, is a testament to the flexibility of the zero-and-one inflated beta distribution. The mean-variance relationship showed that gains in statistical efficiency are possible when distances are measured instead of only recording presence-absence. The reduction in variance is due to the fact that the variance of the beta-distribution cannot exceed the variance of a Bernoulli variable with the same mean. If the plants are small (~ "disc size") compared to \(r_{\text{max}}\), the beta-part of the zero-and-one inflated distribution becomes more important, which explains what we observed. In practice, this means that \(r_{\text{max}}\) should be chosen larger than the average (vertically projected) size of plants (or clumps). This choice will also depend on species richness and search time. For grasslands, this could be in the range of 0.1 – 0.5 m, while for the herbaceous layer of forests 1 – 5 m may suffice in most situations.

The ability to predict relative frequency at a range of circular plot sizes from the CDF, without actually having to lay out a series of circular plots of increasing size, is appealing because it allows comparisons with data from studies or areas where a fixed plot size was used to record presence or absence - provided that the radius of the fixed size circular plot does not exceed the maximum search distance.

The scaling relationships also tell us something about the spatial distribution pattern of the species (see also Baddeley et al. (2015) Fig. 8.8): steep curves that reach an asymptote early are indicative of regular distribution patterns, shallow curves indicate clustered patterns.

The assumption that the data follow an inflated beta distribution is certainly necessary when we want to model the IOP as a function of continuous covariables or (random) factors (regression modelling) that explain the heterogeneity in the vegetation that we sample. With this distribution, we specify the correct mean-variance relationship and this is considered one of the most important steps when choosing a suitable model for the data (Warton et al. 2012; Warton et al. 2014).

An interesting avenue for future research is the joint modelling of multiple species (Warton et al. 2015). The ZOIB package (Liu & Kong 2015) implements separate as well as joint
modelling of multiple zero-and-one inflated beta distributed responses in a Bayesian model framework. For joint modelling, it implements up to three correlated responses with an unstructured covariance matrix, which can be enough if species can be grouped sensibly (Damgaard 2015a).

The expected number of species curve

The individual lines depicting the relative frequency as a function of relative radius can be combined by summing them. Since relative frequency is an occurrence probability, the sum of these values is an estimate of the expected number of species within a plot with radius $r$ (alpha diversity) (Ovaskainen & Hanski 2003). At $r = 0$, this is also the expected number of species (not total cover over species), or, put differently, the expected (= average) number of species that will be touched by a pin when it is lowered at a random location into the vegetation. Our approach to derive the expected number of species does not depend on an assumed spatial distribution pattern, as opposed to formulae in Blackman (1935).

The resulting monotonically increasing curve shows how species richness scales with area up to the size of a circle with radius $r_{max}$. These curves are similar in nature to the curves obtained when a series of (nested) differently sized plots would have been used (as opposed to species accumulation curves obtained from one plot size only). Hence, they relate to the Braun-Blanquet minimal area concept (Braun-Blanquet 1964) where the size of quadrats for classical vegetation relevés is chosen at the point where the species area curve levels off. This concept forms the basis for the variation in plot sizes that are in common use in phytosociology (Chytrý & Otýpková 2003). As a consequence, the scaling of expected species richness with area may be used to optimize the maximum search distance in future surveys. The practice of choosing a plot size on the premise that a single plot must be representative for a homogeneous vegetation community, as in the Braun-Blanquet minimal area concept, is questionable. With point-to-plant distances, replication is obviously necessary to describe a homogeneous area of vegetation adequately. Moreover, we feel that criteria other than species richness may be more important in practice. Especially the search time and the increasing difficulty of finding the shortest distance when we move further and further away from the center point is a limiting factor. Setting the maximum search radius lower means that sample size can be increased, given a time budget constraint.

Practical issues and sample size considerations

We introduced our method as a continuous version of the methods discussed in Morrison et al. (1995) and Damgaard (2015b). In practice, our results indicated that there is no need to establish a series of concentrically placed plots of increasing size if point-to-plant distances are measured. In fact, it can be shown that the estimates of relative frequency in the smallest circular plot obtained by the method of Damgaard (2015b) are quasi identical to the estimate of relative frequency at the corresponding radius with the methods explained in this paper (see Appendix S7 for results). Morrison et al. (1995) compared frequency estimation with either a plot consisting of seven concentrically placed subplots or nine equal area subplots. They found that the former was much less time-consuming and at the same time gave functionally the same results. The
same is likely to hold for point-to-plant distance measurements and our field trial also indicated that the method is relatively fast.

Establishing whether a species is present at the point (distance = zero) suffers from the same difficulties as pin-point sampling. In theory, the "pin" should be as small as possible (a needle). However, we feel that in practice the cross-sectional area of the pin may be larger as long as it is negligible compared to the plant (we used a rod with diameter 0.005 m). As can be seen from the scaling of relative frequency with relative radius, this will result in a small overestimation of cover. Moreover, with pin-point sampling the effect of cases of doubt (is it a hit, record 1, or not, record 0?) is more severe compared to point-to-plant distance measurements (recording a distance of zero or a very short distance will have little effect on the mean relative annulus area).

Point-to-plant distances are also subject to measurement error due to misidentification, overlooking individuals of a plant, ... as is the case for any method for measuring plant abundance. Preliminary results of the field trial (not shown) indicated good reproducibility of the method conditional a species is detected. Still, in even more complex (much layering) vegetation types, these measurements may be prohibitively difficult. The problem in such situations is that the search time will be large (a problem that can be partly overcome by restricting the maximum search radius) and the vegetation will be disturbed during the search (increasing measurement error). However, our method can easily be modified by restricting the search along a line-segment (which can have random or fixed orientation). In this case, the data would need to be transformed differently to reflect this: \[ y = \frac{r}{r_{\text{max}}} \].

After this transformation, the analysis can proceed as before. The interpretation of relative frequency at a distance \( r \) now changes to the occurrence probability on a line segment of length \( r \). However, in Appendix S4, we show that, on average for the simulated species, a 4.8 times larger sample size is needed to obtain the same relative margin of error as with the circular approach. This implies that only in exceptional circumstances the benefits of the line-transect approach (less damaging and possibly easier) will compensate for the loss of information. Furthermore, the estimates for the IOP obtained with line-segments are much closer to the cover estimates. This makes sense because we move from an area-based approach over a line-based approach to a point-based approach. A pure point-based approach (point-intercept sampling) would need, on average, an 8.5 times larger sample size compared to the circular approach.

Connections to density estimation?

In the past, connections have been drawn between density estimates and relative frequency in terms of rooted frequency (e.g. Blackman 1935; Aberdeen 1958; Morrison et al. 1995). Blackman (1935) shows how density can be approximated from an estimate of absence frequency for randomly distributed species. Aberdeen (1958), claims that density can be estimated from absence frequency measurements if combined with an estimate of the average size of the plant unit.

The estimation procedures outlined in this paper cannot be used to estimate a density value, but it is clear that the integral occurrence probability is related to density (Figure 4, see also Morrison et al. (1995)). However, the measurements themselves should, in
principle, allow for the calculation of density estimates for species where individuals can be
easily distinguished (as opposed to clumps) and which are very small relative to the area
searched. The distance measurements will then approximate the position where plants are
rooted (the shortest distance to the shoot will be almost the same as the distance to where
the plant is rooted). When this is the case, the measurements coincide with the proposal of
Batcheler & Bell (1970) and Batcheler (1971) who derives a density estimator that is
applicable to point-to-plant measurements up to a maximum search distance. However, the
density estimator is biased when the distribution is non-random and additional distance
measurements from the plant to the second and third nearest neighbour are needed to
correct for this bias (Batcheler 1971). In fact, density estimators for plotless sampling
techniques often assume complete spatial randomness (Engeman et al. 1994). An
interesting estimator that relies on point-to-plant distances was proposed by Barabesi &
Marcheselli (2002) in a design-based context, thus avoiding assumptions about the spatial
distribution. However, they do not provide a solution to the case where there is a maximum
search distance. Interestingly, Barabesi & Marcheselli (2002) used a transformation that
results in relative squared distances, but it is made relative to the inclusion area of each
sampling point. Their estimator is a kernel density estimate of the transformed distances
near zero. Further research is necessary to see if their approach can be applied to the case
of a maximum search distance - possibly the kernel density estimator proposed by Geenens
(2014) may be useful in this context.

