PatchOccupancy – a FORTRAN program for the study of species spatial distributions

Version 4.1

Werner Ulrich

Nicolaus Copernicus University in Toruń
Department of Animal Ecology
Gagarina 9, 87-100 Toruń; Poland
e-mail: ulrichw @ uni.torun.pl

Latest update: 30.06.2003

1. Introduction

Patch occupancy models are today one of the major tools for the study of spatial (and temporal) distributions of animal or plant species (Gotelli and Graves 1996, Hanski 1991, 1999, Hanski and Gilpin 1997, Hanski et al. 1993, Ulrich 1999a, b, 2000a-d, Wright 1991). The main idea behind this approach is simple. Take a community of S items (say species) with N replicates each (say individuals) and place them either randomly or according to a predefined model into the cells of a large grid. These cells may have different properties. You can now take samples out of this grid or parts of it, study the resulting community structure based not on the whole community but on these samples and compare these structure with the ones found in nature. By this way patch occupancy models are able to generate realistic null models in ecological research and with their aid a variety of ecological patterns might be studied:

Species—area relationships and related patterns (SAR)

Individuals—area relationships (IAR)

Relative abundance distributions (RADs)

Various estimators of community structure based on samples like estimators of species numbers, diversity and evenness, species overlap, detection probabilities, aggregation, and spatial patterns

Patch occupancy dynamics and interpatch migrations

Extinction and colonization rates

Macroecological patterns, for instance biomass distributions

Patterns related to habitat structure.

These examples demonstrate the wide range of potential applications patch occupancy models have. They also show that such models to be realistic have to incorporate a manifold of parameters that determine community and habitat properties. The difficulties around the application of these models then stem mainly from the choice of parameter values. Unrealistic or highly simplified model setting will always result in unreliable results. The program *Patch Occupancy* is an attempt for a general solution that allows computing a wide variety of different types of models with realistic initial settings. The program either uses data sets of real communities, theoretical communities generated by certain RAD models (Ulrich

2000d, 2002), or it generates model communities using a few basic community properties. Cell properties can be varied in such a way that habitat heterogeneity and carrying capacities can be modelled realistically. Additionally, the program computes a set of basic community statistics.

2. Generating communities with Patch Occupancy

Generally, model species assemblages are generated using underlying relative abundance relationships (often log-normal or log-series models, May 1975, Leitner and Rosenzweig 1997, Wilson et al. 1998, Ulrich 2002). However, to get a wide range of 'realistic' model assemblages here another approach is undertaken: Real species assemblages are characterized by three basic features. They have a certain species - weight distribution (SWD) [nearly always normal or log-normal (Novotny and Kindlmann 1996)] and an abundance - weight distribution (AWD; often also termed density—weight distribution DWD) with an upper boundary line (most often described by a power function), which has frequently - but not always - a triangular shape (Currie & Fritz 1993, Currie 1993, Ulrich 1999c). Thirdly, species do not have infinitely low densities, there is a lower density limit below which a species goes locally extinct.

Therefore, *Patch Occupancy* first generates model communities (assemblages) from these three features. Firstly, a SWD is generated from three based input variables: the number of logarithmic body weight classes, the modal body weight class and the variance. SWDs can be defined as normal (input *normal*) or as lognormal (input *logn*).

In a next step a DWD has to be defined. Typical DWDs are power functions (input *power*) with a negative slope (Ulrich 1999c). *Patch Occupancy*, however, allows also other distributions to be computed: a linear (input *linear*), a normal (*normal*), a lognormal (*logn*), or a random (*rnd*). The DWD has then to be defined by a slope (in the *power* or *linear* case) or by a mode and a variance (Fig. 1).

A third computing step then assigns species densities. In other words, the species of assemblages defined by with these two basic features are assigned densities inside the above defined boundaries. In reality species with very high and very low densities are less often found than ones with intermediate abundances. Sometimes, low densities are more probable than high ones. To simulate such a pattern

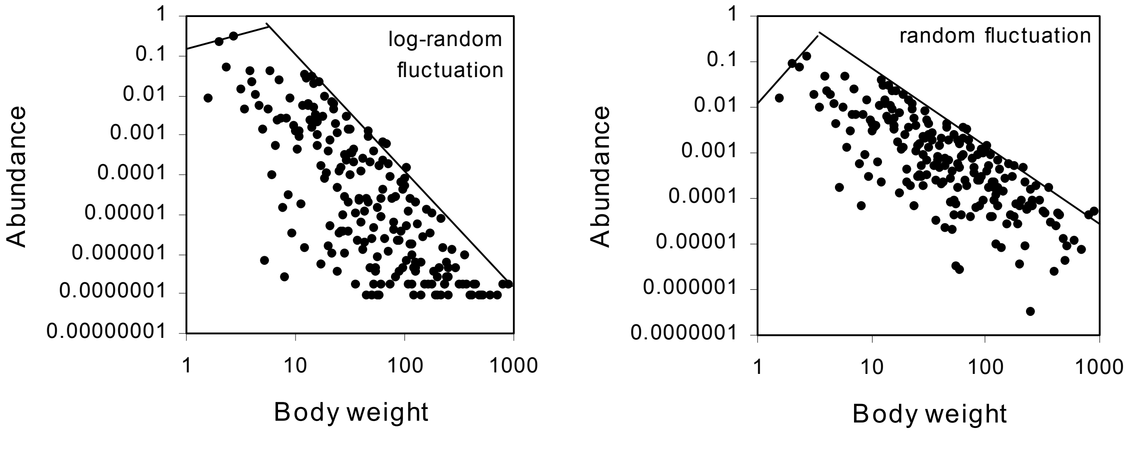


Figure 1: Abundance – weight distributions (AWD) of 2 model communities with 200 species each and a SWD with a median of 8 and a variance of 2. A: abundances were assigned with random numbers on a logarithmic scale; B: abundances resulted from random numbers on a linear scale.

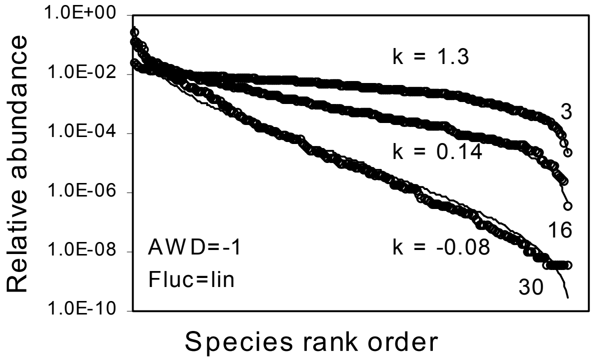


Figure 2: Abundance—weight distributions of three model assemblages generated by the three step method described in the text. We see that typical RADs appear that may be fitted by common models of relative abundance. In this case the power fraction model of Tokeshi (1996) was used for fitting (using the program RAD, Ulrich 2002). The *k*-values refer to the shaping parameter of this model. A power fraction model with $k = 0.1$ would be similar to a canonical log-normal, $k = 1.0$ is nearly identical to a broken stick model. The AWD has a slope of –1, linear random numbers were used to assign densities and model species body weights of the assemblages span over 3 to 30 binary weight classes.

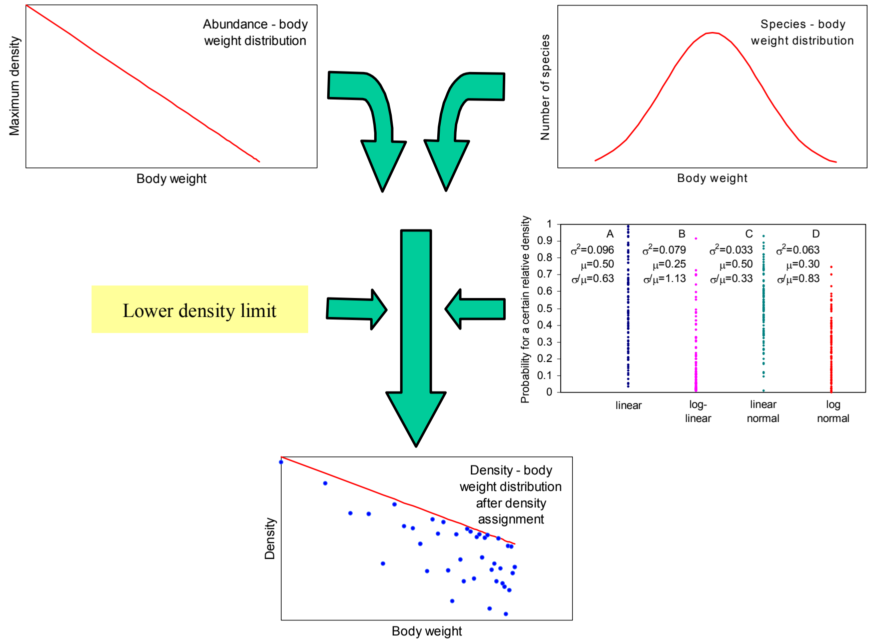


Figure 3: A three step process generates density—weight distributions and also relative abundance distributions. In this process two basic body weight dependent ecological distributions work together. In a first step a density—weight distribution is defined. A second step then assigns species from a given species—weight distributions hypothetical body weights. Random density fluctuations of these species together with an also given lower density limit result then in realistic RADs and DWDs.

densities are either assigned by random fluctuations with log-transformed densities (resulting in a higher number of low density species) or using normally distributed random numbers (resulting in a higher number of intermediate densities). Additionally, simple linear random numbers can be used to assign densities. An example of the resulting relative abundance distributions contains Fig. 2. This Figure also shows that the assemblages have typical density-weight distributions (Currie 1993, Ulrich 2001). The main idea behind the whole process is again exemplified in Fig. 3.

