1. SIMBAA MANUAL

SIMBAA

SImulation Model of Benthic Antarctic Assemblages

Short

SIMBAA MANUAL

-Version 2-

(This manual covers SIMBAA GUI 1.6b and SIMBAA core 1.20.G or higher)

Michael Potthoff, AWI Bremerhaven

Department:	Benthic Ecosystems | Comparative Ecosystem Research
Email:	mpotthoff@awi-bremerhaven.de
Address:	Columbusstrasse
Town:	D-27568 Bremerhaven (Building D-2100)
Telephone:	+49(471)4831-1858
Fax:	+49(471)4831-1149

Last update: 13.06.2005

1. SIMBAA MANUAL 1

1.1 INTRODUCTION 3
1.1.1 Time in the model 3
1.1.2 The environment (a-biotic parameters) 3
1.1.3 The space 3
1.1.4 Disturbance events 5
1.1.5 Disturbance severity 7
1.1.6 The flow 7
1.2 THE BIOLOGY MODEL OF SIMBAA (BIOTIC PARAMETER) 9
1.2.1 List of species traits available in SIMBAA: 9
1.2.2 General information 10
1.2.3 Reproductive Traits 10
1.3 DISPERSAL TRAITS 12
1.4 THE DISPERSAL IN SIMBAA 13
1.4.1 Overview of dispersal kernels available in SIMBAA: 15
1.4.2 Lifespan / mortality 19
1.4.3 Other species traits 21
1.4.4 Growth mode 23
1.5 THE SIMBAA GRAPHICAL USER INTERFACE (SIMBAA-GUI) 24
1.5.1 The main window 25
1.5.2 The disturbance editor 26
1.5.3 The species editor 27
1.5.4 The landscape editor 29
1.5.5 The flow editor 32
1.5.6 The visualise tool window 33
1.5.7 The virtual ROV 36
1.5.8 Cluster analysis tool 38
1.5.9 Rank analysis tool 41
1.5.10 Succession analysis tool, succession state definition 42
1.5.11 Age structure analysis tool 44
1.6 ADDITIONAL SIMULATION PARAMETER DIALOG 45
1.6.1 Simulation stop conditions: 47
1.7 DIVERSITY MEASUREMENTS AVAILABLE IN SIMBAA 51
1.7.1 α-Diversity 51
1.7.2 β-Diversity 52
1.8 THE M-INDEX 53
1.8.1 M-Index, an example: 54
1.9 GENERAL SIMBAA TIPS 59
1.9.1 Appendix 61
1.9.2 SGF-Header 62
1.9.3 State definitions 64
1.9.4 Flow Grid definition 65
1.9.5 Disturbance definitions 66
1.9.6 Species definitions 67
1.9.7 Simulation grid definition 68
1.10 ACKNOWLEDGEMENT, EXTERNAL CODE 70

1.1 Introduction

SIMBAA is a spatial explicit, individual based simulation model. Its main purpose is the analysis of disturbance events on an assemblage of marine sedentary organisms. It was programmed at the Alfred-Wegener-Institute for Polar and Marine Research by Michael Potthoff during his PhD thesis 2003-2006 on the influence of iceberg scouring on the benthos of the Weddell Sea. Therefore some special terminology on marine and polar science may be used throughout this manual, however the mechanisms and features implemented in SIMBAA are quite general and can be applied to other fields of interest as well.

As every other model or simulation SIMBAA makes some assumptions and generalisations. The purpose of this chapter is to introduce these basic mechanisms and generalisations.

1.1.1 Time in the model

Time in SIMBAA proceeds in discrete steps. One time step is equivalent to one simulation cycle, e.g. every individual in the simulation can (according to its state) reproduce, disperse, die (so it can complete its own life cycle ones), disturbance events may happen and free space may be colonised by recruits. The best equivalent of one simulation time step is one season in real life. Internally all information is handled in a way that the update, e.g. the change of the states of all individuals, occurs synchronously.

1.1.2 The environment (a-biotic parameters)

1.1.3 The space

As SIMBAA is a spatial explicit simulation, every individual occupies a certain position in space. Dispersal and competition is dependent on the particular properties of that position in space. However, SIMBAA does not use continuous space. Instead SIMBAA is based on a regular grid of small discrete areas, thus it is a grid-based simulation. The illustration shows the principle orientation of the grid. The grid origin is the lower left corner. Directions are analogue to common maps, 0° (north) is to the top, 90° (east) to the right, 180° (south) towards the bottom and 270° (west) to the left. The smallest spatial resolution is one single grid cell. The equivalent space in

nature depends on other model properties and can be a single square meter, 10th of square meters or even more. It is mandatory to have in mind that one grid cell is the basic spatial unit as all other spatial information (e.g. dispersal distances) are measured in units of grid cells!
The user can determine the grid dimensions, how many individuals (max 100) may life inside of one grid cell and if every cell of the simulation grid can hold the same amount of individuals or not. All Individuals of a cell are placed within a 10 x 10 sub-grid. This information is only used when displaying the individuals and has no other meaning. However, this limits the amount of individuals per cell to 100 individuals.
Each cell has 8 boolean properties S_1-S_8 ("Yes-No" or "enabled-disabled"), representing environmental e.g. sediment conditions. These can influence individuals living in this cell. Individuals in the cell may alter the properties as well as disturbance events do. By default S_1 is enabled. See species traits for more details.

By default the world represented in SIMBAA is a torus- or doughnut-scenario. This means that any object leaving the grid on one side re-enters the simulation on the opposite border. Thus border effects are avoided e.g. no larvae can be lost by "falling" off the grid. When periodic boundary conditions are enabled SIMBAA is a closed system. However, the periodic boundary conditions can be switched off. In this case SIMBAA is an open system and the simulation borders absorb any object leaving the simulation area. Figure 1 illustrates the spatial grid properties.