Acknowledgements

We thank Els De Bie, Els Lommelen and Patrik Oosterlynck for assistance with the field
work and Dirk Leyssens of Agentschap voor Natuur en Bos for the field work permission.
Charlotte Elisabeth Kler kindly helped us with the English language revision. We thank Dr.
János Podani, Dr. Norm Kenkel and one anonymous reviewer for the comments and
suggestions that substantially improved our manuscript. We acknowledge the support of
ALTER-Net in the form of a travel grant.

Supporting information

Additional supporting information may be found in the online version of this article:

Appendix S1: R-code to illustrate fitting of a zero-and-one inflated beta distribution to
transformed point-to-plant distance data

Appendix S2: Technical details and R-code for simulated plant distribution patterns and
calculation of point-to-plant distances

Appendix S3: Additional figures with examples of simulated plant distribution patterns

Appendix S4: Fitting the zero-and-one inflated beta distribution to data from the
simulation experiment

Appendix S5: Occurrence probability as a function of relative squared radius (cumulative
curves) for data from the simulation experiment
Appendix S6: Additional results for expected species richness for the simulation experiment

Appendix S7: Comparison with the Böcher-modified Raunkiaer method

References


Liu, F., & Kong, Y. 2015. ZOIB: an R Package for Bayesian Inferences in Beta and Zero One Inflated Beta Regression Models. *The R Journal* 7:


Appendix to Integral occurrence probability: combining cover and relative shoot frequencies based on bounded point-to-plant distances

Hans Van Calster, Christian Damgaard

Van Calster, H. (corresponding author, hans.vancalster@inbo.be[^1])

Damgaard, C. ([cfd@dmu.dk[^2]])

[^1] Biometry and Quality Assurance, Research Institute for Nature and Forest (INBO), Kliniekstraat 25, 1070 Brussel, BE;

[^2] Department of Bioscience, Aarhus University, Vejlsøvej 25, 8600 Silkeborg, DNK

Appendix S1: R-code to illustrate fitting of a zero-and-one inflated beta distribution to transformed point-to-plant distance data

```r
# R packages
# use install.packages() if some or all of these packages cannot be loaded
library(spatstat)
library(reshape2)
library(plyr)
library(ggplot2)
library(gridExtra)
library(gamlss)
library(gamlss.dist)
library(bbmle)

theme_set(theme_bw(10))
```

The following output prints information about the R session in the paper and supporting information:

```r
sessionInfo()
```

```
## R version 3.3.1 (2016-06-21)
## Platform: i386-w64-mingw32/i386 (32-bit)
## Running under: Windows 7 x64 (build 7601) Service Pack 1
##
## locale:
## [1] LC_COLLATE=Dutch_Belgium.1252 LC_CTYPE=Dutch_Belgium.1252
## [3] LC_MONETARY=Dutch_Belgium.1252 LC_NUMERIC=C
## [5] LC_TIME=Dutch_Belgium.1252
##
```
Calculation of mean and variance from parameters

Before we show how to fit the distribution, we briefly explain the parametrisation used in `gamlss.dist::BEINF` and give formula for the calculation of the first two moments (mean and variance) of the distribution.

The parametrisation in `gamlss` is $$\mu = \mu, \sigma = \sqrt{1/(\phi + 1)}, \nu = p_0/p_2 \text{ and } \tau = p_1/p_2 \text{ where } p_2 = 1 - p_0 - p_1.$$  

Most textbooks will give the beta-distribution in terms of two shape parameters $$a$$ and $$b$$. The relationship with $$\mu$$ and $$\phi$$ is as follows: $$\mu = \frac{a}{a+b}$$ and $$\phi = a + b$$.

The following chunk with r-code explains the various parametrisations and how to calculate the mean and variance from these parameters.

```r
# a set of chosen values for the zero-one-inflated beta distribution
# parametrisation see ?gamlss.dist::BEINF
mu <- 0.2  # mean of the beta part
sigma <- 0.3  # dispersion parameter for the beta-part
tau <- 0.1
nu <- 0.5

# beta part of the distribution:
# ...............................................................
# in terms of the standard shape parameters for beta(a,b)
```
a <- mu * (1 - sigma ^ 2)/(sigma ^ 2)
b <- a * (1 - mu)/mu

# the mean of the beta part
checkmeanbeta <- a/(a + b)  # equal to parameter mu
all.equal(mu, checkmeanbeta)

## [1] TRUE

# dispersion or precision parameter
phi <- a + b  # precision parameter
checksigma <- sqrt(1/(a + b + 1))
all.equal(checksigma, sigma)

## [1] TRUE

# the variance of the beta part
varbeta <- mu * (1 - mu) * sigma ^ 2
# variance of beta distribution
checkvarbeta <- a * b / (((a + b) ^ 2)*(a + b + 1))
all.equal(varbeta, checkvarbeta)

## [1] TRUE

# Bernoulli part of the distribution

# the proportion zero and ones
p10 <- (tau + nu)/(1 + tau + nu)  # the proportion zero and ones
p1 <- tau * (1 - p10)  # the proportion ones
p0 <- nu * (1 - p10)  # the proportion zeroes
all.equal(p10, p1 + p0)

## [1] TRUE

# Mean and variance of the zero-and-one inflated beta distribution

# mean
# overall mean of the zero-and-one inflated beta
meanbeinf <- p1 + (1 - p0 - p1)*mu
# in terms of gamlss parametrization
checkmeanbeinf <- (tau + mu) / (1 + nu + tau)
all.equal(meanbeinf, checkmeanbeinf)

## [1] TRUE
# the variance:

\[ \text{varbeinf} \leftarrow \frac{p_0}{p_0 + p_1} + \frac{1 - p_0 - p_1}{(p_0 + p_1)} \cdot \mu \cdot \left(1 - \mu\right)^2 + \left(\frac{p_0 + p_1}{p_0 + p_1} - \mu\right)^2 \]

We also demonstrate the relationship of the mean with the area under the cumulative distribution function. Approximating the area underneath the curve is cumbersome, and, luckily, not necessary because it can be calculated through a general relationship between the mean of a distribution and its CDF (Hajek 2015). For a given CDF, the mean is equal to the area of the region above the CDF curve and below the horizontal line where the y-axis equals one for positive values along the x-axis minus the area of the region below the CDF curve (and above the x-axis) for negative values along the x-axis. Since, x-axis values are between 0 and 1 in our case, the mean is equal to the area of the region above the CDF curve and below the horizontal line where the y-axis equals one. This is why in the paper, the integral occurrence probability is calculated as one minus the mean of \( \text{beinf}(y, p_0, p_1, \mu, \phi) \). Note that different CDF curves can have the same mean.

The following chunk with R-code, demonstrates by example the equivalence of the area underneath the CDF curve and one minus the mean of the zero-and-one inflated beta distribution.

```r
# relationship of the mean with the area under the curve of the cumulative
# distribution = "integral occurrence probability" if the point-to-plant
# distances, r, are transformed as \( r^2/r_{\text{max}}^2 \).
# the function pBEINF() is the cumulative distribution function
# we can calculate the area under the curve using the integrate() function
# note that we do not include the density associated with the point mass at one
auc <- integrate(pBEINF, lower = 0, upper = 1 - .Machine$double.eps,
                  mu = mu, sigma = sigma, nu = nu, tau = tau, lower.tail = TRUE)

# the same value can be calculated from the complement of the mean
# of the zero-one-inflated beta
all.equal(1 - meanbeinf, auc$value)
```

## Non-parametric bootstrap function for confidence bounds around CDF

The function boot.pBEINF can be used to calculate lower and upper bounds for the cumulative distribution function. The default setting produces 95% confidence intervals (alpha = 0.05).

```r
# y = transformed point-to-plant distances
# if y = \( r^2/r_{\text{max}}^2 \), use lower.tail = TRUE
# if y = 1 - \( r^2/r_{\text{max}}^2 \), use lower.tail = FALSE
# Nboot = number of bootstrap samples
# xvec = vector of values at which to evaluate pBEINF()
```
Fitting a zero-and-one inflated beta distribution with `gamlss`