However, it is also possible to run *Patch Occupancy* either with real data about relative abundances and species body weights or with theoretical distributions generated by other programs like RAD (Ulrich 2002).

3. Running the program

3.1. Computing a model assemblage

After program start *Patch Occupancy* first asks ‘What to do?’. You can either generate only a relative abundance distribution (input: *m*) as described above, or sampling a generated or given distribution from the cells of a grid (see below) (input: *s*), or printing the basic batchfile *Patch.txt* (Tab. 1) with which you can run *PatchOccupancy* (input: *b*).

If you want to sample a grid the program asks whether you want to print the surface structure of the grid (see below). If you answer *yes PatchOccupancy* will print the mean individual numbers per

Table 1: The batch file *Patch.txt* generated by *PatchOccupancy*. The file contains all the necessary parameter values to run the program. Other filenames are also allowed, the input needs not to be formatted only the sequence of variables has to be kept. The number of input lines is unlimited.

AreaI	SampA	maxSa	minSa	StepSa	Samp-fac	Method	Spe-cies	maxD	minD	Dfluc	Flucmy	Fluc-StD	DWD	SWD	Class	SWD-my	SWD-StD	DWDmy	DWDStD	Aggreg	Area-Het	Niche N	Rnd-Het	FracM	Scale1	Scale2	Range	Gradi-ent	DivInd	Permut	Grid-Het
20	20	10	4	2	1	dep	20	10	0.001	0	0.5	0.01	power	nor-mal	6	3	1	0.9	0	2	5	5	rnd	rnd	2	3	0.5	10.000	sha	20	fix
1000	1	1	1	1	1	ind	1	0.1	0.001	0	0.5	0.02	linear	logn	20	5	2	5	0	1	10	0	fix	rnd	1	4.3	1	0.0	simp	10	rnd
100	100	100	100	1	1	seq	1000	10	0	0	0.5	0.01	rnd	logn	30	5	1	1	0.2	0.5	20	5	rnd	super	2.5	3	0	0.000	evar	2	fix
20	20	100	100	1	1	seq	100	1	0	0	0.5	0.01	normal	nor-mal	10	5	2	10	1	3	20	5	fix	super	2	3	0.5	50.0	sha	20	fix
100	100	100	100	1	1	seq	100	1	0	0	0.5	0.01	logn	nor-mal	10	5	1	1	0.1	3	20	5	fix	rnd	2	3	0.9	1.0	evar	20	rnd

species that can be placed into the cells per placing step (see below). The respective output file is *Grid.txt*. Essentially, the values printed show the grid surface structure.

After the model option *PatchOccupancy* asks whether you want to run the program by hand or via a batch file. Table 1 shows an example of this batch file that is printed by the program. Batch files allow multiple modelling during one program run.

To compute a model the following input options are needed.

Area and SampA: Define by this option a hypothetical area (the grid) in which the species occur. An input of a *Area*-value i defines the area i * i cells, an input of 10 defines therefore a grid of 100 cells. Sometimes it is convenient to sample not the whole grid but a predefined part of it. With the option *SampA* you define such a part (located randomly inside the total grid). For instance, if you defined the size of the grid with *Area* = 1000 you can define an area to be sampled of S*ampA* = 10. The effect is that only from the cells from i =*ran* to *ran* + 10 and from j = *ran* to *ran* + 10 samples will be taken

Species: Give the number of species of the assemblage. The maximum number is 10000.

maxD: Here, you define the mean density per cells p_1 (this is the input value) of the most abundant species. The total number of individuals of that species is calculated as *maxD* = p_1*i*i. If you coose the input value 0 the program adjusts all densities so that the least abundant species will have exactly 1 individual.

minD: Sets the mean density of the least abundant species. Together these two values define the range of densities of the community. The input 0 sets the minimum density to 1 individual in the whole grid (*minD* = 1/(i*i).

DWD: Here you define the type of the density—weight distribution (or abundance—weight distribution) that defines the upper range of possible densities for each species. *PatchOccupancy* can assign densities using five different types of models. Most often used (and most often found in nature) is a power function type of DWD (input: *power*). In this case the DWD is assumed to follow the a power function: D =*maxD* * W^{-z}. You may also use a linear model (D = *maxD* – zW; input: *linear*), a normal model: D=*maxD* * exp(-(z - mean)^2/2var); input: *normal*), a log-normal model (D=*maxD* * exp(-(ln(z) - mean)^2/2var); input: *logn*), or a random model (D=*maxD*). The next two input variables define now the parameter values *z* and *var*.

DWDmy: This is the input value for *z*.

DWDStD: This is the input value for the standard deviation of the normal and log-normal

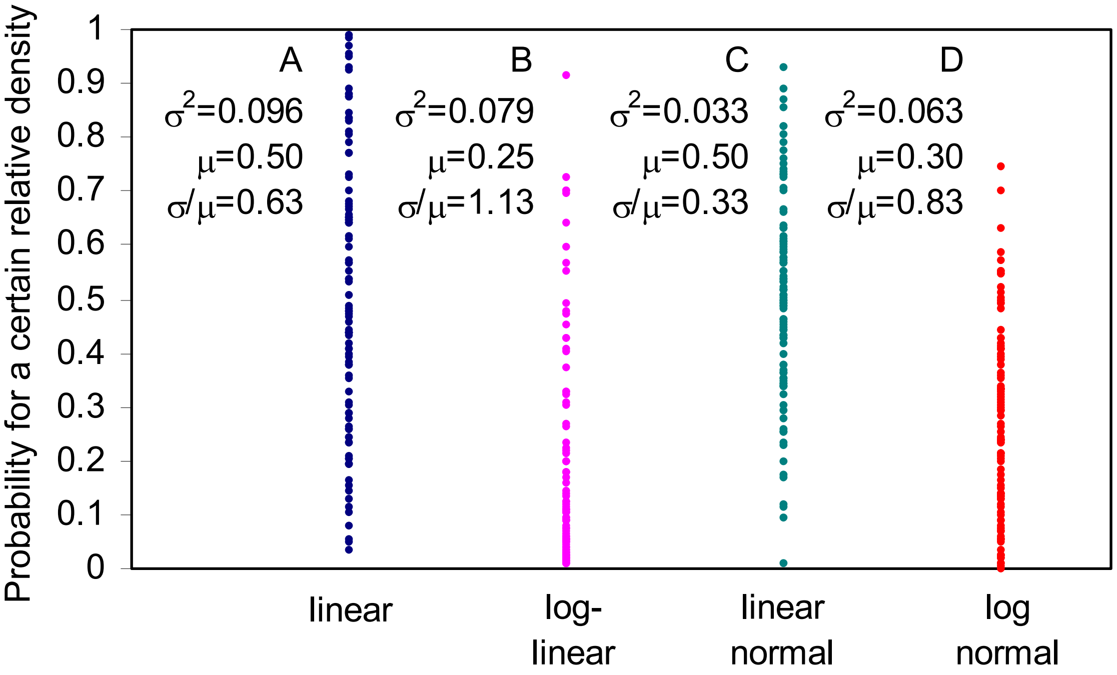


Figure 4: Four possibilities how densities may be assigned. In case A variability is a linear random variable, that means all densities between an upper and a lower boundary are equally probable; In B log transformed random number are used that lead to a lower mean density; C and D equal A and B but normal random numbers are used leading to a situation that intermediate densities (either on a linear or on a log scale) are more probable. Given are in each case the associated mean m, the variance s2 and the coefficient of variation CV (s / m) as a measure of variability. Note that CV is always higher after log transformations. Normal distributed densities around the mean reduce the value of CV.

model (*var=StD*^{2}). For *DWD* = *linear* and *DWDmy* and *DWDStD* both being 0 the program initially gives all species the same number of individuals (*maxD*)

Dfluc defines the type of random numbers by which densities are assigned. The input 0 lets the program using random numbers on a linear scale, the input 1 random numbers on a logarithmic scale and an input i between 0 and 1 causes the program to use for i% of species a logarithmic and for (100-i)% a linear scale. An input of –1 excludes any density fluctuation and uses the density boundary as the actual number of individuals.