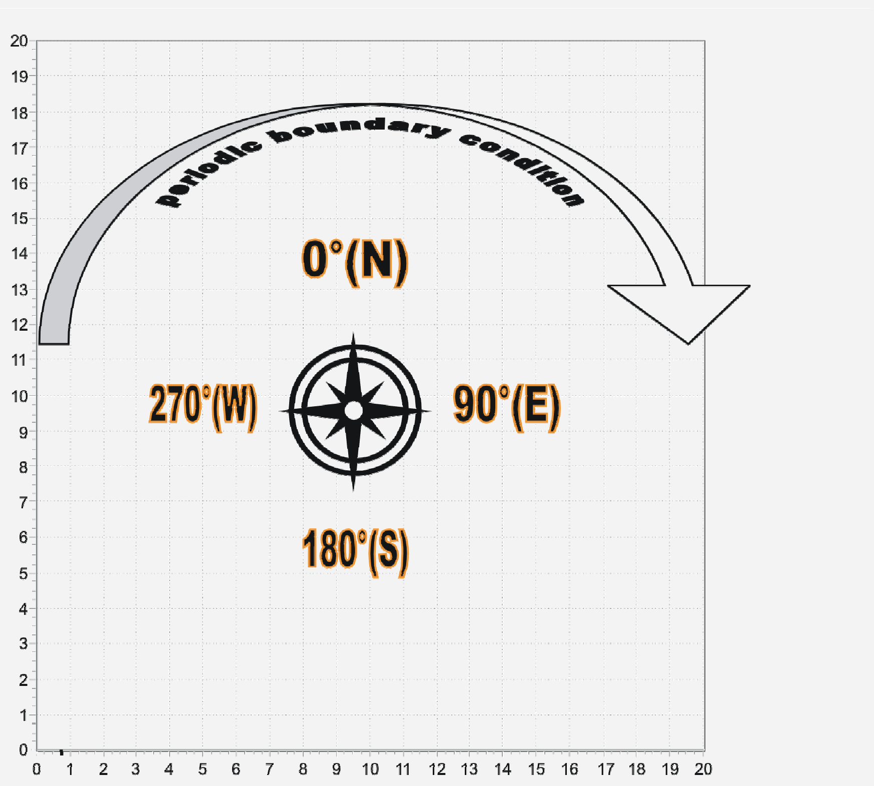


Figure 1, Orientation of the simulation grid in SIMBAA

1.1.4 Disturbance events

In a SIMBAA-simulation disturbance events can be defined. Each disturbance event is characterised by its spatial information, intensity and occurrence probability. Disturbance events may be restricted to start in a certain proportion of the simulation grid. This region is always rectangular and orthogonal to the grid axis. Disturbances may be clipped by this area, what means that they do not affect any space outside the defined region (see Figure 2).

Disturbances are always rectangular with a defined length and width and aligned in a certain direction. All these spatial information may deviate within a given, normally distributed region around the average values (see illustration below).

Disturbance probability is independent on prior realisations and can be given as probability. However, it is mainly expressed in terms of the **rotation period**. This is the time in which the whole simulation grid is (statistically) disturbed once. The rotation period can be computed by Formula 1:

$$rotation\ period\left[time\ steps\right]=\frac{simulation\ grid\ size\left[x*y\right]}{disturbance\ size\left[x*y\right]*disturbance\ probability\left[\frac{1}{time\ steps}\right]}$$

Formula 1, computation of the rotation period

SIMBAA allows a disturbance event to consist of several sub-events. Then several disturbance events of the same size and intensity happen at the same time. Thus a disturbance with 2 sub-events means that these two disturbances occur always together, although their start points are independently chosen. However, as they are sub-events, the disturbance probability and rotation period are based on their co-occurrence. This gives the possibility to create temporal and spatial correlated disturbance.

Disturbance area definition and clipping

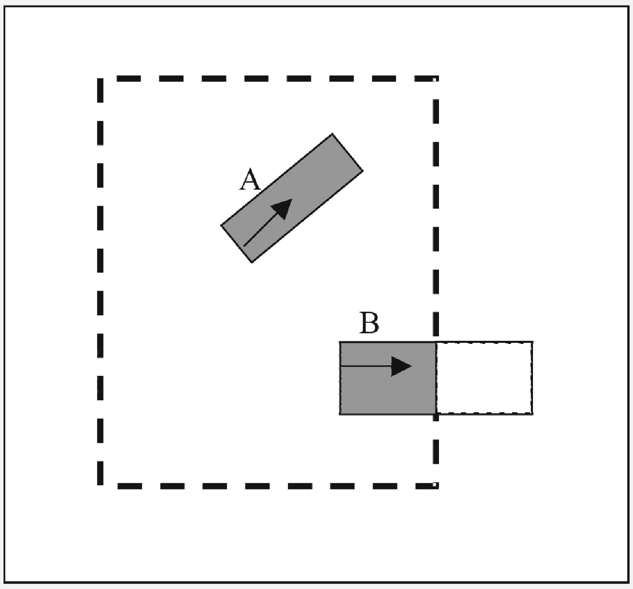


Figure 2, This is an example for disturbance definitions. The white area represents the complete simulation area. Disturbance A is completely in the disturbance area, defined by the dashed line. Disturbance B starts in this area (indicated by the arrow). However, it extends partially outside the disturbance area. If clipping is enabled, the shaded part would be omitted and B would be restricted to the defined area.

1.1.5 Disturbance severity

If an event occurs, the proportion of individuals removed from the disturbed area is analogue to the disturbance severity, e.g. 100% means all individuals are removed whereas 50% means, that any individual has a 50% chance to be removed.

1.1.6 The flow

As SIMBAA is used to model marine systems it contains a hydrographical sub-model of the flow to model larval dispersion. However the hydrographical sub-model is very simplified. The user can define how many flow cells lie on top of the entire simulation. If there is only one cell, its flow properties define the flow of the entire simulation grid. If there are more flow cells, each simulation grid cell is projected to the flow cell on top of it. If you use the same number of flow and grid cells you can define the flow properties for each grid cell individually. It is not advisable to use more flow cell than grid cells on the simulation. The flow property of a cell is based on the flow speed and flow direction (heading and deviation). The hydrographical sub-model is static in time. Therefore this information must be defined before a simulation run and cannot be changed while the simulation is running. However, each time step the flow is sampled a deviation of the given order around its average heading is computed. Thus the flow is normally distributed around the given average heading and standard deviation. Figure 3 illustrates the general properties:

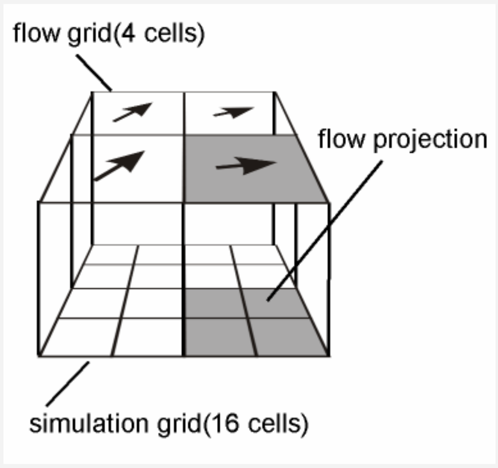


Figure 3, Schema of the hydrographical sub-model of SIMBAA. Arrows represent the direction and strength of flow. In this example the flow grid consists of 4 cells (top), the simulation grid of 16 cells (bottom). Thus each flow cell is projected to a cluster 4 simulation grid cells (e.g. shaded in the grey).

1.2 The biology model of SIMBAA (biotic parameter)

SIMBAA is an individual based simulation. This means that the smallest biological information is one individual. However, the user can see an individual as super-individual, representing more than one individual organism in reality (e.g. a colony). Each individual in the simulation belongs to one of several species. Currently up to 50 different species can be defined. All individuals of a particular species share the same species traits. In particular SIMBAA contains the following life-history traits:

1.2.1 List of species traits available in SIMBAA:

<u>**General**</u>	
Species name	Name listed in the graphs and data files
Guild	Guild of the species
POV output	number of species macro used by POV
Colour	Colour used for screen drawing
<u>**Reproductive traits**</u>	
Fecundity	number of potential recruits (see below!)
First reproduction	age of maternity [simulation time steps]
Reproduction interval	[simulation time steps]
Reproduction synchrony	
<u>**Dispersal traits**</u>	
Dispersal distance	[grid units]
Dispersal patch size	[grid units]
External drift	(migration possibility)
<u>**Lifetime traits**</u>	
Normal lifespan →mortality	[simulation time steps]
Maximal lifespan	[simulation time steps]
<u>**Other**</u>	
Growth on/ changes substrate	
Growth mode	

The following sections describe these traits in some more details:

1.2.2 General information

1.1.1.1 Species Name

The species name is used to identify the species to which an individual belongs

1.1.1.2 Guild

A species can belong to one of four guilds. The possible guilds are: R0, R1, R2 and UD. These guild names are arbitrary. This association is not necessary and can be undefined (guild "*"). The guild information can be used to define succession states and to assign a succession state to a grid cell.