To obtain the same results as in the main text

```r
set.seed(19)

samplesize <- 100

# squared measured shortest distance at samplesize locations divided by
# squared max search distance
sqrelrad <- rBEINF(n = samplesize, mu = mu, sigma = sigma, nu = nu, tau = tau)

# re-estimate the parameters for the sample
# fit the distribution
# ~ 1 specifies that only an intercept is estimated
fit.sqrelrad <- gamlss(sqrelrad ~ 1, 
                       sigma.formula = ~1, 
                       nu.formula = ~1,
```
\[ \text{tau.formula} = \sim 1, \]

# BEINF specifies a zero-and-one inflated distribution
family = BEINF(),
trace = FALSE)

# extract the parameters and transform from link scale to scale of the response
mu.sqrelrad <- plogis(coef(fit.sqrelrad))  # mean of the beta part
# dispersion parameter of the beta part
sigma.sqrelrad <- plogis(coef(fit.sqrelrad, parameter = c("sigma")))
# proportion ones divided by proportion not zero or one
tau.sqrelrad <- exp(coef(fit.sqrelrad, parameter = c("tau")))
# proportion zeroes divided by proportion not zero or one
nu.sqrelrad <- exp(coef(fit.sqrelrad, parameter = c("nu")))

# bootstrap the 100 squared relative distance values
sample1 <- boot.pBEINF(y = sqrelrad, Nboot = 500)
sample1 <- as.data.frame(sample1)

# plug-in the estimated parameters to evaluate the probability
density or cumulative distribution
sample1$sample_pdf <- dBEINF(sample1$xvec,
mu = mu.sqrelrad,
sigma = sigma.sqrelrad,
u = nu.sqrelrad,
tau = tau.sqrelrad,
log = FALSE)

sample1$sample_cdf <- pBEINF(sample1$xvec,
mu = mu.sqrelrad,
sigma = sigma.sqrelrad,
nu = nu.sqrelrad,
tau = tau.sqrelrad,
lower.tail = TRUE)

# for comparison of the sample with the population,
calculate the true density and cumulative distribution
sample1$pop_pdf <- dBEINF(x = sample1$xvec,
mu = mu,
sigma = sigma,
nu = nu,
tau = tau,
log = FALSE)

sample1$pop_cdf <- pBEINF(q = sample1$xvec,
mu = mu,
sigma = sigma,
nu = nu,
tau = tau,
lower.tail = TRUE)
plot1 <- ggplot(sample1, aes(x = xvec)) + 
  geom_line(data = sample1[sample1$xvec > 0 & sample1$xvec < 1, ],
            aes(y = pop_pdf)) + 
  geom_point(data = sample1[sample1$xvec == 0 | sample1$xvec == 1, ],
             aes(y = pop_pdf, shape = 21, size = 3)) + 
  geom_segment(data = sample1[sample1$xvec == 0 | sample1$xvec == 1, ],
                aes(x = xvec, xend = xvec, y = c(0, 0), yend = pop_pdf)) + 
  geom_line(data = sample1[sample1$xvec > 0 & sample1$xvec < 1, ],
             aes(y = sample_pdf, linetype = 2)) + 
  geom_point(data = sample1[sample1$xvec == 0 | sample1$xvec == 1, ],
             aes(y = sample_pdf, shape = 22, size = 3)) + 
  geom_segment(data = sample1[sample1$xvec == 0 | sample1$xvec == 1, ],
                aes(x = xvec, xend = xvec, y = c(0, 0),
                     yend = sample_pdf, linetype = 2)) + 
  xlab(expression(r^2/r[\text{max}]^2)) + 
  ylab("Probability density") + 
  ggtitle("Probability density function")

plot2 <- ggplot(sample1, aes(x = xvec, y = pop_cdf)) + 
  geom_line() + 
  geom_line(aes(y = sample_cdf, linetype = 2)) + 
  geom_ribbon(aes(ymax = upper, ymin = lower, alpha = 0.2)) + 
  xlab(expression(r^2/r[\text{max}]^2)) + 
  ylab("Relative (shoot) frequency \n= occurrence probability") + 
  ggtitle("Cumulative distribution function")

ggrid.arrange(plot1, plot2, nrow = 1)
Figure S1.1: Example of a PDF and CDF for a zero-and-one inflated beta distribution with $\mu = 0.2$, $\phi = 10.11$, $p_0 = 0.31$ (point mass probability at $0 = \text{cover}$) and $p_1 = 0.063$ (point mass probability at $1 = \text{absence probability, i.e. the probability that the species is not observed at } r < r_{\text{max}}$) (full lines). The dashed lines show the curves for the estimation with gamlss() for a sample of size 100. The full lines give the true curves from the model that generated the data. For the cumulative distribution function, 95% confidence bounds depict the uncertainty associated with the sample estimate.

Estimated coefficients, their standard errors, Information Criteria and other summary information can be viewed with the summary() function:

```r
summary(fit.sqrelrad)
```

```r

# ** Summary table for fitted model:
# Family:  c("BEINF", "Beta Inflated")
# Call:  gamlss(formula = sqrelrad ~ 1, sigma.formula = ~1, nu.formula = ~1,
# tau.formula = ~1, family = BEINF(), trace = FALSE)
# Fitting method: RS()
# Mu link function:  logit
# Mu Coefficients:
# Estimate Std. Error t value Pr(>|t|)
# (Intercept)  -1.2474  0.0848  -14.7  <2e-16 ***
# ---
```

The probability density function (PDF) and cumulative distribution function (CDF) are shown in the figure. The full lines represent the true curves from the model, while the dashed lines indicate the estimated curves for a sample of size 100. The 95% confidence bounds for the cumulative distribution function illustrate the uncertainty in the sample estimate.
Residuals and fitted values can be extracted from the model for model validation purposes. See ?residuals.gamlss() for details.

```r
fit.sqrelrad.validation <-
data.frame(y = sqrelrad,
E.zscore = residuals(fit.sqrelrad),
E.mu = residuals(fit.sqrelrad, what = "mu"),
E.sigma = residuals(fit.sqrelrad, what = "sigma"),
E.nu = residuals(fit.sqrelrad, what = "nu"),
E.tau = residuals(fit.sqrelrad, what = "tau"),
mu = predict(fit.sqrelrad, what = "mu", type = "link", se.fit = FALSE),
sigma = predict(fit.sqrelrad, what = "sigma", type = "link"),
```
Examples of model validation plots:

Example 1

```r
ggplot(fit.sqrelrad.validation, aes(x = y, y = E.zscore)) + geom_point()
```

![Figure S1.2: Model validation: observed values versus normalized and randomized quantile residuals.](image)

Example 2

```r
ggplot(fit.sqrelrad.validation, aes(x = E.zscore)) + geom_density()
```
Figure S1.3: Model validation: density plot of normalized and randomized quantile residuals.

Example 3

```r
ggplot(fit.sqrelrad.validation, aes(x = E.mu)) + geom_density()
```
Figure S1.4: Model validation: density plot of simple residuals for mu-parameter.

Example 4

```r
ggplot(fit.sqrelrad.validation, aes(x = E.sigma)) + geom_density()
```
Figure S1.5: Model validation: density plot of simple residuals for sigma-parameter.

Example 5

```r
ggplot(fit.sqrelrad.validation, aes(x = E.nu)) + geom_histogram()
```
Figure S1.6: Model validation: histogram of simple residuals for nu-parameter.