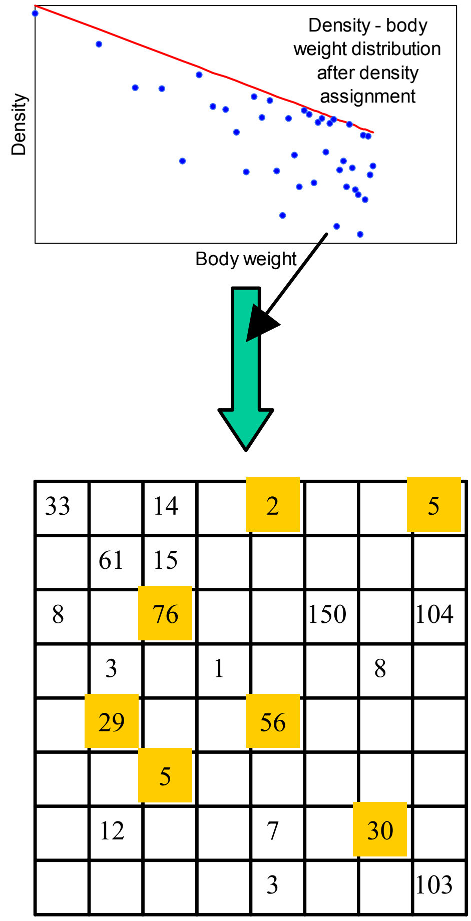
Flucmy: Random numbers used for density assignment may be linearly or normally distributed. The input 0 causes linear distributed random numbers, an input between 0 and 1 defines the mean of a normal distribution.

FlucStD: the input 0 defines again linear random numbers. An input between 0 and 1 defines the standard deviation of the normal distribution. The above three input parameters define therefore four different types of randomness by which densities might be assigned: linear and normal distributed random numbers either on a linear or on a logarithmic scale (Fig. 4).

The next four input variables define the species— weight distribution:

SWD defines the type of the species—weight distribution. You may either choose a normal distribution (over $\log_2$ body weight classes)(input: *normal*) or a log-normal distribution (input: *logn*). The species defined in *Species* are assigned hypothetical body weights according to the distribution below defined by the three parameters *Class*, *SWDmy*, and *SWDStD*.

Class defines the number of $\log_2$ body weight classes.



33		14		2			5
	61	15					
8		76			150		104
	3		1			8	
	29			56			
		5					
	12			7		30	
				3			103

Figure 5: Sampling cells of a grid of 64 cells. The individuals of the community defined in Figure 2 are spread onto a grid of 64 cells. This is shown for one highly aggregated species (marked by the arrow). Afterwards 7 cells (yellow) are sampled.

SWDmy defines the mean of the SWD.

SWDStD defines the standard deviation of the SWD. The species defined in *Species* are now assigned hypothetical body weights according to the above defined distribution.

The above parameters fully define a model assemblages of S species and result in species rank order distributions of which three examples are shown in Fig. 2.

3.2. Sampling individuals from a grid

If you compute a model assemblage as described above or you took data from a real community (see below) you may spread the individuals of this community into a grid defined by *Area*. Then, you may take samples (sample units are most often whole cells of this grid or combinations of cells) and analyse the resulting patterns. *PatchOccupancy* offers a wide range of options to define the cell properties as well as the way of sampling.

3.2.1 Defining the sample process

To define the sample process five input parameters are necessary. The program samples whole sets of cells during one program run. So, you may define a whole sampling program at once.

maxSa defines the maximum number of sample units to be sampled

minSa defines the minimum number of sample units to

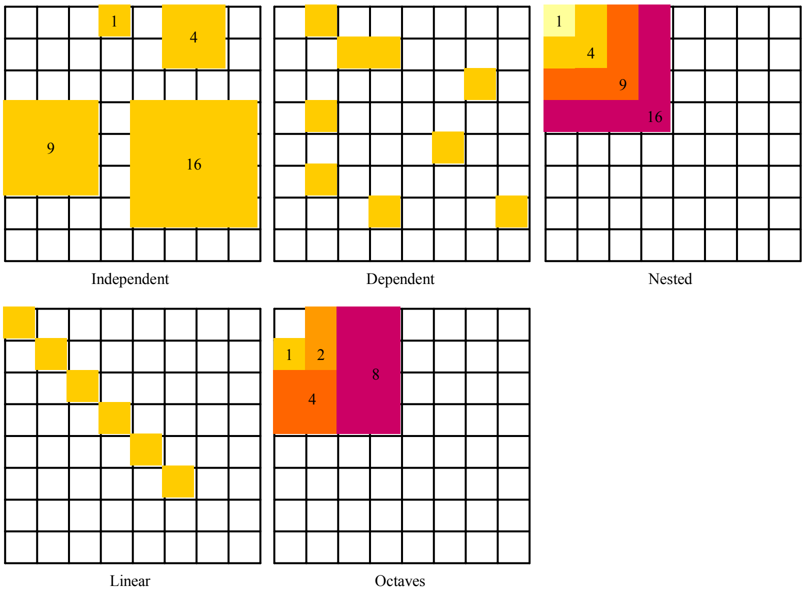


Figure 6: Fives types of sampling cells from a grid. Independent: In this case adjacent quadratic areas of different largeness are taken. The specification *independent* stems from the way the program computes species—area relationships. In this case SARs are computed from independent areas. Dependent: a series of basic units of sampling (most often single cells) are sampled. SARs are computed from random combinations of these cells (see below), therefore the classification dependent. Nested sampling starts from one corner and proceeds sequentially. Linear: samples one of the diagonals of the grid. Octaves: samples contingent independent areas of which the next larges is always two times as large as the one before. In this latter case the last sample is always the whole square in which the samples are (in this case the sum of samples with areas 1, 2, 4, and 8 plus the blank cell.

be sampled

StepSa defines the step width in the number of sample units to be sampled. For instance after the definition of *maxSa*=100, *minSa*=10 and *StepSa*=10 *PatchOccupancy* samples during one program run 10, 20, 30, 40,… 100 cells and analyses the resulting patterns independently for each step (for 10 cells, then for 20, for 30, and so on).

SampFac: Here you define the basic unit of sampling. The above example refers to *SampFac*=1. With *SampFac*=2 you would sample 10 times a square of 2 *2 adjacent cells, 20 times 4 adjacent cells, and so one. *SampFac* = 3 uses a square of 3*3 cells as the basic unit of sampling.

Method: *PatchOccupancy* offers five basic methods of sampling the grid (Fig. 6): a nested design (input *seq*), an independent design (input: *ind*), and a dependent design (input: *dep*), a linear design (*lin*), and sampling of independent octaves (*oct*). In the cases of *seq* and *ind* maximum sample sizes (defined by *maxSa*) are in cases of too high input values adjusted to maximum possible sample sizes.

3.2.2. Species spatial patterns

Species often occur in an aggregated manner. *PatchOccupancy* simulates this behaviour with the input parameter *Aggreg*. As the input you have to give the number of individuals placed together. The actual number of individuals placed is then either this number (*RndHet*: *fix*) or a normally distributed random number with mean *Aggreg* and variance *Aggreg* (*RndHet*: *rnd)*. The *Aggreg* value is typically similar to the value of the Lloyd index of mean crowding (see below) and realistic parameter values have therefore the same magnitude (0 to more than 10). ***PatchOccupancy* does not place all individuals of a species simultaneously during program run but places each individual or several individuals (as defined by *Aggreg* and *RndHet,* see below) sequentially.** For instance, of a species with in total 100 individuals and a value of *Aggreg* of 5 *PatchOccupancy* might place in the first step 6 individuals, in a second step 3, in a third step 5 and so on until all individuals are placed into cells of the grid. This allows a much more flexible treatment of the placement process.

3.2.3. Defining the grid and cell properties

Real habitats or local resource patches always differ more or less. For a realistic model differences in cell features have therefore to be included. *PatchOccupancy* offers for this task seven options.

AreaHet defines the degree of heterogeneity of the grid. This means in this case the degree by which the cells of the grid differ. *PatchOccupancy* models this difference by the number of individuals N of a certain species that can be placed into one cell at each placing step. The actual number of individuals placed is then either this number (*RndHet*: *fix*) or a normally distributed random number with mean *AreaHet* and variance *AreaHet* (*RndHet*: *rnd)*.

You can choose whether the grids struc-

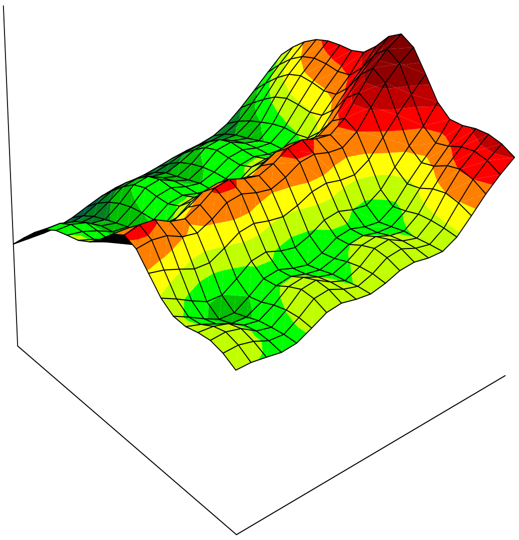


Figure 7: Cells of a grid with 20 * 20 cells may differ in the maximum number of individuals per species that can be placed in each step.