1.1.1.3 POV-Output

This is used when a snapshot of the simulation is saved as POV[1]-file for 3D output.

1.1.1.4 Colour

This is the colour used for graphical representation of an individual. Although it is not necessary, SIMBAA tries to assign each species a unique colour based on its position in the species list.

1.2.3 Reproductive Traits

SIMBAA does not include detailed information about an individual's sex. It is assumed that all individuals are capable of reproduction. Thus species in SIMBAA reproduce either vegetative or by parthenogenesis. In case of modelling a species with different sex, individuals represented by SIMBAA must be seen as the reproductive active sex (most commonly female) and that fertilisation is not limited.

[1] POV (Persistence Of View) is a freeware raytracer. Raytracer can render photo realistic pictures of scenes. SIMBAA is capable to write a simulation snapshot in the POV scene description language (SDL). These scene files can be used to render a realistic 3D view of the simulation. However, the scene files may require extensive manual refinement, so the user needs good knowledge of the POV SDL and experience in this field.

1.1.1.5 Fecundity

This is a fractional number, roughly representing the per capita number of possible recruits rather than the real fecundity. SIMBAA does not model real larvae trajectories nor does it include larval mortality. Thus this species trait represents the number of possible recruits taking part in the competition for space after they have completed their dispersal phase and have become competent for recruitment.

In nature the number of produced larvae may be enormous, but only a fraction of these become competent and only an even smaller fraction of these will really recruit. This last phase, the recruitment, modelled in SIMBAA and in some more detail explained when the lottery competition is explained.

1.1.1.6 First reproduction

This is the time period in simulation time steps an individual needs to become maternal after recruitment.

1.1.1.7 Reproduction interval

This is the time period in simulation time steps between two consecutive larvae releases of an individual. This is normally based on the age of an individual. SIMBAA can be initialised in a special way thus that all individuals of a particular species are in the same reproduction phase. Thus it is possible to create a reproductive separation in time for different species. It is also possible to force a synchronisation on model time. Then the reproduction interval is triggered when the simulation time is dividable by the reproduction interval.

1.1.1.8 Reproduction synchrony (experimental feature)

This trait is only of interest when the species has a reproduction interval different from 1. In this case all individuals do not reproduce in each time step. Normally, the reproduction of a single individual is determined by its life cycle, i.e. the time since its last reproduction. When species are reproductively synchronised, many (or most) individuals reproduce at the same time. The reproduction synchrony gives quality of the synchronisation as probability of an individual's reproduction cycle to be in phase with the whole population. 100% means that an individual will only reproduce when the reproduction interval is valid, whereas 90% means that any individual has a 10% chance to reproduce even when it is not in the reproduction cycle.

1.3 **Dispersal traits**

1.1.1.9 Dispersal Distance

This is the distance [in grid cells] all larvae of a specie originating from a specific grid cell travel together as a swarm. If one does not want the larvae to be distributed as a swarm, set this number to zero. This distance is modified (multiplied) by the flow speed of the originating cell.

1.1.1.10 Dispersal patch size

This is the diameter [in grid cells] of the area where the larvae of a swarm “reach” then simulation grid after dispersal and compete for space.

1.1.1.11 External drift

This is a switch that defines if a species has the potential to migrate from outside the grid. If enabled, you can globally define the probability of such a migration event and the maximal number of larvae migrating.

1.4 The dispersal in SIMBAA

SIMBAA does not follow each released larvae and does not simulate larval mortality. Instead it uses a simplified approach. It is assumed that all larvae of a species released in a specific grid cell become dispersed together as a swarm. As a time step in SIMBAA roughly represents a whole season and real larval dispersal may be completed within a shorter time, just the characteristic distance of this first dispersal phase is given as dispersal distance. Please note that SIMBAA only models simplified larval trajectories. Dispersal occurs always along a strait vector or line. The length and orientation of this vector is depended on the flow properties of the birth cell and the selected dispersal kernel (see below).

After the first dispersal phase it is assumed that the whole swarm becomes competent and sinks to the sea floor. The centre of this patch where the larvae land is defined by the dispersal distance. The diameter of the area is defined by the dispersal patch size. Figure 4 illustrates the principle dispersal mechanism and Formula 2 shows a pseudo-code for the algorithm in SIMBAA.