Example 6

```r
ggplot(fit.sqrelrad.validation, aes(x = E.tau)) + geom_histogram()
```
Next, we can predict the values of the estimated parameters or derived quantities (i.e. the overall mean response) for the same data or for new data. In the example, we fit only an intercept. Prediction becomes more interesting when covariates are included in the model.

```R
# calculate predictions (fitted values) in the link scale
# convert to response scale and calculate upper and lower bounds of
```

Figure S1.7: Model validation: histogram of simple residuals for tau-parameter.
# confidence intervals

gamlss.response <- adply(gamlss.link, .margins = 1, transform,
    mu.lwr = plogis(mu - qnorm(1 - 0.05/2) * mu.se),
    mu.upr = plogis(mu + qnorm(1 - 0.05/2) * mu.se),
    sigma.lwr = plogis(sigma - qnorm(1 - 0.05/2) * sigma.se),
    sigma.upr = plogis(sigma + qnorm(1 - 0.05/2) * sigma.se),
    tau.lwr = exp(tau - qnorm(1 - 0.05/2) * tau.se),
    tau.upr = exp(tau + qnorm(1 - 0.05/2) * tau.se),
    nu.lwr = exp(nu - qnorm(1 - 0.05/2) * nu.se),
    nu.upr = exp(nu + qnorm(1 - 0.05/2) * nu.se),
    mu = plogis(mu),
    sigma = plogis(sigma),
    tau = exp(tau),
    nu = exp(nu))

# predict the mean of the zero-and-one inflated beta,
# which is a quantity derived from the estimated parameters

gamlss.response$meanbeinf <- with(gamlss.response, (mu + tau)/(1 + nu + tau))

# add sample size

gamlss.response$nreps <- samplesize

We cannot use the upper and lower bounds on the estimated parameters to construct confidence intervals on a derived quantity. Again, bootstrapping is needed. The following function implements a parametric bootstrap which gives confidence intervals for the mean of the zero-and-one inflated beta.

# parametric bootstrap
# use install.packages("magrittr") if this line does not work
`%>%` <- magrittr::%>%

gamlss.response <- gamlss.response %>%
    # select first row (we only fitted intercept)
    # predictions are the same for each row
dplyr::slice(1) %>%
dplyr::mutate(`(Intercept)` = "intercept") %>%
dplyr::group_by(`(Intercept)`) %>%
dplyr::do(broom::tidy(replicate(50,
    mean(gamlss.dist::rBEINF(n = .$nreps,
        mu = .$mu,
        sigma = .$sigma,
        tau = .$tau,
        nu = .$nu)),
    simplify = "array"))) %>%
dplyr::rename(meanbeinf = x) %>
Instead of the model-based approach (zero-and-one inflated beta fit), we can also calculate the arithmetic mean and a non-parametric bootstrap confidence interval. The results are quasi identical. The latter can only be used for simple applications (no covariates, no random effects, ...).

```r
dplyr::summarise(meanbeinf.lwr = quantile(meanbeinf, probs = 0.025),
                 meanbeinf.upr = quantile(meanbeinf, probs = 0.975)) %>%
dplyr::right_join(gamlss.response %>%
                  dplyr::slice(1) %>%
                  dplyr::mutate((Intercept) = "intercept"))
```

```r
# compare parametric with non-parametric approach
# function mean_cl_boot calculates the mean and a non-parametric
# 95% confidence interval
sqrelrad_df <- data.frame(y_values = sqrelrad)
nonpar_cl <- mean_cl_boot(sqrelrad_df$y_values) %>%
             dplyr::mutate(method = "arithmetic_mean")

par_cl <- gamlss.response %>%
         dplyr::select(meanbeinf, meanbeinf.lwr, meanbeinf.upr) %>%
         dplyr::mutate(method = "beta_inflated_mean") %>%
         dplyr::rename(y = meanbeinf, ymin = meanbeinf.lwr, ymax = meanbeinf.upr)

compare_cl <- dplyr::bind_rows(par_cl, nonpar_cl)

ggplot(compare_cl, aes(x = method)) +
  geom_pointrange(aes(y = y, ymin = ymin, ymax = ymax)) +
  scale_y_continuous("Mean and 95% confidence interval") +
  theme(axis.title.x = element_blank())
```
Figure S1.8: Comparison of model-based estimate of the mean response with 95% parametric bootstrap confidence intervals and arithmetic mean with 95% non-parametric confidence interval.

Appendix S2: Technical details and R-code for simulated plant distribution patterns and calculation of point-to-plant distances

The R package *spatstat* was used for the simulation experiments (Baddeley et al. 2015). We provide here more detail about the simulation experiments and the R-code to reproduce them as well as the code to calculate point-to-plant distances from a sample consisting of an 8 x 8 grid of sampling points.

Simulation experiment

In the simulation experiment, we simulated species point patterns with varying point intensity and degree of clustering/regularity in the unit square. Next, we transformed each point in a given pattern into a disc of fixed size. The resulting number of points and the degree of overlap between discs determines the true cover of the species in the unit square.

As a reference pattern, the homogeneous Poisson point process was simulated at three point intensities (50, 100 and 500). These point patterns represent complete spatial randomness. The intensity is the *expected* number of points per unit area. Thus, on average, the simulated number of points in the unit square will equal the intensity.
Clustered point patterns were generated with a Matern’s cluster process. This followed a two-step procedure. First, a uniform Poisson point process was generated, which are parent points. Second, each parent point was replaced by a random cluster of offspring points with the number of offspring points being Poisson distributed with mean equal to 10. To arrive at the same expected overall point intensity as the uniform Poisson point patterns, the intensity of parent points was set at one-tenth of the three point intensities (5, 10 and 50). Clusters were set to be circles with radius 0.1 units.

Regular point patterns were generated with a simple sequential inhibition point process. Each new point is generated uniformly in the window and independently of preceding points. If the new point lies closer than an inhibition distance from an existing point, then it is rejected and another random point is generated. The desired number of points was sampled from a Poisson distribution with mean equal to the intensity (50, 100 or 500). To ensure that these point patterns were equally regular at different intensities, the inhibition distance was dependend on the intensity, i.e. inhibition distance equalled the reciprocal of two times the square root of intensity.

Each type (uniform, clustered, regular) in combination with three levels of overall intensity (50, 100, 500) was replicated five times (45 point patterns). Each of these point patterns was then converted to polygonal objects. For a given point pattern, this meant that each point was converted to a disc with the same area. Each realised point pattern was used three times with different sizes of the discs (radius equal to 0.005, 0.015 or 0.03 units), resulting in a total of 135 representations of the spatial distribution of species cover in the unit square.

R-code for simulation of random, regular and clustered spatial point patterns

```
# to obtain the same results as in the main text
set.seed(18777)

# vector with intensity of point pattern
intens <- factor(c(50, 100, 500))

# vector of radius for the size of the discs
# multiplied by 1000 (makes data handling easier later on)
radvec <- factor(c(5, 15, 30))

# replication
reps <- 5
simvec <- factor(1:reps)

# simulate point patterns
# random distributions (homogeneous Poisson point patterns,
# also known as uniform point patterns)
uni <- list()
```
for (i in `levels`(intens)) {
    for (j in `levels`(radvec)) {
        uni[[i]][[j]] <- rpoispp(as.numeric(i), nsim = reps)
    }
}

# regular distributions
# we use rSSI() because it is better able to create regular
# distribution with high intensity compared to rMaternI()
# for rSSI the point pattern intensity is an upper bound
# we therefore sample the intensity from a poisson distribution
# this ensures comparability with rpoispp and rMatClust
# the inhibition distance should also dependent on intensity
# so that regularity is the same
reg <- `list`()
for (i in `levels`(intens)) {
    for (j in `levels`(radvec)) {
        for (k in `levels`(simvec)) {
            reg[[i]][[j]][[k]] <- rSSI(r = 1/(2*sqrt(as.numeric(i))),
                                         n = rpois(n = 1, lambda = as.numeric(i)))
        }
    }
}

# clustered distributions
clu <- `list`()
clusrad <- 0.1
mufactor <- 10
for (i in `levels`(intens)) {
    for (j in `levels`(radvec)) {
        clu[[i]][[j]] <- rMatClust(as.numeric(i)/mufactor, scale = clusrad, mu = mufactor, nsim = reps)
    }
}

# rename the lowest level of the nested list
# makes it easier later on to handle the data
for (i in `levels`(intens)) {
    for (j in `levels`(radvec)) {
        names(clu[[i]][[j]]) <- simvec
    }
}
for (i in `levels`(intens)) {
    for (j in `levels`(radvec)) {
        ...
R-code to turn point patterns in spatial regions composed of discs with given centres and radii