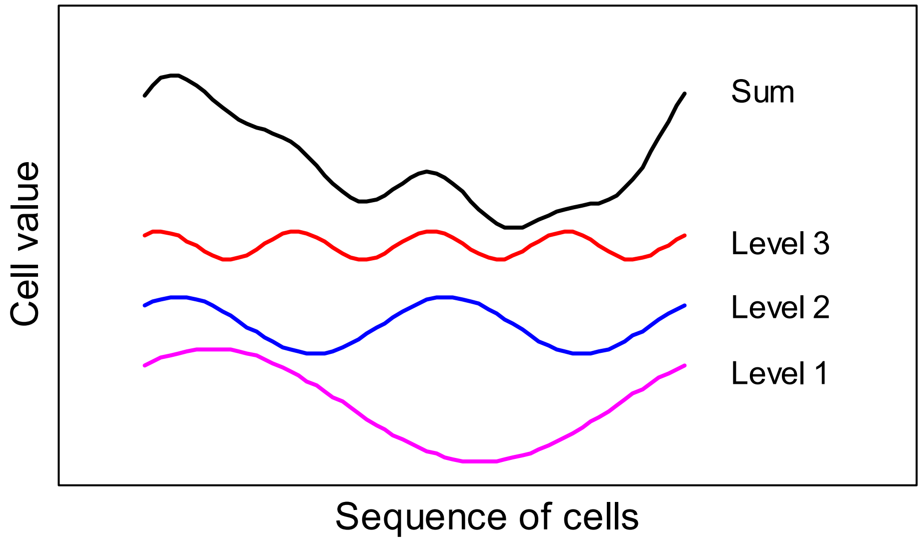


Figure 8: The superposition of trigonometric functions [in this case sin(x)=cos(x)] results in an irregular pattern of cell properties that that can be described by fractal geometry (for further details see Burrough 1983).

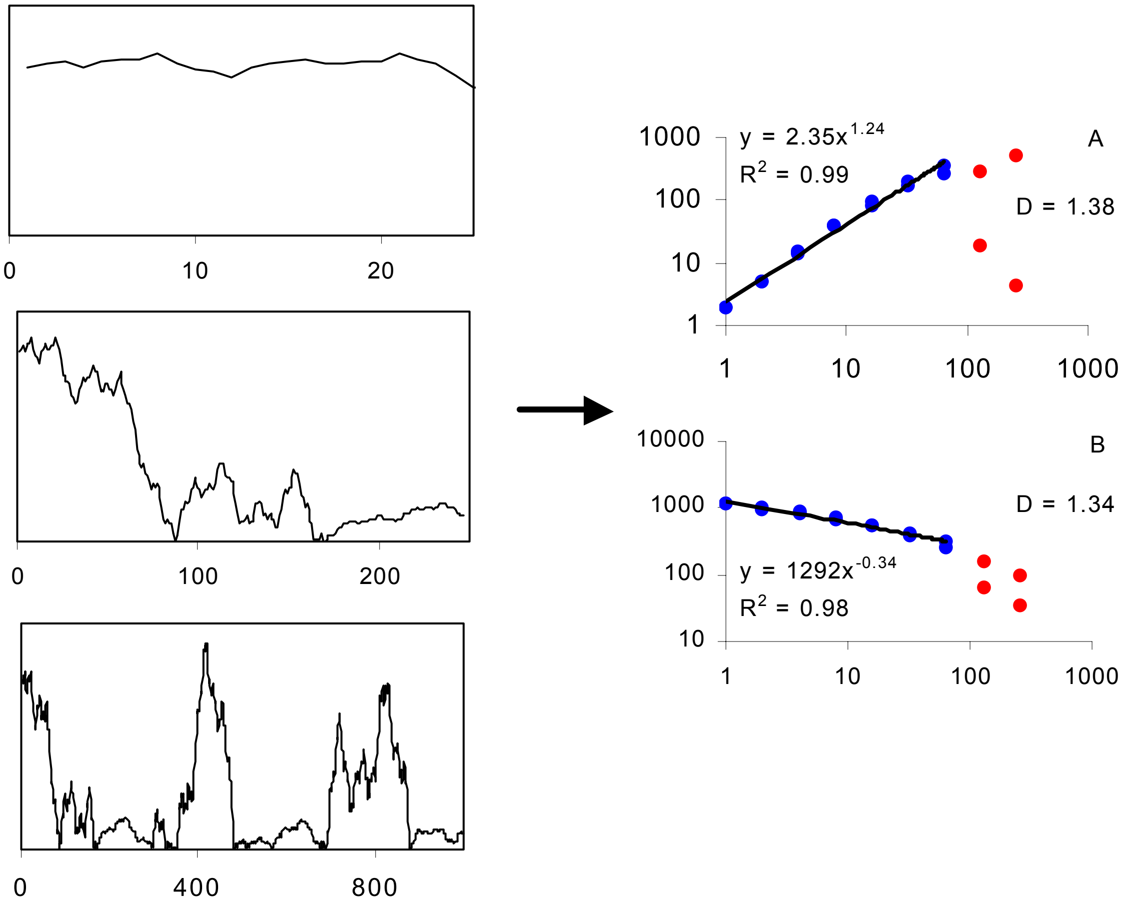


Figure 9: The diagonal of a grid (1000x1000 cells) with parameter settings that are shown in Table (*AreaHet* = 20, *RndHet* = rnd, *Scale1* = 2.2, *Scale2* = 2.7) seen under three different magnifications. From the two grid diagonals fractal dimensions are computed using either the semivariogram method (A) or the ruler length method. Note that the red data points already belong to a different scaling region. Patch Occupancy excluded them from computation.

ture is identical for all species (*Gridhet*: *fix*) or whether the grid structure is computed anew for each species (*GridHet*: *rnd*). The latter option results in a certain randomization of total species spatial distribution patterns.

Fig. 7 shows a simple example from a grid of 20 * 20 cells and *AreaHet* = 10. The input value *AreaHet* = 0 gives all cells the same properties and one individual will be placed during each step if not otherwise defined by *Aggreg* and *Range*.

PatchOccupancy is able to compute a wide range of different habitat types including fractal landscapes. The latter can be computed in two different ways, by a superposition of trigonometric functions (*FracM*: *super*) or by random midpoint displacement (*FracM*: *rmd*).

In the first case the basic computing procedure is a superposition process (Burrough 1983) that is similar to a discrete Fourier transformation (Hastings and Sugihara 1993). In this case this process is based on a superposition of trigonometric functions of the general form:

$$N = \sum_{i=1}^{k(scale1)} \frac{\text{AreaHet}}{\text{scale1}^{k-1}} \left(\cos(\frac{x}{\text{scale2}^{k}}) + \sin(\frac{y}{\text{scale2}^{k}}) \right)$$

In this equation x and y denote the coordinates of the grid, *Areahet*, *scale1*, and *scale2* are the parameters (see below), N is the number of individuals placed, and *k(scale)* the number of iterations that depends on the grid size and the scaling factor *scale1 (k = 2ln(Area)/ln(scale1))*.

In the case of the random midpoint displacement algorithm area sizes n are adjusted to values of $2^{n}+1$. Because this algorithm results in positive and negative values for landscape roughness values are shifted by *value*=*value**3**scale1* + *scale1*. As in the previous method input parameters are *Areahet*, *scale1*, and *scale2*. *Scale1* defines the fractal dimension (2 ≤ scale1 ≤ 3) and is then transormed to the Hurst exponent by H = 3—D. *Scale2* defines the variances of the random offset during the displacement procedure.

Figure 9 shows the diagonal through a grid of 100000 cells (*Area* = 1000) under three different magnifications generated by the superposition method. From the two diagonals of the grid *PatchOccupancy* computes the fractal dimension of the grid from a semivariogram (Hurst 1951, Burrough 1983, Hastings and Sugihara 1993) that uses the relation between variance per distance h and distance

$$Variance(h) \propto h^{2H}$$

H is the so-called Hurst exponent and the fractal dimension D is related to H by D = 2 - H.

Additionally, D is computed from the ruler length method based on the relation

$$Length(h) \propto h^{1-D}$$

AreaHet defines how many individuals are in the mean placed at each placing step and has to be a non-negative real number. An input of 0 results in a homogeneous grid.

Scale1 and *Scale2* are two positive parameters that define the structural properties of the grid. Setting *Scale1* to 1 defines a non-fractal grid.