Dispersal mechanism used in SIMBAA

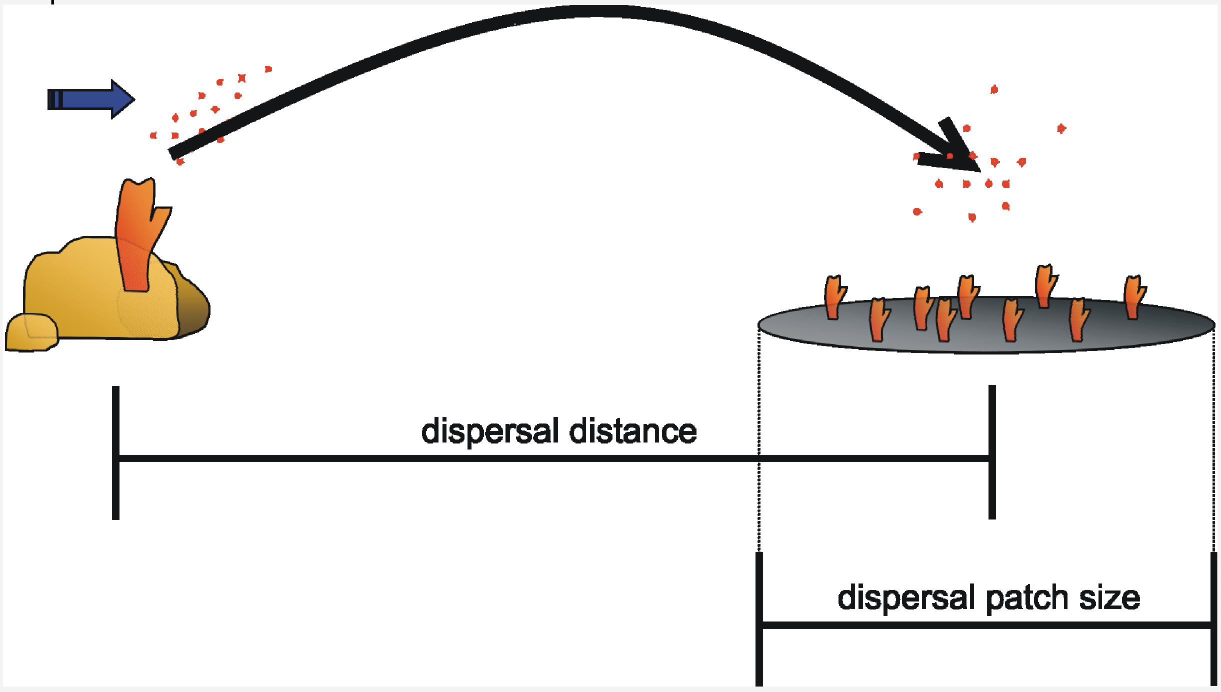
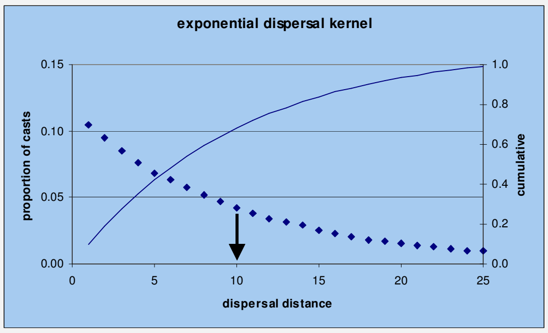
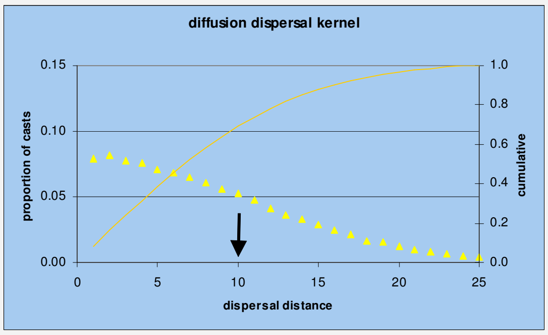
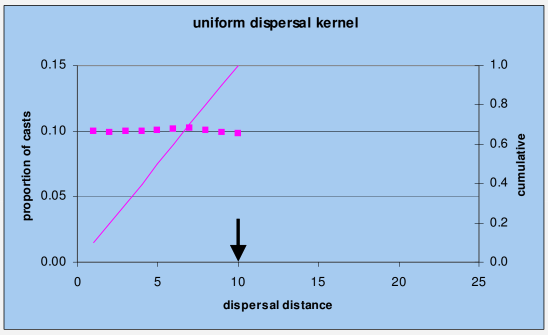


Figure 4, sketch of the dispersal mechanism used in SIMBAA. The mature individual on the left releases the larvae. A current (blue arrow) disperses the swarm over the “dispersal distance” to a new habitat on the right. Then the larvae swarm settles within a “dispersal patch size” area. Both dispersal distance and patch size are species-specific traits.

To compute the realised dispersal distance and the distribution within the dispersal patch, SIMBAA offers three different dispersal kernels: **exponential**, **diffusion** and **uniform**. These are descript in detail below. It is possible to select the kernels for the dispersal distance independently from the kernel for the patch size. However, these selections are global and it is (currently) not possible to select the kernels for each species independently.

The dispersal results in a local larvae pool for each single simulation grid cell. All larvae reaching a cell are colleted in this pool and compete for free space. Some recruit successful, however larvae that do not recruit die at the end of a time step. Thus no larvae live longer than one time step. For a detailed description of the recruitment process see the section lottery competition below.

1.4.1 Overview of dispersal kernels available in SIMBAA:

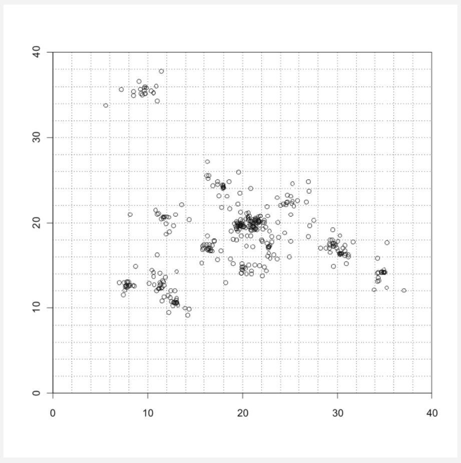
Dispersal Kernel	Description
	**<u>Exponential dispersal kernel</u>** Formula: `−ln(rnd) * d` Roughly ~70% of all casts are below d. Some extreme distances may occur, but rarely.
	**<u>Diffusion kernel</u>** Formula: `norm_rnd * d` Similar, ~70% of all casts have shorter distances and extreme values are possible but less likely than with an exponential kernel.
	**<u>Uniform kernel</u>** Formula: `rnd * d` All distances are uniform distributed between 0 and d. No values higher than d can occur.
The examples assume a dispersal distance=10. Dots represent the realized dispersal distance, lines the cumulative distribution. (10000 cast each).	d= dispersal distance rnd= uniform random number between 0 and 1 norm_rnd = normal distributed random number (μ=0, φ=1)

Dispersal Algorithm of SIMBAA

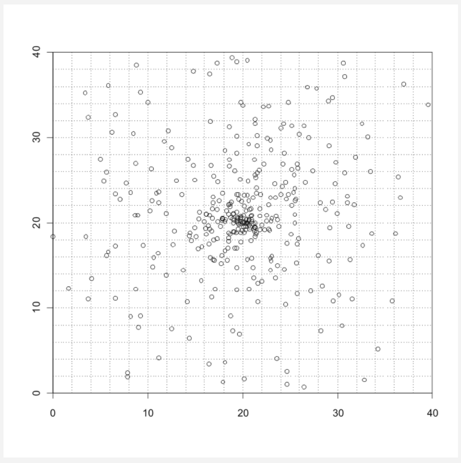
```
• count larvae of a species released from a position (X)
• determine flow speed and flow direction (s, α ) of X
• compute centre of dispersal patch (CODP) according to
  the dispersal distance (d), dispersal kernel ($f_{kernel}$),
  flow speed and flow direction

  →(CODP = X + $f_{kernel}$(s*d,α))

• distribute each larvae around CODP according to the
  patch kernel ($f_{patch}$) and patch size (ps)

  →(larva position=COPD + $f_{patch}$(s*ps,α))
```

Formula 2, Pseudo-code for the dispersal mechanism of SIMBAA

Example of a dispersal pattern generated by SIMBAA



Clumped dispersal



Isotropic dispersal

Figure 5, Both pictures show the final larvae pattern resulting from repetitively (10x) distributing same number of larvae (10) from the centre of the pictures. An exponential kernel was used both for dispersal distance and patch size. The flow speed was 1 and no specific flow direction was given (e.g. direction = any angle, deviation ±360°). The difference between both pictures was that in the right picture the dispersal distance was set to zero, so no larvae where distributed as swarm. Patch size was set to 10. Instead in the left picture patch size was set to 1 and dispersal distance to 10. This resulted in the patch distributed larvae clusters whereas in the first case no clusters apart from the origin can be observed.

1.1.1.12 Migration, external drift

SIMBAA offers the possibility of an external larvae pool independent on the actual simulation situation. From this outside pool a migration or external drift of larvae can occur. As the external pool is independent on the simulation it may contain larvae of species that have actually become extinct in the simulation.