# turn point patterns in polygons by means of discs()
unip <- list()
clup <- list()
regp <- list()

# radvec was multiplied by 1000, so division by 1000
for (i in levels(intens)) {
  for (j in levels(radvec)) {
    for (k in levels(simvec))  {
      clup[[i]][[j]][[k]] <- discs(clu[[i]][[j]][[k]],
                                  radii = rep(as.numeric(j)/1000,
                                  clu[[i]][[j]][[k]]$n))
      regp[[i]][[j]][[k]] <- discs(reg[[i]][[j]][[k]],
                                  radii = rep(as.numeric(j)/1000,
                                  reg[[i]][[j]][[k]]$n))
      unip[[i]][[j]][[k]] <- discs(uni[[i]][[j]][[k]],
                                  radii = rep(as.numeric(j)/1000,
                                  uni[[i]][[j]][[k]]$n))
    }
  }
}

Calculation of point-to-plant distances

Grid of sampling points

# to avoid edge effects, systematic grid with fixed starting position at
# x = y = 0.0625
xvec <- seq(1/16, 1 - 1/16, 1/8)
syspoints <- expand.grid(x = xvec, y = xvec)
ranpos <- as.ppp(X = syspoints, W = square(1))

# define a maxdist beyond which we do not look for a species
maxdist <- 0.04 # should be smaller than 1/16 to avoid edge effects
ranposp <- discs(ranpos, radii = rep(maxdist, ranpos$n))
Calculate shortest distances from sampling points to plant distribution patterns up to a maximum search distance

Circular approach

Here we calculate the shortest distance when we search the area around the sampling point up to a maximum search distance.

distr <- vector("list", 3)
patterns <- factor(c("uni", "reg", "clu"))

names(distr) <- patterns
distr[["uni"]]<- unip
distr[["reg"]]<- regp
distr[["clu"]]<- clup

shortdist.sim1 <- vector("list", 3)

for (p in levels(patterns)) {
  for (i in levels(intens)) {
    for (j in levels(radvec)) {
      for (k in levels(simvec)) {
        # function to evaluate the shortest distance from a point (x,y)
        # to the spatial distribution pattern
        f <- distfun(distr[[p]][[i]][[j]][[k]])
        # evaluate the function at the 8 x 8 grid locations
        shortdist.sim1[[p]][[i]][[j]][[k]] <- f(ranpos$x, ranpos$y)
      }
    }
  }
}

# turn the list that holds the distances into a data.frame
shortdist.sim1 <- as.data.frame(shortdist.sim1)
shortdist.sim1$plot <- 1:ranpos$n # add plot (sampling point) IDs

shortdist.sim1 <- melt(shortdist.sim1,
  id.vars = "plot",
  value.name = "distance",
  variable.name = "species") # convert from wide to long format

# make a data.frame with variables encoding for each "treatment"
grid <- expand.grid(plot = 1:ranpos$n,
  pattern = patterns,
  intens = intens,
  radvec = radvec,
  simvec = simvec)

grid$species <- factor(paste(grid$pattern,
  grid$intens,)}"
grid$radvec,
grid$simvec,
sep = ".")

# add these variables to the data.frame with shortest distances
shortdist.sim1 <- merge(grid, shortdist.sim1)

# and set distance to maxdist if the species is further away than this distance
shortdist.sim1$distance <- ifelse(shortdist.sim1$distance >= maxdist,
                                   maxdist,
                                   shortdist.sim1$distance)

Distance along line transects

Here we show how shortest distances can be calculated when we search a line segment starting from the sampling points in northern direction up to a maximum search distance.

# create line segments
lineseg <- psp(x0 = ranpos$x,
y0 = ranpos$y,
x1 = ranpos$x,
y1 = ranpos$y + maxdist,
window = owin())

# calculate crossing points between boundaries of distribution patterns and line segments

crossings <- vector("list", 3)
names(crossings) <- patterns
for (p in levels(patterns)) {
  for (i in levels(intens)) {
    for (j in levels(radvec)) {
      for (k in levels(simvec)) {
        crossings[[p]][[i]][[j]][[k]] <-
        crossing.psp(edges(distr[[p]][[i]][[j]][[k]]), lineseg)
      }
    }
  }
}

# check if sampling points are inside polygons defined by distribution patterns
inside <- vector("list", 3)
names(inside) <- patterns
for (p in levels(patterns)) {
  for (i in levels(intens)) {
    for (j in levels(radvec)) {

for (k in levels(simvec)) {
  inside[[p]][[i]][[j]][[k]] <- inside.owin(x = ranpos, w =
  distr[[p]][[i]][[j]][[k]])
}
}
}

# calculate shortest distances between sampling points and crossings

disttocross <- vector("list", 3)
names(disttocross) <- patterns
for (p in levels(patterns)) {
  for (i in levels(intens)) {
    for (j in levels(radvec)) {
      for (k in levels(simvec)) {
        f <- distfun(crossings[[p]][[i]][[j]][[k]])
        # evaluate the function at the 8 x 8 grid
        disttocross[[p]][[i]][[j]][[k]] <- f(ranpos$x, ranpos$y)
      }
    }
  }
}

# if sampling point is inside, set distance to 0
# else, return the shortest distance (first crossing)

linedist <- inside
for (p in levels(patterns)) {
  for (i in levels(intens)) {
    for (j in levels(radvec)) {
      for (k in levels(simvec)) {
        linedist[[p]][[i]][[j]][[k]] <-
        ifelse(linedist[[p]][[i]][[j]][[k]] == TRUE,
               0,
               unlist(disttocross[[p]][[i]][[j]][[k]])))
      }
    }
  }
}

# turn the List that holds the distances into a data.frame
linedist <- as.data.frame(linedist)
linedist$plot <- 1:ranpos$n  # add plot (sampling point) IDs
linedist <- melt(linedist,
                 id.vars = "plot",
                 value.name = "distance",
variable.name = "species") # convert from wide to long format

# make a data.frame with variables encoding for each "treatment"
grid <- expand.grid(plot = 1:ranpos$n,
   pattern = patterns,
   intens = intens,
   radvec = radvec,
   simvec = simvec)
grid$species <- factor(paste(grid$pattern,
   grid$intens,
   grid$radvec,
   grid$simvec,
   sep = ".")

# add these variables to the data.frame with shortest distances
linedist <- merge(grid, linedist)

# and set distance to maxdist if the species is further away than this distance
linedist$distance <- ifelse(linedist$distance >= maxdist,
   maxdist,
   linedist$distance)

Appendix S3: additional figures with examples of simulated plant distribution patterns

# more examples
 ggclup.all <- data.frame()
 for (i in levels(intens)) {
   for (j in levels(radvec)) {
     temp <- as.data.frame(clup[[i]][[j]][[1]])
     temp$intens <- i
     temp$radvec <- j
     temp$pattern <- "clu"
     ggclup.all <- rbind(ggclup.all, temp)
   }
   ggclup.all
 }

 ggunip.all <- data.frame()

 for (i in levels(intens)) {
   for (j in levels(radvec)) {
     temp <- as.data.frame(unip[[i]][[j]][[1]])
     temp$intens <- i
     temp$radvec <- j
     temp$pattern <- "uni"
ggunip.all <- rbind(ggunip.all, temp)
}
ggunip.all
}
ggregp.all <- data.frame()
for (i in levels(intens)) {
  for (j in levels(radvec)) {
    temp <- as.data.frame(regp[[i]][[j]][[1]])
    temp$intens <- i
    temp$radvec <- j
    temp$pattern <- "reg"
    ggregp.all <- rbind(ggregp.all, temp)
  }
}
ggregp.all
}

ggunip.all$intens <- factor(ggunip.all$intens,
  levels = c(50, 100, 500))
ggunip.all$radvec <- factor(ggunip.all$radvec,
  levels = c(5, 15, 30),
  labels = c("Radius = 0.005",
            "Radius = 0.015",
            "Radius = 0.030"))
ggregp.all$intens <- factor(ggregp.all$intens,
  levels = c(50, 100, 500))
ggregp.all$radvec <- factor(ggregp.all$radvec,
  levels = c(5, 15, 30),
  labels = c("Radius = 0.005",
            "Radius = 0.015",
            "Radius = 0.030"))

ggclup.all$intens <- factor(ggclup.all$intens,
  levels = c(50, 100, 500))
ggclup.all$radvec <- factor(ggclup.all$radvec,
  levels = c(5, 15, 30),
  labels = c("Radius = 0.005",
            "Radius = 0.015",
            "Radius = 0.030"))
Figure S3.1: Examples of simulations based on random point patterns with varying point pattern intensity and disc radius. Each combination was replicated five times in the simulation experiment.
Figure S3.2: Examples of simulations based on regular point patterns with varying point pattern intensity and disc radius. Each combination was replicated five times in the simulation experiment.
Figure S3.3: Examples of simulations based on clustered point patterns with varying point pattern intensity and disc radius. Each combination was replicated five times in the simulation experiment.