RndHet defines whether at each step exactly N individuals are placed (input: *fix*) or whether this number is a normally distributed random variable with mean N and variance N.

NicheNo: Cells might not only differ in the number of individuals that can be placed (the carrying capacity) but also in the total number of

* Density Body weight (micoGr) Community A

388	39.9
229	7.1
85	164
37	548.1

* Density Body weight (micoGr) Community B

0.57	5.7
0.355	138.6
0.28	7.9
0.25	163
0.15	98.4
0.1	163
0.055	391.6

Table 2: An input file with data from two communities. The first column has to contain abundance data, the second can (but need not) contain species body weights. The file needs not to be formatted

Table 3: The output file *Distribution.txt* that shows the original community used or generated by the program.

Characteristics of whole community

Areal	SampA	maxSa	minSa	StepSa	Sampfac	Method	Species	maxD	minD	Dfluc	Flucmy	Flucvar	DivInd	Permut	Grid-Het			
20	10	12	4	4	1	dep	20	3	0.0001	0	0.5	0	sha	20	fix			
DWD	SWD	Class	SWDmy	SWDvar	DWDmy	DWDvar	Aggr	AreaHet	Scale1	Scale2	NicheNo	Range	Hab-Grad	RndHet	Model	Const =k	Const =X	TruncF
power	normal	6	3	1	0.9	0.1	1.5	2.8	1.5	2.8	3	0.5	10.000	rnd		0	0	0

No. of species per weight class

1	5	8	5	1	0

SpeNo	Weight	OrigInd	AssInd	Density boundary
10	6	7128	7128	19937
6	4	1246	1246	28717
20	0	0	0	0
3	2.8	17982	17982	39587
8	5	13997	13997	23492
11	6.5	12235	12235	18551
2	2.4	2889	2889	45478
18	14.4	1411	1411	9067
17	12.8	156	156	10081
1	2	24876	24876	53588
9	5.5	21509	21509	21561
19	16	7706	7706	8246
5	3.6	27230	27230	31573
12	7	16083	16083	17354
4	3.2	29007	29007	35104
16	11.2	8967	8967	11368
13	7.5	7725	7725	16309
15	9.6	5728	5728	13060
14	8	6426	6426	15389
7	4.5	17491	17491	25829

No. species	Ind. of Ass.	Total biomass	SD lb (dens)	RAD slope	RAD Inter	Diversity	Evenness	Alpha
19	229792	1215116	1.8364	-0.1256	-1.8606	2.6583	0.9028	1.61

species. *NicheNo* defines such an upper limit in species diversity. Again, a similar trigonometric function as above is used (again with ran being a linear random variable):

Maximum=*NicheNo**(sin(ran**x-coodinate*)+cos (ran**y-coordinate*)) +(ran+1)**NicheNo*.

However, we immediately see the problem with such an approach. *PatchOccupancy* places the species sequentially into the cells starting generally from the most abundant. If we define upper limits of species numbers the first and most abundant species will be placed without problems. However, more rare species will have more and more difficulties to find 'free' cells and in some cases it will be impossible to place them at all because all cells are already filled to their carrying capacity. Otherwise, starting with the least abundant species would at the end disturb the original relative abundance distribu-

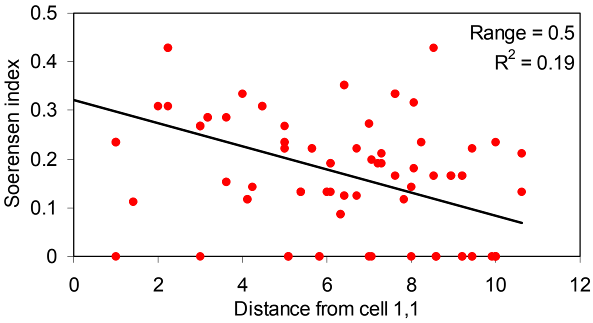


Figure 10: 100 species of a hypothetical assemblage were placed in a homogeneous grid with no habitat heterogeneity (*AreaHet* = 0, and *Aggreg* = 1). Defining Range = 0.5 results in a slight but significant dependence of the Soerensen index of similarity between two cells and the distance of these cells.

tion because many individuals of the more common species will not find cells. There is no simple way out of this dilemma. *PatchOccupancy* tries to minimize both effects by an initial randomising of the species sequence. In this case abundant and rare species have equal chances to be placed. Note, however, that the whole effect is only severe when *NicheNo* is set to low values (in relation to the species number of the community) and when mean densities per species were set to values well above 1.

Range: In real medium sized to larger regions (above the local scale) species differ in range size and adjacent places are generally more similar in species composition than more distant places. More technically speaking, any index of species similarity (like the Soerensen or the Jaccard index or the Bray-Curtis measure) should be a function of patch distance. To mimic such a pattern *PatchOccupancy* allows defining range sizes. The parameter *Range* defines how narrow species home range on the grid are. Placing probabilities per grid are multiplied with a distant dependent function of the form

$$prob = \left(1 - \frac{Range * Distance}{Area}\right)^{10}$$

where *Distance* denotes the distance from the centre of a species home range (defined by the coordinates of the first individuals being placed) and *Area* the total grid area. The input values of *Range* are from 0 (no effect) to 1 (a strong effect). This means for instance that for an input value of 0.9 the most abundant species will occupy about 5% of total area, a *Range* value of 0.1 spreads this species on about 3/4 of area. Fig. 10 shows how *Range* effects the species similarity of cells in dependence of cell distance.

DivInd defines the diversity and evenness statistics *PatchOccupancy* uses. You may choose three diversity indices with it's associated evenness measures:

a) Shannon – Wiener (*sha*):

$$H_{sha} = -\sum p_i \ln(p_i); E_{sha} = H_{sha} / \ln(S)$$

b) Simpson (*simp*):

$$H_{simp} = 1 / \sum p_i^2; E_{simp} = H_{simp} / S$$

c) Smith – Wilson (*evar*) (Smith and Wilson 1996):

$$E_{\mathrm{var}} = 1 - \frac{2}{\pi}\arctan\left(\frac{\sum_{i=1}^{S}\left(\ln(p_i) - \frac{\sum_{i=1}^{S}\ln(p_i)}{S}\right)^2}{S}\right);$$

$$H_{\mathrm{var}} = E_{\mathrm{var}} \times S$$

E_{var} is equivalent to the arctan transformed Gaussian width. For these three indices *PatchOccupancy* gives also standard deviations obtained from the respective values of each replicate. Additionally the program computes the α-diversity index (Magurran 1988, Rosenzweig 1995).

The last input parameter is *Permut*. If you compute species—area relations from a summation process of independent cells (the *Method dep* option; see Fig. 6) the outcome highly depends on the ordering of cells. To minimize this effect *PatchOccupancy* reshuffles the sampling order up to 20 times according to the setting in *Permut*. Colwell and Coddington (1994) showed that 20 such reshufflings are sufficient to get stable SARs.

4. Using real communities for sampling

PatchOccupancy is able to read extern

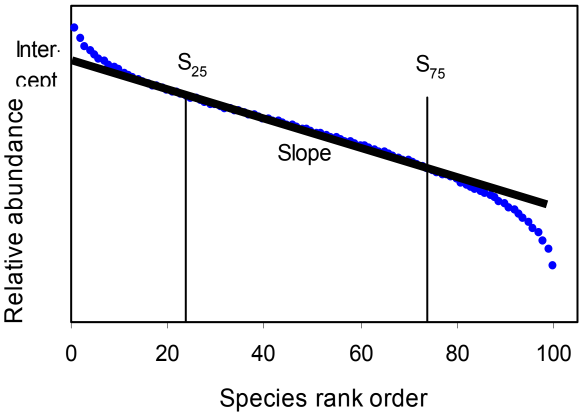


Figure 11: A relative abundance distribution (the blue data points stem from a Random fraction model of 100 species) can be characterized by a linear regression through the middle ranking 50% of species in a semi-logarithmic plot (see Ulrich 2001b, c). *PatchOccupancy* computes the slope and the intercept of this regression.

files in the text-format that contain data about abundances and body weights (optional) of real communities. It also reads the *Assemblage.txt* file generated by the program RAD (Ulrich 2002) that generates theoretical relative abundance distributions. *PatchOccupancy* asks at the beginning what type of file to be used. In the case of files other than *Assemblage.txt* the file has to contain a * as the first sign before the beginning of the data. Because *PatchOccupancy* can run via a batch file an unlimited number of real community data can be processed during one program run. **The data have to be sorted according to density!**

Table 1 shows an example of such a batchfile. Note that body weights are later printed in the fixed format Fn.2. It is therefore recommended to use an appropriate unit of measurement for body weights.

In Table 1 densities are given as numbers of individuals. However, *PatchOccupancy* does not know how density data are given. But, you can influence the way of data processing. If the first species in the list has a density equal to or above 1 *PatchOccupancy* assumes that the data are given as individuals. It then adjusts this density to the maximum defined by *maxD*. If the first species has a value below 1 *PatchOccupancy* assumes that densities are given as relative densities. In this case the program first adjusts the data in such a way that the least abundant species has exactly 1 individual. In a next step these

Table 4: The output file *Sample.txt* shows the total sample as defined by *maxSa*.