The probability of such a migration event is globally defined. Each time step and for each cell it is individually checked, if such an event occurs. If so, the number of larvae is determined (draw from a uniform random distribution between 1 and a user defined max) and the according number of larvae is added to local larvae pool of that cell. The species of these larvae are randomly chosen out of the species that are capable of external drift.

1.1.1.13 Lottery competition

In SIMBAA competition and interaction between individuals occurs only in the settlement phase. Once established, individuals do not interact with each other. SIMBAA does not include a detailed competition module. It uses the simplest competition model available: lottery competition. This means that all larvae in a local larvae pool of each cell compete for available space in that cell. If there is space, either by the death of established individuals or due to a disturbance event, one larva out of the local pool is randomly selected and allowed to recruit. All larvae in the pool have the same chance to win. The established larva is removed from the pool. If there is still free place this procedure is repeated until either no free space is left or the local larvae pool is depleted. At the end of a time step, all local larvae pools are cleared. Thus no larvae are carried over into the next time step.

1.4.2 Lifespan / mortality

SIMBAA uses a special approach to determine the mortality of an individual. The user can define a “normal life span”. From recruitment at age 0 until this time an individual has a fixed mortality. The exact mortality can be given an absolute number. However, SIMBAA offers the possibility to simply define what proportion of a population (survival rate) shall reach this age and then computes the **instaneous mortality** according to Formula 3:

$$instaneous\ mortality = \frac{-\ln(survival\ rate)}{normal\ life\ span}$$

Formula 3, computation of instaneous mortality

If an individual becomes older than the defined “normal life span”, is mortality rises linear until it reaches “1” with the age “maximal life span”. Figure 6 illustrates the mortality of an individual during its life:

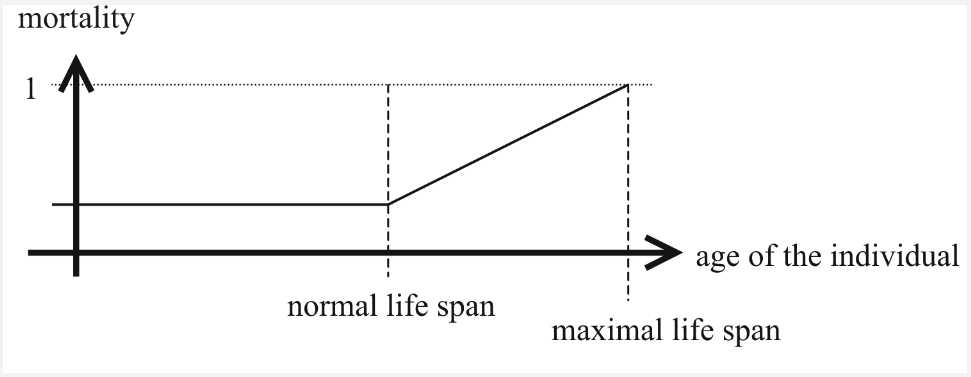
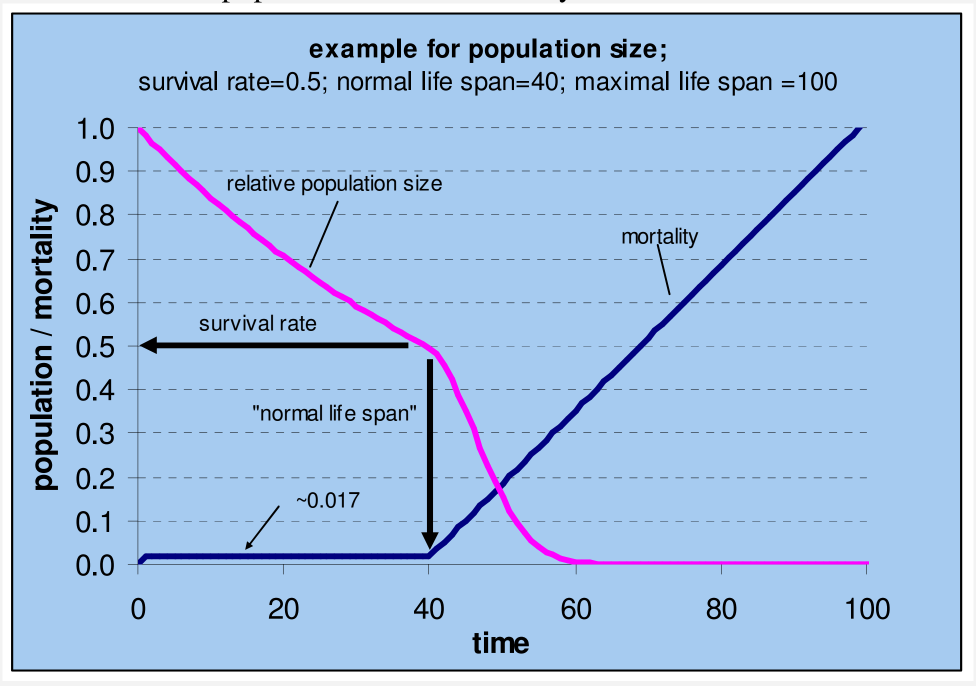


Figure 6, mortality computation in SIMBAA

NOTE:

When *selecting survival rate = 0.5* the “normal life span” is equal to the average life span!

Example calculation of population size & mortality:



Survival rate	= 0.5
Normal life span	= 40
Maximal life span	= 100
Mortality	= -ln(0.5) / 40
	~ 0.0173

While an individual is younger than 40 time steps its mortality is constant (~0.0173). Between an age of 40 to 100 time steps, the mortality raises linear with a rate of (1-0.0173)/(100-40)~ 0.0164 per time step. However, rarely any individual becomes older than 60 time steps.

1.4.3 Other species traits

1.1.1.14 Growth on/ changes substrate

This mechanism offers a possibility to mimic substrate specific features. These features are implemented as simple binary (“enabled-disabled”) switches. In total 8 different substrate switches S_1-S_8 are available. Each species can have special demands on the substrate conditions (“growth on”-conditions). When an individual of a species becomes older than its “normal life span”, it alters the substrate state according to a defined rule (“changes to”-rules).

! **NOTE:**
The “changes to” rule is technically implemented as a XOR-operation! The following examples assume that □=disabled and ■=enabled. !

Rule	Sediment condition	“Changes to”	→Resulting state
A	□	□	□
B	□	■	■
C	■	□	■
D	■	■	□

A disturbance resets all substrate switches to the first switch (S_1=enabled). By default, all species can life on S_1. If more than one species lives inside a cell, all possible interactions are summed up and work together:

“changes to”	S_8	S_7	S_6	S_5	S_4	S_3	S_2	S_1
Species A	□	□	□	□	□	□	□	■
Species B	□	□	□	□	□	□	□	■
Species C	□	□	□	□	□	■	■	□
Result	□	□	□	□	□	■	■	■

Some examples:

<u>Example 1</u>

	S_8	S_7	S_6	S_5	S_4	S_3	S_2	S_1
growth on	□	□	□	□	□	□	□	■
changes to	□	□	□	□	□	□	□	□

This is the default. A species with this schema will grow on a substrate S_1 and does not change it. As (by default) all cells have substrate state S_1 enabled and disturbances reset the state to S1 this is equivalent to a situation where no sediment properties are relevant.