Appendix S4: Fitting the zero-and-one inflated beta distribution to data from the simulation experiment

Circular approach

Important notes:

- The calculated $y$ in the R-code below is $1 - y$ in the paper (cf. the final paragraph of the section entitled *Integral occurrence probability: a new measure that combines the concepts of cover and relative frequency* in the main text of the paper. We choose here to directly model the point-to-plant distances in terms of relative annulus area because than the mean of the zero-and-one inflated beta equals the integral occurrence probability.

- The code in Appendix S2 needs to be executed first in order to be able to run this code.

```r
# transformation of point-to-plant distances
# we choose here for relative annulus area because than
# the mean of the zero-and-one inflated beta equals
# the integral occurrence probability
shortdist.sim1$y <- 1 - shortdist.sim1$distance ^ 2 / maxdist ^ 2
```
# modelling each parameter as a function of species without intercept
# (~ 0 + species) gives separate coefficient estimates for each species.
# The coefficients are the same if we would run separate models y ~ 1
# for subsets of the data restricted to one species at a time
# additional control settings can be necessary for model convergence
con <- gamlss.control(n.cyc = 100, trace = FALSE)
m.sim1 <- gamlss(y ~ 0 + species,
    sigma.formula = ~0 + species,
    nu.formula = ~0 + species,
    tau.formula = ~0 + species,
    data = shortdist.sim1,
    family = BEINF(), control = con)

results.sim1 <- data.frame(species = levels(shortdist.sim1$species),
    mu = plogis(coef(m.sim1)), # mean of the beta part
    sigma = plogis(coef(m.sim1, parameter = c("sigma"))),
    tau = exp(coef(m.sim1, parameter = c("tau"))),
    nu = exp(coef(m.sim1, parameter = c("nu"))))

# the proportion zero and ones
results.sim1$p10 <- with(results.sim1, (tau + nu)/(1 + tau + nu))
# the proportion ones, this is an estimate of cover like in pin-point methods
results.sim1$p1 <- with(results.sim1, tau * (1 - p10))
# the proportion zeroes
results.sim1$p0 <- with(results.sim1, nu * (1 - p10))

results.sim1$relfreqatrmax <- 1 - results.sim1$p0
results.sim1$meanbeinf <- with(results.sim1, (tau + mu) / (1 + nu + tau))
results.sim1$varbeinf <- with(results.sim1,
    p0 * p1/(p0 + p1) +
    (1 - p0 - p1) * mu * (1 - mu) * sigma^2 +
    (p0 + p1) * (1 - p0 - p1) * (p1/(p0 + p1) - mu)^2)

columns <- reshape2::colsplit(results.sim1$species,
    pattern = "\.",
    names = c("pattern", "intens", "radvec",
    "simvec"))
results.sim1 <- cbind(columns, results.sim1)
results.sim1$pattern <- factor(results.sim1$pattern,
    levels = c("clu","uni","reg"),
    labels = c("Clustered", "Random", "Regular"))
results.sim1$intens <- factor(results.sim1$intens)
results.sim1$radvec <- factor(results.sim1$radvec,
levels = c("5","15","30"),
labels = c("0.005","0.015","0.030"))

results.sim1$Radius <- results.sim1$radvec
results.sim1$simvec <- factor(results.sim1$simvec)
results.sim1$nreps <- ranpos$n

# parametric bootstrap
# use install.packages("magrittr") if this line does not work
`%>%` <- magrittr::`%>%`

results.sim1 <- results.sim1 `%>%`
dplyr::group_by(species) `%>%`
dplyr::do(broom::tidy(replicate(500,
  mean(gamlss.dist::rBEINF(n = .$nreps,
    mu = .$mu,
    sigma = .$sigma,
    tau = .$tau,
    nu = .$nu)),
    simplify = "array"))) `%>%`
dplyr::rename(IOP = x) `%>%`
dplyr::summarise(IOP.lwr = quantile(IOP, probs = 0.025),
  IOP.upr = quantile(IOP, probs = 0.975)) `%>%`
dplyr::right_join(results.sim1)

Line transect approach

The same, but this time using the line transect data and transformation to $1 - \frac{r}{r_{\text{max}}}$. 

# transformation of point-to-plant distances for line transects
linedist$y <- 1 - linedist$distance / maxdist

# we have to remove simulated species that
# remain undetected when line transects are used
# we can do this by removing species for which variance of distance is 0
linedist2 <- ddply(linedist, .(species), summarise,
  zerovardist = var(distance) == 0)
linedist <- merge(linedist, linedist2)
linedist <- droplevels(linedist[linedist$zerovardist == FALSE,])

# this time we did explicit separate regressions in a for-loop
# because there were problems fitting all at once
# the try() function captures any errors if the model does not fit
# additional control settings can be necessary for model convergence
con <- gamlss.control(n.cyc = 100, trace = FALSE)
m.lines <- lapply(
  levels(linedist$species),
  function(i){

try(gamlss(y ~ 1,
    sigma.formula = ~1,
    nu.formula = ~1,
    tau.formula = ~1,
    data = linedist[linedist$species == i,],
    family = BEINF(), control = con))
)

names(m.lines) <- levels(linedist$species)

# remove species for which gamlss resulted in an error
m.lines <- m.lines[sapply(m.lines, inherits, what = "gamlss", USE.NAMES = TRUE)]

# extract estimated parameters in link scale
results.lines <- ldply(.data = m.lines, .fun = function(x){summary(x)[1:4,1]})

colnames(results.lines) <- c("species", "mu.link",
    "sigma.link", "nu.link", "tau.link")

# calculate parameters in response scale
results.lines <- ddply(results.lines, .(), transform,
    mu = plogis(mu.link), # mean of the beta part
    sigma = plogis(sigma.link),
    tau = exp(tau.link),
    nu = exp(nu.link))

# the proportion zero and ones
results.lines$p10 <- with(results.lines, (tau + nu)/(1 + tau + nu))
# this is an estimate of cover like in pin-point methods
results.lines$p1 <- with(results.lines, tau * (1 - p10))
# the proportion zeroes
results.lines$p0 <- with(results.lines, nu * (1 - p10))

results.lines$relfreqatrmax <- 1 - results.lines$p0

results.lines$meanbeinf <- with(results.lines,
    (tau + mu) / (1 + nu + tau))
results.lines$varbeinf <- with(results.lines,
    p0 * p1/(p0 + p1) + (1 - p0 - p1) * 
    mu * (1 - mu) * sigma ^ 2 + 
    (p0 + p1) * (1 - p0 - p1) * 
    (p1/(p0 + p1) - mu) ^ 2)

columns <- reshape2::colsplit(results.lines$species,
    pattern = "\\.",
    names = c("pattern", "intens",
    "mu", "sigma", "nu", "tau")
)
results.lines <- cbind(columns, results.lines)
results.lines$pattern <- factor(results.lines$pattern,
    levels = c("clu", "uni", "reg"),
    labels = c("Clustered", "Random", "Regular"))
results.lines$pattern <-
results.lines$intens <- factor(results.lines$intens)
results.lines$radvec <- factor(results.lines$radvec,
    levels = c("5", "15", "30"),
    labels = c("0.005", "0.015", "0.030"))
results.lines$Radius <- results.lines$radvec
results.lines$simvec <- factor(results.lines$simvec)
results.lines$nreps <- ranpos$n