Characteristics of total sample

Areal	SampA	maxSa	minSa	StepSa	Sampfac	Method	Species	maxD	minD	Dfluc	Flucmy	Flucvar	DivInd	Permut	Grid-Het
20	10	12	4	4	1	dep	20	3	0.0001	0	0.5	0	sha	20	fix

DWD	SWD	Class	SWDmy	SWDvar	DWDmy	DWDvar	Aggr	AreaHet	Scale1	Scale2	NicheNo	Range	Hab-Grad	RndHet	Model	Const =k	Const =X	TruncF
power	normal	6	3	1	0.9	0.1	1.5	2.8	1.5	2.8	3	0.5	10.000	rnd		0	0	0

SpeNo	Weight	Ind	Density	StDev	Log2 Dens	Lloyd	Density boundary
8	5	5	0.41667	0.75920	-1.263	1.92	18.96
12	7	5	0.41667	1.38193	-1.263	9.6	14.04
15	9.6	6	0.50000	0.95743	-1	2.667	10.56
18	14.4	4	0.33333	0.84984	-1.585	4.5	7.32
16	11.2	0	0.00000	0.00000	0	0	9.12
5	3.6	0	0.00000	0.00000	0	0	25.56
6	4	10	0.83333	1.14261	-0.263	1.68	23.16
1	2	34	2.83333	2.03443	1.503	1.163	43.32
2	2.4	29	2.41667	2.81242	1.273	1.941	36.72
19	16	1	0.08333	0.27639	-3.585	0	6.60
13	7.5	0	0.00000	0.00000	0	0	13.20
9	5.5	6	0.50000	0.86603	-1	2	17.40
10	6	4	0.33333	1.10554	-1.585	9	16.08
11	6.5	0	0.00000	0.00000	0	0	15.00
4	3.2	1	0.08333	0.27639	-3.585	0	28.32
14	8	0	0.00000	0.00000	0	0	12.36
20	32	0	0.00000	0.00000	0	0	3.48
3	2.8	0	0.00000	0.00000	0	0	32.04
17	12.8	0	0.00000	0.00000	0	0	8.16
7	4.5	0	0.00000	0.00000	0	0	20.88

No. species	Ind. Sample	Total biomass	SD lb (dens)	RAD slope	RAD Inter	Diversity	Evenness	Alpha	Area sampl.	Unit area
11	105	429	1.5372	-0.1336	-2.1553	1.8991	0.792	3.1	12	1

modified abundances are again adjusted according to the settings in *maxD*. In the example of community A of Table 1 the first species has a density value D > 1. Given that you defined *maxD* as 10 individuals per cell and use a grid of 100 cells (*Area* = 10) total individual numbers per species S_i to be placed in the grid are computed from $S_i = D_i*(100*10/D_1)$. The first species gets therefore the individual number 1000, the second 590. If you want to use the individual numbers of the input file you have therefore to set *maxD* to $D_1/(Area*Area)$.

In community B of Tab. 2 D_1 is less than 1. In this case all data are first divided through the density of the least abundant species D_S. Total individual numbers per species are therefore again computed from $S_i = D_i*(100*10/D_1)$. In our example the first species gets again 1000 and the second species 623 individuals. The least abundant species gets 96

Table 5: *Analysis.txt* is an output file that gives basic features of the sampling process and computes additional statistics.

First row: Area samples with 1,2,...n sample units
Second row: Maximum number of niches per sample unit
Third row: No. species in exactly 1, 2...n sample units
Fourth row: Mean number of new species per sample unit
Fifth row: Number of species found only in that sample (endemics)
Sixth row: Cumulative number of new species per sample unit
Seventh row: No. species of assemblage, No. species/unit area, No. species sampled,
rel. species density (species per unit area /No. species in the assemblage),
Bootstrap summand, total number of samples, Species found in exactly one and in two sample units
Eigth row: SARS and EARS: SAR intercept, slope and variance explanation of the power function, and the exponential model, and the power function EAR
Ninth, tenth and eleventh row: Variances of cell properties (ninth) and surface length (tenth) with increment (ruler length) of factor 2 (of the two diagonals of the grid)
Twelfth row: Number of binary scaling classes, number of data points included in the regression, intercept of log(variance)-log(increment) plot, slope, fractal dimension,
intercept of log(surface length) - log(ruler length) plot, slope, fractal dimension

Areal	SampA	maxSa	minSa	StepSa	Sampfac	Method	Species	maxD	minD	Dfluc	Flucmy	Flucvar	DivInd	Permut	GridHet			
100	20	12	4	4	1	dep	20	10	1E-08	0	0.5	0	sha	20	fix			
DWD	SWD	Class	SWDmy	SWDvar	DWDmy	DWDvar	Aggr	Area-Het	Scale1	Scale2	NicheNo	Range	Hab-Grad	RndHet	Model	Const=k	Const=X	TruncF
power	normal	6	3	1	0.9	0	2	5	2	1.5	5	0.5	10.000	rnd		0	0	0
1	2	3	4	5	6	7	8	9	10	11	12							
5	10	13	17	22	28	38	40	44	52	62	65							
1	3	2	1															
4.3	1.5	0.2	1															
0.0	0.0	0.0	0.0															
4.3	5.8	6	7															
20	4.3	7	0.215	0.5	4	1	3											
4.372	0.331	0.956	4.337	1.81	0.953	0	0	0										
2	0	2	1	3	1	0	0											
4.5	2.3	1	0.4	0.4	0.2	0.2	0											
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0											
4.5	6.8	7.8	8.2	8.6	8.8	9	9											
20	4.5	9	0.225	0.7	8	2	0											
5.016	0.319		0.909	4.986	2.141	0.953	0	0	0									
1	0	0	2	1	2	1	0	1	1	0	0							
4.2	2.3	1	0.9	0.1	0.1	0.2	0.2	0.1	0.2	0	0							
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0							
4.2	6.4	7.4	8.3	8.4	8.5	8.6	8.8	8.8	9	9	9							
20	4.2	9	0.21	0.4	12	1	0											
5.049	0.269		0.841	5.074	1.789	0.897	0	0	0									
1	1	2	2	4	4	8	8											
0.3032	0.3032	1.0708	1.19711	3.49506	2.70127	28.3225	6.63819											
24.222	24.2223	22.096	22.7423	18.4393	22.0916	14.6799	22.0916											
7	8	0.3020	1.794	1.103	24.4674	-0.143	1.143											

individuals. If you want the least abundant species to have exactly 1 individual you would have to set *maxD* to 1/96 = 0.0104.

5. The output files

5.1. Distribution.txt

PatchOccupancy generates 6 output files: *Distribution.txt*, *Sample.txt*, *Cells.txt*, *Analysis.txt*, *Map.txt*, and *Grid.txt*.

Distribution.txt (Tab. 3) gives the original community used or generated by the program. This is the single output file after the *m*-option in 'What to do?'. Table 3 shows an example of this file.

The first five lines in this file show once again the input parameters. They contain four variables *Model*, *Const=K*, *Const=X*, and *TruncF* that refer to the case when the program uses a theoretical relative abundance distribution generated by the program RAD (Ulrich 2002).

The next line contains the number of species per $\log_2$ weight class. In this example 6 classes were defined, but for 20 species distributed according to a normal species—body weight distribution with mean = 3 and a standard deviation of 1 only 5 weight classes are possible.

Then, *PatchOccupancy* gives the assigned body weights per species (*Weight*), the total number of individuals per species before placing them into the grid (*OrigInd*), the total number of individuals per species after placing (*AssInd*), and the theoretical upper abundance limit per species as defined by the density—body weight distribution (*Density boundary*). *OrigInd* is nearly always lower then the density

Table 6: The optional output file *Cells.txt* gives the whole sample matrix, the coordinates of the sampled cells and the maximum number of niches per sample unit.