<u>Example 2</u>

	S_8	S_7	S_6	S_5	S_4	S_3	S_2	S_1
growth on	□	□	□	□	□	□	□	■
changes to	□	□	□	□	□	□	■	■

In this example the species will also grow on S1. However, if it becomes older than its “normal life span” it will alter the switches S_1 and S_2. As S_1 is enabled (probability, as it is necessary for the recruitment of the species), it will turn off the S_1-state (rule D). When S_2 was disabled it will also turn on S_2 (rule B). In this example the species is likely to prepare the sediment conditions for S_2 and, when older than its normal life span, hinders its own species to recruit at this particular place.

1.4.4 Growth mode

The growth mode determines how the age is translated into an individual's size. This information is only used for visualisation purpose and does not influence the simulation. The user can select two different growth modes:

1.1.1.15 Linear growth: (default)

This is the default growth mode. The size is linear to the age, respectively to the "normal life span" when it becomes "max size". If an individual becomes older than "normal life span" the size does not grow further.

1.1.1.16 Exponential growth:

The size growth is exponential with age. When selecting this growth mode, the user must give a growth factor and a maximal growth. Formula 4 is used to compute the size:

$$size = \max size * (1 - \exp\left[\frac{-\,growth\ factor}{\left(\dfrac{100 * age}{\max life\,span}\right)}\right])$$

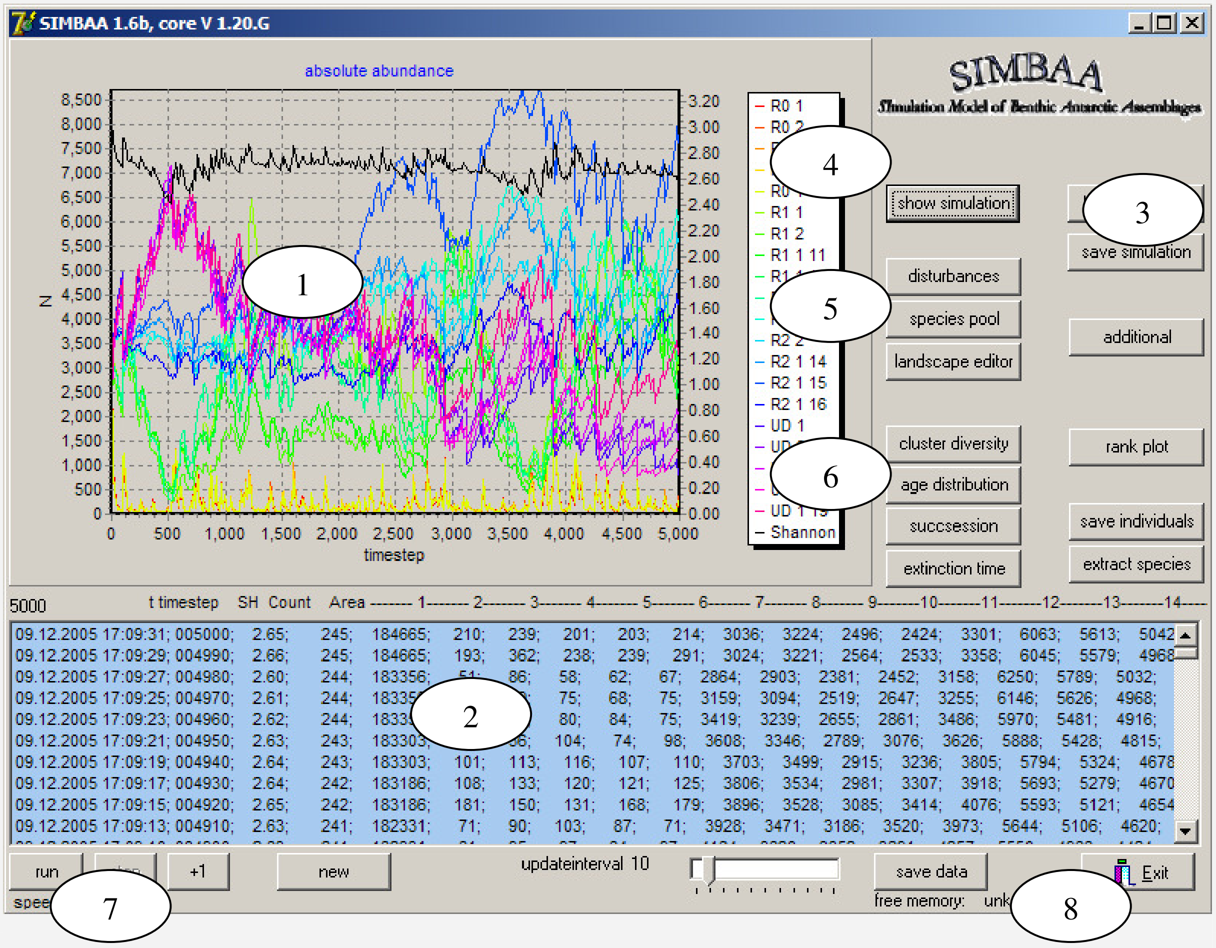
Formula 4, exponential growth function

1.5 The SIMBAA Graphical User Interface (SIMBAA-GUI)

The SIMBAA GUI is the main user interface for the simulation program. Simulations can be created, executed and evaluated on the Windows-desktop using this interface. When the program is started, first an info-screen is shown and then the main window (Figure 1) is loaded.

The following sections will show a screen-shot of the several windows used to interact with SIMBAA. A short explanation will be given. Circles with inscribed numbers make interface elements that are further descript in the text.