# parametric bootstrap
#use install.packages("magrittr") if this line does not work
```r
results.lines <- results.lines %>%
    dplyr::group_by(species) %>%
    dplyr::do(broom::tidy(replicate(500,
        mean(gamlss.dist::rBEINF(n = .nreps,
            mu = .mu,
            sigma = .sigma,
            tau = .tau,
            nu = .nu)),
        simplify = "array"))) %>%
    dplyr::rename(IOP = x) %>%
    dplyr::summarise(IOP.lwr = quantile(IOP, probs = 0.025),
        IOP.upr = quantile(IOP, probs = 0.975)) %>%
    dplyr::right_join(results.lines)
```

**Point-intercept approach**

# transform distance values to presence-absence at each sampling point
shortdist.sim1$pinpoint <- ifelse(shortdist.sim1$distance == 0, 1, 0)

# remove species that would not have been observed with point-intercept method
shortdist.sim2 <- shortdist.sim1
shortdist.sim2 <- shortdist.sim2 %>%
    dplyr::group_by(species) %>%
    dplyr::filter(sum(pinpoint) > 0) %>%
    droplevels()

m.point <- lapply(
    levels(shortdist.sim2$species),
    function(i){
        ...


```r
try(glm(pinpoint ~ 1,
        data = shortdist.sim2[shortdist.sim2$species == i,],
        family = binomial))
}
)

names(m.point) <- levels(shortdist.sim2$species)

# remove species for which glm resulted in an error
m.point <- m.point[sapply(m.point, inherits, what = "glm", USE.NAMES = TRUE)]

# extract estimated parameters in link scale
results.point <- ldply(.data = m.point, .fun = function(x){coefficients(x)})

colnames(results.point) <- c("species", "mu.link")

columns <- reshape2::colsplit(results.point$species,
                               pattern = "\.",
                               names = c("pattern", "intens",
                                         "radvec", "simvec"))

results.point <- cbind(columns, results.point)
results.point$mu <- plogis(results.point$mu.link)
results.point$pattern <- factor(results.point$pattern,
                                 levels = c("clu", "uni", "reg"),
                                 labels = c("Clustered", "Random", "Regular"))
results.point$intens <- factor(results.point$intens)
results.point$radvec <- factor(results.point$radvec,
                                levels = c("5", "15", "30"),
                                labels = c("0.005", "0.015", "0.030"))
results.point$Radius <- results.point$radvec
results.point$simvec <- factor(results.point$simvec)
results.point$nreps <- ranpos$n

# parametric bootstrap

results.point <- results.point %>%
    dplyr::group_by(species) %>%
    dplyr::do(broom::tidy(replicate(500,
                              mean(rbinom(n = .nreps, size = 1, prob = .mu)),
                              simplify = "array"))) %>%
    dplyr::rename(cover = x) %>%
    dplyr::summarise(cover.lwr = quantile(cover, probs = 0.025),
```

---

**Note:** The code snippet above is a continuation of the previous one, focusing on various ecological modeling techniques, including the use of `glm` for species distribution modeling, parameter extraction, and parametric bootstrap methods. The script demonstrates how to handle errors gracefully, extract and transform estimated parameters, and perform simulations to assess model performance.
Comparison of circular, line transect and point-intercept approach

```r
cover.upr = quantile(cover, probs = 0.975)) %>%
dplyr::right_join(results.point)
```

**Comparison of circular, line transect and point-intercept approach**

```r
ggplot(results.lines, aes(x = simvec)) +
  geom_pointrange(shape = 15, aes(ymin = IOP.lwr, y = meanbeinf, ymax = IOP.upr)) +
  geom_pointrange(data = results.sim1, shape = 17, aes(ymin = IOP.lwr, y = meanbeinf, ymax = IOP.upr)) +
  geom_pointrange(data = results.point, aes(ymin = cover.lwr, y = mu, ymax = cover.upr)) +
  ylab("Predicted mean value +/- 95% confidence interval") +
  xlab("Replicate") +
  facet_grid(pattern ~ intens + radvec, scales = "free_x")
```

**Figure S4.1:** Predicted values and 95% confidence intervals for the circular approach, a line-based approach and point-intercept sampling.

```r
comparison <- results.sim1 %>%
dplyr::select(species, pattern, intens, radvec, meanbeinf, IOP.lwr, IOP.upr) %>%
dplyr::rename(estimate = meanbeinf, lwr = IOP.lwr, upr = IOP.upr) %>%
dplyr::mutate(CIwidth = upr - lwr,
              RelMarginError = CIwidth/2/estimate,
              Approach = "Circular") %>%
dplyr::bind_rows(results.lines %>%
dplyr::select(species, pattern, intens, radvec, meanbeinf,
```

```r
...```
The following figure shows the increase in relative margin of error when a line-based approach is used instead of a circular approach and when classical point-intercept is used. On average, we see an increase in the relative margin of error from 0.25 over 0.55 to 0.73. The relative margin of error is the half-width of the confidence interval for the integral occurrence probability divided by the estimated integral occurrence probability. Thus, for a line-based approach (where we use for the length of the line segment the same maximum search distance as in the circular approach), the confidence interval will be approximately $\pm 0.55 \times \text{the estimated integral occurrence probability}$. In relative terms this will be more than twice as wide compared to the circular approach and a 4.8 times larger sample size will, on average, be needed to obtain a relative margin of error that is comparable to the circular approach. The point-intercept approach will need, on average, an 8.5 times larger sample size to have a relative margin of error that is comparable to the integral occurrence probability with the circular approach.
Figure S4.2: Increase in relative margin of error (RME) from circular over line to point-based approach. RME values for the same simulated species are connected by a grey line. The mean RME and 95% confidence interval is shown in black.

The next figure shows that the estimated integral occurrence probability obtained through a line based approach is always lower than the one obtained with the plot-based approach. Hence, the estimated integral occurrence probability obtained with the line-based approach will be closer to the cover estimate (point-intercept sampling - this is logical because we move from an area, over a line to a point). Again, in the limit, when the search distance goes to zero, the integral occurrence probabilities coincide with cover.

```R
ggplot(comparison, aes(x = Approach, y = estimate)) + geom_line(aes(group = species, colour = "grey")) + stat_summary(fun.data = "mean_cl_boot") + ylab("Estimate (circular IOP, line IOP or cover") + facet_grid(pattern ~ intens + radvec) + theme(axis.text.x = element_text(angle = 90, vjust = 0.5), axis.title.x = element_blank())
```
Figure S4.3: Estimated values (integral occurrence probability in a circle or on a line or cover) for circular, line and point-based approach. Estimated values for the same simulated species are connected by a grey line. The mean and 95% confidence interval is shown in black.

Appendix S5: occurrence probability as a function of relative squared radius (cumulative curves) for data from the simulation experiment

The code in this Appendix depends on code in Appendix S2 and S4.