Areal	SampA	maxSa	minSa	StepSa	Sampfac	Method	Species	maxD	minD	Dfluc	Flucmy	Flucvar	DivInd	Permut	Grid-Het			
20	10	12	4	4	1	dep	20	3	1E-08	0	0.5	0	sha	20	fix			
DWD	SWD	Class	SWDmy	SWDvar	DWDmy	DWDvar	Aggr	AreaHet	Scale1	Scale2	NicheNo	Range	Hab-Grad	RndHet	Model	Const =k	Const =X	TruncF
power	normal	6	3	1	0.9	0.1	1.5	2.8	1.5	2.8	3	0.5	10.000	rnd		0	0	0
	Coordinates of Cells																	
	3	4	3	3	3	6	1	6	7	8	7	6						
	7	4	6	4	2	6	9	4	4	6	3	1						
	Maximum number of niches per sample unit																	
	3	5	4	4	5	4	1	5	5	4	5	6						
Species	Data, each row is the sampling of one of the					20 species												
8	0	2	0	0	2	0	0	0	0	1	0	0						
12	0	0	0	0	0	0	5	0	0	0	0	0						
15	0	2	0	3	0	0	0	1	0	0	0	0						
18	0	0	3	1	0	0	0	0	0	0	0	0						
16	0	0	0	0	0	0	0	0	0	0	0	0						
5	0	0	0	0	0	0	0	0	0	0	0	0						
6	2	0	1	0	1	0	0	1	0	1	0	4						
1	3	3	3	3	2	2	0	3	2	5	0	8						
2	0	6	0	8	0	4	0	3	0	2	0	6						
19	0	0	0	0	0	1	0	0	0	0	0	0						
13	0	0	0	0	0	0	0	0	0	0	0	0						
9	1	0	0	0	0	0	0	3	1	0	1	0						
10	0	0	0	0	0	0	0	0	4	0	0	0						
11	0	0	0	0	0	0	0	0	0	0	0	0						
4	0	0	0	0	0	0	0	0	1	0	0	0						
14	0	0	0	0	0	0	0	0	0	0	0	0						
20	0	0	0	0	0	0	0	0	0	0	0	0						
3	0	0	0	0	0	0	0	0	0	0	0	0						
17	0	0	0	0	0	0	0	0	0	0	0	0						
7	0	0	0	0	0	0	0	0	0	0	0	0						

boundary except in cases where the DWD model or the model used for density fluctuations had been a normal or a log-normal distribution.

Note that *OrigInd* and *AssInd* sometimes differ. That means not all individuals could be placed into the grid due to the limitations set by the grid property parameters. This occurs especially in cases when *NicheNo* was set to values were not all individuals could be placed.

Each species got originally a species number (*SpeNo*) with which it can be identified. *PatchOccupancy* does not sort the species list after placing and sampling.

The last output lines contain some statistics about the assemblage after placing. The first value is the number of species placed in the grid. This value is often smaller than the original species number of the community. Then comes the total number of individuals of the assemblage after placing. The product of individual number and species body weight gives the biomass. *PatchOccupancy* sums up these biomasses and prints the sum (*Total biomass*). In cases where no species biomasses are available a zero value is printed.

The next output value is the Gaussian weight of the assemblage after placing (SD lb(dens). The Gaussian weight is the standard deviation of $\log_2$ transformed densities of each species and a measure of community structure (see Sugihara 1980 and Tokeshi 1993). *RADslope* and *RADinter* give the slope and the intercept of the relative abundance distribution of the assemblage after placing (Ulrich 2001b, c, 2002)(Fig. 11).

The next output values are a diversity and the accompanying evenness index (see above). At last the alpha-diversity index is printed. It is computed from

$$S = \alpha \ln(1 + \frac{N}{\alpha})$$

with S being the total number of species and N the total individual number.

5.2. Sample.txt

In the case of a sample 3 other files are printed. *Sample.txt* gives the basic features of the sampled community. An example that corresponds to the community in Tab. 3 contains Tab. 4.

Again the first four output lines contain the program settings. Then, the program prints the species numbers (the same as in Tab. 3) and the body weights assigned by the program. *Ind* contains the total number of individuals found in all cells as defined by *maxSa*. *Density* gives the mean density per sample unit and *StDev* its standard deviation. The latter is only given in cases when *Method* is defined as *dep*. *Log2dens* gives the logarithm ($\log_2$) of *Density*. Lloyd computes the Lloyd index of aggregation. This index is defined by

$$L = \frac{StDev^2}{Density^2} - \frac{1}{Density} + 1$$

At last, the program gives the density boundary. This is the maximum number of individuals per sampling unit.

At the end of the output *PatchOccupancy* gives the number of species and the number of individuals sampled, the total biomass of the sample [=Σ (Density * Weight)], the Gaussian weight, the area sampled, and the unit of area (unit of sampling). The latter may be cells or squared multiples of cells (as defined by *Sampfac*). *Diversity*, *Evenness*, and *Alpha* refer to the diversity statistics chosen. RAD slope and RAD intercept again refer to the regression line through the middle ranking 50% of species as explained in Figure 11.

5.3. Analysis.txt

The third basic output file is *Analysis.txt* (Tab. 5). *Analysis.txt* prints basic features and statistics of the sampling process. Again, at the beginning *PatchOccupancy* prints the program settings. Then

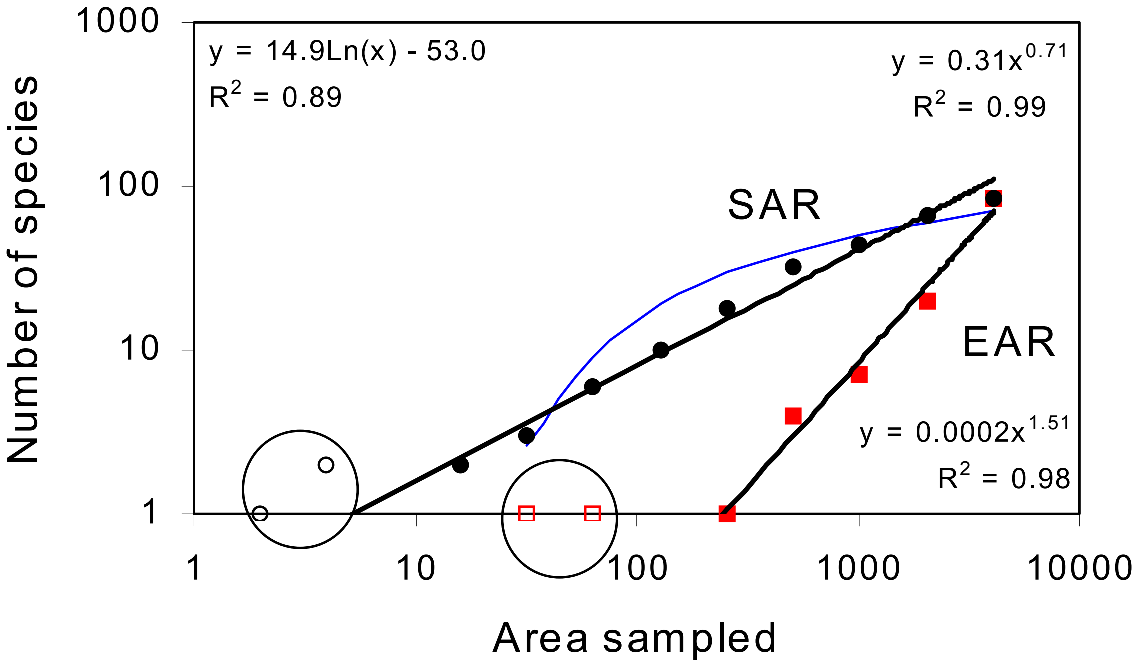


Fig. 12: An example how *PatchOccupancy* computes species - area and endemics - area relations. Logarithmic and power function models are fitted to the data set. The four data points marked with circles are not included in the computation.

the program prints the area sampled with n sample units. In the case of Tab. 5 the maximum sample size was defined as 12 and the sample unit is 1, that means a single cell. The maximum area is therefore 12 cells, the last value in the respective line. Below the sample area line *PatchOccupancy* prints the maximum number of niches in this area. In our case *NicheNo* was set to 53. We would expect 5*12=60 niches in an area of 12 cells. The actual value is 65. However, remember that the number of niches per sample unit is an approximately normally distributed random number around *NicheNo*. The values given in that line are therefore sometimes even higherthan the theoretical maximum.

The next lines print species numbers and some statistics of the sampling process. These are

1. The number of species found in exactly 1, 2, … n sampling units

2. The mean number of new species per sampling unit,

3. The number of species found only in that sample (the endemic species) (this option refers only to the sampling methods *oct* and *ind*), and

4. The total number of species found in the area given above. This latter line together with the area line, of course, define the species—area relationship.

The next line gives 8 basic values:

1. The number of species of the assemblage (assemblage refers always to the species actually placed into the grid),

2. The mean number of species found per sampling unit at this step,

Table 7: The output file *Grid.txt* prints the maximum number of individuals to be placed at each placing step for both diagonals (1. Diag from 1,1 to *area, area*; 2. Diag from 1,*area* to *area*,1)

Maximum number of individuals that can be placed into the cells of both grid diagonals at each placing step

Cell	1. Diag	2. Diag
1	4	4
2	3	5
3	2	7
4	3	6
5	2	4
6	5	5
7	7	2
8	4	1
9	1	3
10	3	4

3. The number of species found at this step,
4. The relative species density. This is the mean number of species found per unit of area divided through the total number of species of the assemblage,
5. The next value is the so called bootstrap summand, necessary to estimate species number via a bootstrap method. This value is defined as:

$$bootstrap = S_S + \sum_{i=1}^{S_S} (1 - p_i)^n$$

(Smith and van Belle 1984) with S_S being the number of species found in n samples and p_i its relative density.
6. The total number of samples per step
7. The number of species found in exactly 1 and 2 samples.