1.5.1 The main window

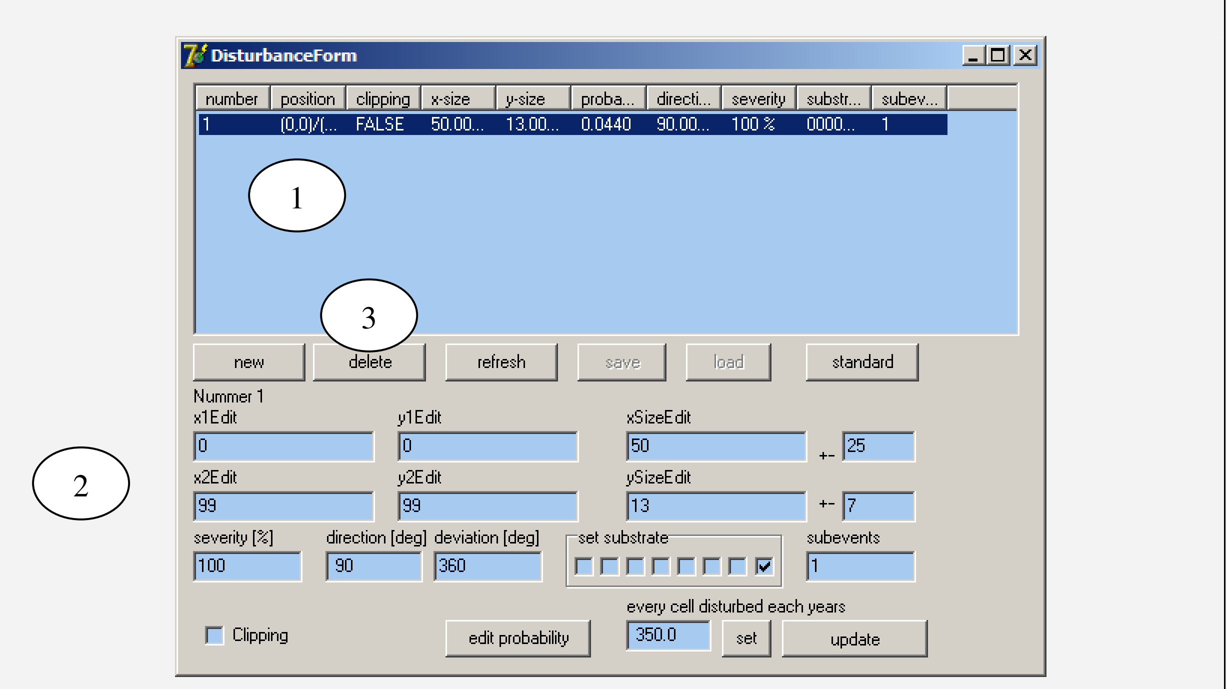


```
5000            t timestep     SH  Count    Area ------- 1------- 2------- 3------- 4------- 5------- 6------- 7------- 8------- 9-------10-------11-------12-------13-------14----
09.12.2005 17:09:31; 005000;  2.65;    245;  184665;   210;   239;   201;   203;   214;  3036;  3224;  2496;  2424;  3301;  6063;  5613;  5042
09.12.2005 17:09:29; 004990;  2.66;    245;  184665;   193;   362;   238;   239;   291;  3024;  3221;  2564;  2533;  3358;  6045;  5579;  4968
09.12.2005 17:09:27; 004980;  2.60;    244;  183356;    51;    86;    58;    62;    67;  2864;  2903;  2381;  2452;  3158;  6250;  5789;  5032;
09.12.2005 17:09:25; 004970;  2.61;    244;  18335[illegible]     75;    68;    75;  3159;  3094;  2519;  2647;  3255;  6146;  5626;  4968;
09.12.2005 17:09:23; 004960;  2.62;    244;  1833[illegible]      80;    84;    75;  3419;  3239;  2655;  2861;  3486;  5970;  5481;  4916;
09.12.2005 17:09:21; 004950;  2.63;    243;  183303[illegible] 56;   104;    74;    98;  3608;  3346;  2789;  3076;  3626;  5888;  5428;  4815;
09.12.2005 17:09:19; 004940;  2.64;    243;  183303;   101;   113;   116;   107;   110;  3703;  3499;  2915;  3236;  3805;  5794;  5324;  4678
09.12.2005 17:09:17; 004930;  2.64;    242;  183186;   108;   133;   120;   121;   125;  3806;  3534;  2981;  3307;  3918;  5693;  5279;  4670
09.12.2005 17:09:15; 004920;  2.65;    242;  183186;   181;   150;   131;   168;   179;  3896;  3528;  3085;  3414;  4076;  5593;  5121;  4654
09.12.2005 17:09:13; 004910;  2.63;    241;  182331;    71;    90;   103;    87;    71;  3928;  3471;  3186;  3520;  3973;  5644;  5106;  4620;
```

Figure 7, Main SIMBAA window elements:

1. Visualisation of the development of the population structure over time
2. List of population structure and other simulation parameter
3. Buttons to load/save the current simulation
4. Open the spatial visualisation tool window
5. Edit simulation parameter, e.g. disturbances, species pool, landscape
6. Different analyse functions
7. Start/stop the current simulation
8. Exit the programme

1.5.2 The disturbance editor



number	position	clipping	x-size	y-size	proba...	directi...	severity	substr...	subev...
1	(0,0)/(...	FALSE	50.00...	13.00...	0.0440	90.00...	100 %	0000...	1

Nummer 1

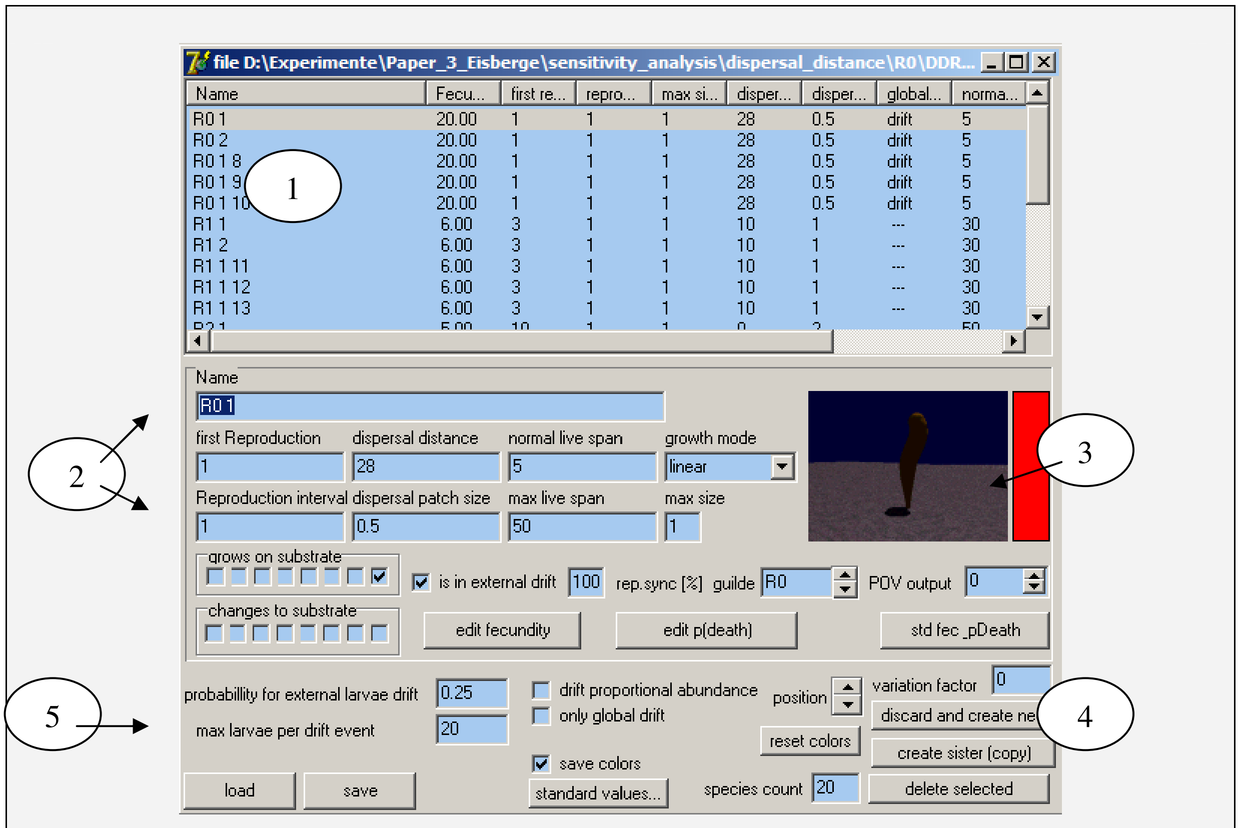
x1Edit: 0	y1Edit: 0	xSizeEdit: 50 +- 25
x2Edit: 99	y2Edit: 99	ySizeEdit: 13 +- 7

severity [%]: 100
direction [deg]: 90
deviation [deg]: 360
set substrate: ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☑
subevents: 1
☐ Clipping
every cell disturbed each years: 350.0