```r
yvec <- seq(0, 1, 0.0025)  # a vector of relative squared annulus area values
# we don't want the cumulative curve to plot the point mass at 1
yvec[yvec == 1] <- 1 - .Machine$double.eps

grid3 <- expand.grid(pattern = patterns, intens = intens, radvec = radvec,
                      simvec = simvec, yvec = yvec)

grid3$intens <- factor(grid3$intens)
grid3$radvec <- factor(grid3$radvec,
                      levels = c("5", "15", "30"),
                      labels = c("0.005", "0.015", "0.030"))
grid3$pattern <- factor(grid3$pattern,
                      levels = c("uni", "reg", "clu"),
                      labels = c("Random", "Regular", "Clustered"))
```
cumulativecurves <- `merge`(results.sim1, grid3)

cumulativecurves <- `ddply`(cumulativecurves, .(pattern, intens, radvec, simvec),
                           transform,
                           relfreq = `pBEINF`(q = yvec,
                                              mu = mu,
                                              sigma = sigma,
                                              tau = tau,
                                              nu = nu,
                                              lower.tail = FALSE))

cumulativecurves$relsqrad <- 1 - cumulativecurves$yvec

ggplot(cumulativecurves, aes(x = relsqrad, y = relfreq, colour = intens)) +
  geom_line(aes(group = species)) +
  xlab(expression(r^2/r[max]^2)) +
  ylab("Occurrence probability (Relative frequency") +
  facet_grid(pattern ~ radvec)

Figure S5.1: Occurrence probability as a function of relative squared point-to-plant distance (radius).
Appendix S6: additional results for expected species richness

This code depends on Appendices S2, S4 and S5.

**All 135 simulated species combined**

```r
enscurve <- ddply(cumulativecurves, .(relsqrad), summarise,
                   expectedrichness = sum(relfreq))

ggplot(enscurve, aes(x = relsqrad, y = expectedrichness)) +
   geom_line() +
   xlab(expression(r^2/r[\text{max}]^2)) +
   ylab("Expected number of species")
```

![Graph showing expected species richness as a function of relative squared point-to-plant distance (radius).]

**Figure S6.1: Expected species richness as a function of relative squared point-to-plant distance (radius).**

**Grouped according to treatment**

```r
enscurve2 <- ddply(cumulativecurves, .(pattern, intens, radvec, relsqrad),
                    summarise,
                    expectedrichness = sum(relfreq))

ggplot(enscurve2, aes(x = relsqrad, y = expectedrichness)) +
   geom_line(aes(colour = pattern, linetype = pattern)) +
   xlab(expression(r^2/r[\text{max}]^2)) +
```

![Graph showing expected species richness grouped by treatment.]
**ylab("Expected number of species") + facet_grid(intens ~ radvec)**

![Graph showing expected species richness as a function of relative squared point-to-plant distance (radius) per treatment group.](image)

*Figure S6.2: Expected species richness as a function of relative squared point-to-plant distance (radius) per treatment group.*

**Appendix S7: Comparison with the Böcher-modified Raunkiaer method**

Damgaard (2015) shows how to analyse data where a series of concentrically placed circular plots of increasing size is used and for each species the smallest circular plot is noted in which the species is present. He uses combinatorial logic combined with a beta-binomial distribution to find the probabilities of observing each state of the discrete events (i.e. absence of a species, presence in the inner-most circular sampling unit, presence in one of the annuli). We show here how the likelihood of the distribution can be maximised using R code. We take a frequentist approach, although in the paper a Bayesain analysis procedure is suggested (both approaches give quasi the same results - results not shown, but can be requested from the first author). Next we show that this approach is equivalent with the methods presented in the present paper regarding the estimation of relative frequency in the inner-most circular plot.

We use the data from the simulation experiment to demonstrate this.

The code depends on Appendix S2.
Discretizing the data from the simulation experiment

We assume a series of four concentrically placed circular plots.

bocher: The recorded value of the smallest circle where the species is present; bocher = 0 denotes the event that the species is absent from the plot; bocher = 4 denotes presence in the smallest circle, ..., bocher = 1 denotes presence in the largest circle.

n: The area of the plot (=the area of the largest circle) is n times the area of the smallest circle.

d: Size vector \( d = 1, \ldots, d_i, d_j, \ldots, n \), which is the area of a circle divided by the area of the smallest circle in increasing order of the circle areas.

\[
\begin{align*}
rads & \leftarrow c(\maxdist/4, \maxdist/\sqrt{8}, \maxdist/2, \maxdist) \\
radsarea & \leftarrow \pi \times rads^2 \\
d & \leftarrow radsarea/radsarea[1] \quad \# \text{ size vector} \\
n & \leftarrow \text{tail}(d,1)/\text{head}(d,1) \quad \# \text{ area of the largest circle} \\
\end{align*}
\]

\# recode 4 = smallest circle, ...
shortdist.sim1$bocher <- \text{cut}(\text{shortdist.sim1$distance}, \text{breaks} = c(0, rads), \\
\quad \text{include.lowest} = \text{TRUE}, \text{labels} = c(4, 3, 2, 1))
shortdist.sim1$bocher <- \text{as.numeric}(\text{as.character}(\text{shortdist.sim1$bocher}))

\# when shortdist.sim1$distance == maxdist the species was absent
shortdist.sim1$bocher <- \text{ifelse}(\text{shortdist.sim1$distance} \geq \maxdist, 0, \\
\quad \text{shortdist.sim1$bocher})

In the next code chunk the following parameters are estimated with maximum likelihood:

\( p \): Estimated mean probability of observing the species in a subplot with an area equal to the smallest circle.

\( \delta \): Estimated intra-plot correlation parameter that measures the within-site spatial aggregation of the plant species as the correlation between the outcomes of successive Bernoulli trials Qu et al. (1993)

\# the negative loglikelihood:
\# gamlss beta-binomial (dBB)
\# parametrization \( p = \mu, \delta = \sigma / (1 + \sigma) \) or \( \sigma = \delta / (1 - \delta) \)

\[
\text{neglogllh} \leftarrow \text{function}(y, p, \delta, n, d, \ldots) \{ \\
\quad \mu \leftarrow p \\
\quad \sigma \leftarrow \delta / (1 - \delta) \\
\quad p0 \leftarrow \text{dBB}(x = 0, \ bd = n, \ mu = \mu, \ sigma = \sigma, \ log = \text{FALSE}) \\
\quad \text{output} \leftarrow \text{rep}( \\
\quad \quad \text{dBB}(x = 0, \ bd = n, \ mu = \mu, \ sigma = \sigma, \ log = \text{TRUE}), \\
\quad \text{shortdist.sim1$bocher})
\}
\]
length(y)
)

r <- seq_len(n)

output[y == 4] <- log(sum(r/n * dBB(x = r, bd = n, mu = mu, sigma = sigma)))
output[y == 3] <- log(sum((1 - r/n - choose(n - d[2], r)/choose(n, r)) *
                        dBB(x = r, bd = n, mu = mu, sigma = sigma)))
output[y == 2] <- log(sum((choose(n - d[2], r) - choose(n - d[3], r)) /
                          choose(n, r) *
                          dBB(x = r, bd = n, mu = mu, sigma = sigma)))
output[y == 1] <- log(sum(choose(n - d[3], r) / choose(n, r) *
                          dBB(x = r, bd = n, mu = mu, sigma = sigma)))

return(-sum(output))
}

# estimation of p and delta with mle2
# function mle2() needs the negative log-likelihood

# put p and delta in logit scale to assure they are bounded between
# 0 and 1 in original scale

neglogllh2 <- function(logit.p, logit.delta, y, n, d) {
  neglogllh(p = plogis(logit.p), delta = plogis(logit.delta), y, n, d)
}

# using the neglogllh in mle2()

resBocher <- data.frame(species = unique(shortdist.sim1$species))
resBocher$p <- NA
resBocher$delta <- NA

shortdist.sim1.m <- dcast(shortdist.sim1, plot ~ species,
                           fun.aggregate = NULL, value.var = "bocher")
shortdist.sim1.m$plot <- NULL

for (i in 1:ncol(shortdist.sim1.m)) {
  opt <- mle2(neglog1lh2,
              start = list(logit.p = 0, logit.delta = 0),
              data = list(y = shortdist.sim1.m[,i], n = n, d = d),
              control = list(maxit = 1000),
              method = "Nelder-Mead")
  resBocher$p[i] <- plogis(coef(opt)[1])
  resBocher$delta[i] <- plogis(coef(opt)[2])
}
# compare p to the results based on zero-and-one inflated beta:
resBEINF <- subset(cumulativecurves, 
  relsqrad == head(rads, 1) ^ 2/tail(rads, 1) ^ 2)

resJoin <- merge(resBocher, resBEINF)

ggplot(resJoin, aes(x = relfreq, y = p)) + 
  geom_point() + 
  geom_smooth() + 
  geom_abline() + 
  xlab("Occurrence probability in circle with radius 0.01 based on point-to-plant distances") + 
  ylab("Occurrence probability in circle with radius 0.01 based on four concentrically placed circles of increasing size")

Figure S7.1: Scatterplot of estimates of occurrence probabilities based on either the Böcher-modified Raunkiaer method or point-to-plant distances. A line through the origin with slope 1 and a smoother with 95% confidence bounds are displayed.

References