The next lines give the power function and the logarithmic type of species area—relationships as derived from each sample step. The first three values are the intercept, the slope, and the variance explanation of the power function model, the next three values are the intercept, the factor, and again the variance explanation of the logarithmic SAR. Additionally, *PatchOccupancy* computes a so called endemics—area relation (EAR)(Fig. 12). The last three values of this line are the intercept, the slope and the variance explanation of a power function EAR model. The input data are the values printed in the endemics line. (Nr. 3 above). All values refer to least square fits. Starting point for the computations is always the highest nonzero sample (Fig. 12).

Table 5 shows that the above values are given for each sample step, that means in our example after sampling of 4, 8, and 12 cells. By this way *PatchOccupancy* allows an easy estimate of the sample effectiveness and how good for instance species numbers, diversities or SAR patterns are approximated by different sample sizes.

At last, the program prints the distance h (measured as cells) of both grid diagonals, the associated variance in cell properties (to define a semivariogram), and the total surface length. The next line contains first the number of binary ruler length classes h and the number of ruler length classes used to computes the fractal dimension D. Because *PatchOccupancy* uses a superposition of trigonometric functions one grid might contain more than one scaling regions. Therefore, the program first looks for steps in the variance distribution and adjusts then the number of ruler length classes.

The next values of this output line are, the intercept of the ln (variance) - ln (h) regression, the associated slope and the fractal dimension. Then follow the intercept and the slope of the ln (surface length) - ln (h) regression and again the fractal dimension. In the case of Tab. 5 the grid has approximately the properties of a rescaled random variable and the fractal dimension is nearly 2. Note that sometimes computed fractal dimensions are outside the possible range. This happens when grid structures are chosen with either symmetrical features or several scaling regions. It is therefore recommended to check grid structure using the output file *Grid.txt* that gives the cell properties of both diagonals.

5.4. Cells.txt

Cells.txt is an optional file and gives the numbers if individuals per species found in each sample (Tab. 6).

Again *Cells.txt* contains the program set-

Table 8: The output file *Map.txt* prints the number of individuals placed into the cells of the total grid

Species number: 1 Centre of range: x 7. y: 8.

X-Coord.	Y-Coord.	Individuals
1	4	4
1	3	5
1	2	7
1	3	6
1	2	4
2	5	5
2	7	2
2	4	1
2	1	3
2	3	4

tings. If *Method* was defined as ‘dep’ it then gives the coordinates of the cells sampled. In the cases of *Method* = ‘seq’ and *Method* = ‘ind’ *Cells.txt* gives in this place the area sampled. The next line gives the maximum number of niches per area sampled.

Then, *PatchOccupancy* prints the species number and the number of individuals found in each sample.

5.5 Grid.txt

An additional optional file is *Grid.txt*. It gives the maximum number of individuals that can be placed (at each placing step) into the cells of both diagonals of the whole grid (Tab. 7).

5.6 Map.txt

The last optional output file is *Map.txt* (Tab. 8). *Map.txt* shows for a predefined species the whole distribution matrix in the grid together with the centre of home range (if defined).

6. Some problems with this version

The definition of large grids and larger species numbers may result in extraordinary large computing times. Define therefore these values not larger than necessary to solve a given problem. Care has to be taken that the initial program setting for the SWD and the DWD are realistic, Unrealistic settings result later in assemblage structures that have nothing in common with structures of real world communities. In a few cases unrealistic initial settings force the program into runtime errors or endless loops.

7. System requirements

PatchOccupancy is written in FORTRAN 95 and runs under Windows 9.x and XP. Computation abilities are only limited by the computer’s memory.

8. Citing *PatchOccupancy*

PatchOccupancy is freeware but nevertheless if you use RAD in scientific work you should cite *PatchOccupancy* as follows:

Ulrich W. 2002 - PatchOccupancy – a FORTRAN program for the study of species spatial distributions - www.uni.torun.pl/~ulrichw

8. Acknowledgements

The development of this program was supported by a grant of the Polish Science Committee (KBN, 3 F04F 03422).

10. References

Burrough P. A. 1983 - Multiscale sources of spatial variation in soil II. A non-Brownian fractal model and its application in soil survey - J. Soil Sci. 34: 599-620.

Colwell R. K., Coddington J. A. 1994 - Estimating terrestrial biodiversity through extrapolation - Phil. Trans. R. Soc. Lond. B: 345: 101-118.

Currie D. J. 1993 - What shape is the relationship between body mass and population density - Oikos 66: 353.

Currie D. J., Fritz J. T. 1993 - Global patterns of animal abundance and species energy use - Oikos 67: 56-68.

Gotelli N. J., Graves G. R. 1996 - Null Models in Ecology - Smith. Inst. Press, Washington D.C.

Hanski I. 1991 - Single-species metapopulation dynamics: concepts, models and observations - Biol. J. Linnean Soc. 42: 17-38.

Hanski I. 1999 - Metapopulation ecology - Oxford (Univ. Press).

Hanski I., Kouki J., Halkka A. 1993 - Three explanations of the positive relationship between distribution and abundance of species (In: Species Diversity in Ecological Communities, Eds.: R. E. Ricklefs, D. Schluter) - Univ. Press, Chicago, pp. 108-116.

Hanski I., Gilpin M. E. (Eds.) 1997 - Metapopulation Biology - London Academic Press, 512 pp.

Hastings H. M., Sugihara G. 1993 - Fractals - A User's Guide for the Natural Sciences - Oxford Univ. Press.

Hurst H. E. 1951 - Long-term storage capacity of reservoirs - Trans. Am. Soc. Civil Eng. 116: 770-808.

Leitner W. A., Rosenzweig M. L. 1997 - Nested species-area curves and stochastic sampling: a new theory - Oikos 79: 503-512.

Magurran A. E. 1988 - Ecological diversity and its measurement - Princeton (Univ. Press), pp. 1-179.

May R. M. 1975 - Patterns of species abundance and diversity (in: Ecology and evolution of communities, Eds.: M. L. Cody, J. M. Diamond) - Belknap (Cambridge), pp. 81-120.

Novotny V., Kindlmann P. 1996 - Distribution of body sizes in arthropod taxa and communities - Oikos 75: 75-82.

Smith E. P., van Belle G. 1984 - Nonparametric estimation of species richness - Biometrics 40: 119-129.

Smith B., Wilson, J. B. 1996 - A consumer's guide to evenness indices - Oikos 76: 70-82.

Sugihara G. 1980 - Minimal community structure: an explanation of species abundance patterns - Am. Nat. 116: 770-787.

Tokeshi M. 1993 – Species abundance patterns and community structure – Adv. Ecol. Res. 24: 111-186.

Tokeshi M. 1996 - Power fraction: a new explanation of relative abundance patterns in species-rich assemblages - Oikos 75: 543-550.

Wilson W. B., Gitay H., Steel J. B., King W. McG. 1998 - Relative abundance distributions in plant communities: effects of species richness and spatial scale - J. Veg. Sci. 9: 213-220.

Ulrich W. 1999a - Estimating species numbers by extrapolation I: Comparing the performance of various estimators using large model assemblages - Pol. J. Ecol. 47: 271-291.

Ulrich W. 1999b - Estimating species numbers by extrapolation II: determining the minimum sample size to obtain a certain fraction of species in a community - Pol. J. Ecol. 47: 293-305.

Ulrich W. 1999c - Abundance, biomass and density boundaries in the Hymenoptera: analysis of the abundance - body size relationship and differences between forest and open landscape habitats - Polish J. Ecol 47: 73-86.

Ulrich W. 2000a - On species - area relationships I: fit of power function and exponential model - Pol. J. Ecol. 48: 3-20.

Ulrich W. 2000b - On species - area relationships II: slope and factor values of power function and exponential model - Pol. J. Ecol. 48: 21-35.

Ulrich W. 2000c - On species - area relationships III: the intercept of the power function and the exponential model - Pol. J. Ecol. 48: 37-48.

Ulrich W. 2000d - Species area relations generated by theoretical relative abundance distributions: parameter values, model fit and relation to species saturation studies - Pol. J. Ecol. 48: 67-78.

Ulrich W. 2001a - Relative abundance distributions of species: The need to have a new look at them - Pol. J. Ecol.: 393-407.

Ulrich W. 2001b - Models of relative abundance distributions I: model fitting by stochastic models - Pol. J. Ecol. 49: 145-157.

Ulrich W. 2001c - Models of relative abundance distributions II: diversity and evenness -statistics - Pol. J. Ecol. 49: 159-175.

Ulrich W. 2002 - RAD – a FORTRAN program for the study of relative abundance distributions – http//www.

Wright D. H. 1991 - Correlations between incidence and abundance are expected by chance - J. Biogeogr. 18: 463-466.