Figure 8, The disturbance editor window

When selecting the button "**disturbance**" in the main window (Figure 7) the disturbance editor is opened (Figure 8). This editor is used to create, change and delete disturbance events. In the upper part (1) all defined disturbance events are listed. The user can select one disturbance out of the list by a double-click. Then the information (2) is updated. On the left side, the area where the disturbance may start [(x1/y1) - (x2/y2)] and clipping can be defined. The disturbance size can be edited on the right part of the panel. In the lower row disturbance severity, directional information, resulting substrate state and number of sub-events can be entered.
The rotation period can be directly entered. Please use "set" to calculate the according probability. However, it is also possible to enter the disturbance probability directly by the "edit probability"-button. Then a new input window is opened.
Please use the "update"-button, especially when it is coloured in red, to update the disturbance definition.
(3) gives you the possibility to create a new disturbance with the information of the edit-panel ("new") or delete existing disturbances form the list.

1.5.3 The species editor



file D:\Experimente\Paper_3_Eisberge\sensitivity_analysis\dispersal_distance\R0\DDR...

Name	Fecu...	first re...	repro...	max si...	disper...	disper...	global...	norma...
R0 1	20.00	1	1	1	28	0.5	drift	5
R0 2	20.00	1	1	1	28	0.5	drift	5
R0 1 8	20.00	1	1	1	28	0.5	drift	5
R0 1 9	20.00	1	1	1	28	0.5	drift	5
R0 1 10	20.00	1	1	1	28	0.5	drift	5
R1 1	6.00	3	1	1	10	1	---	30
R1 2	6.00	3	1	1	10	1	---	30
R1 1 11	6.00	3	1	1	10	1	---	30
R1 1 12	6.00	3	1	1	10	1	---	30
R1 1 13	6.00	3	1	1	10	1	---	30
R2 1	5.00	10	1	1	0	2		50

Name: R0 1

first Reproduction: 1
dispersal distance: 28
normal live span: 5
growth mode: linear

Reproduction interval: 1
dispersal patch size: 0.5
max live span: 50
max size: 1

grows on substrate: ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☑

changes to substrate: ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐

☑ is in external drift 100 rep.sync [%] guilde R0 POV output 0

probabillity for external larvae drift: 0.25
max larvae per drift event: 20

☐ drift proportional abundance
☐ only global drift
☑ save colors

variation factor: 0

species count: 20

Figure 9, The species editor window

When selecting “species pool” in the main window (Figure 7), the species pool editor is opened (Figure 9). Using the species editor all species traits can be controlled. For a detailed description of all species traits see preceding chapter.

The upper panel (1) lists all defined species with their traits. By a click in this list a species can be selected. The traits of the selected species can be edited in the middle panel (2). The buttons “edit fecundity” and “edit p(death)” open new input windows to edit fecundity, respectively mortality. On the right side, a graphical representation is shown and colour information is shown (The picture shows a early version of the POV output). By default, SIMBAA tries to assign each species a unique colour in the spectrum, determined by the position in the species list. A click on the colour bar opens a standard colour dialog, where the user can select the colour. The lower right part of the panel (4) contains elements to control and manipulate the pool. With the position-buttons, the currently selected species can be moved up and down in the list. “reset colours” re-calculates the colour scheme according to the current species order. If “save colors” is not selected, the colour information is not saved in the simulation file and recalculated on loading each time. “Discard and create new”

removes the current species pool and creates a new one with “species count” species. “create sister” and “deleted selected” create a copy of the current selected species, respectively removes the selected species from the pool.
On the bottom left the global drift (migration) properties can be modified. Please see the dispersal chapter for more details. You can define the probability for a migration event and the maximal number of larvae added to the local larvae pool in case of such an event. If “drift proportional abundance” is selected, the probability for a species to occur in the global drift is proportional to its (global) abundance. When “only global drift” is selected, no explicit larval dispersal is computed (**thus SIMBAA’s dispersal model is turned off**) and only global drift is used.

1.5.4 The landscape editor

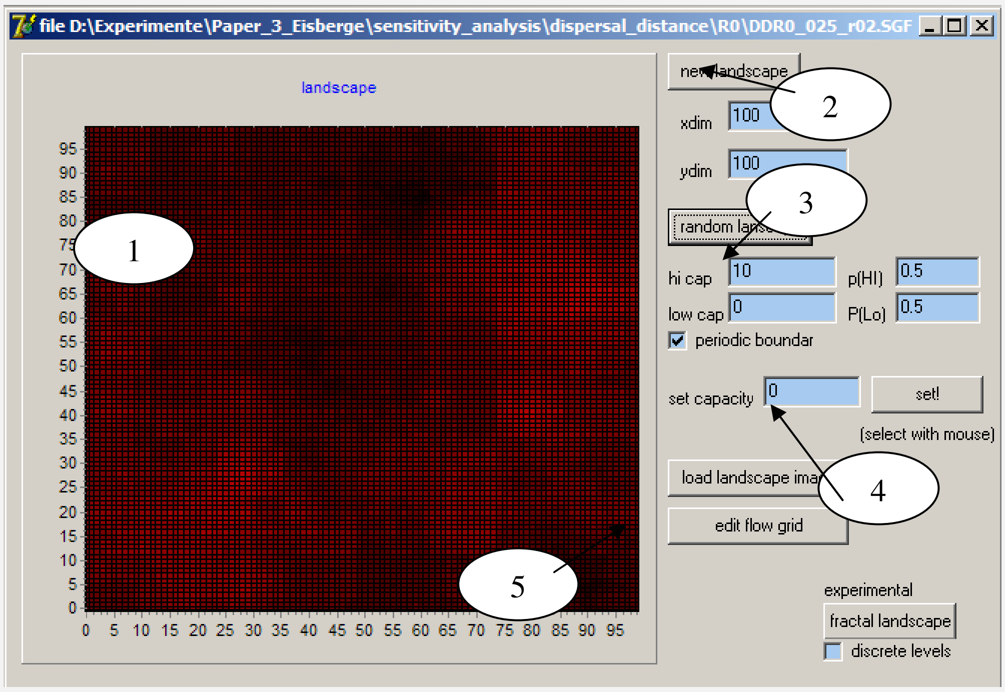


Figure 10, The landscape editor

The landscape editor (Figure 10), where the simulation grid is defined is opened by the button "landscape editor" in the main window (Figure 7). On the left panel (1), the complete simulation grid is shown. Different shades of red indicate the cell capacity. As lighter the colour, as higher the cell capacity (number of individuals, the cell can support). A white colour represents matrix cells with a zero capacity. These cells do not support any individual and will remain empty. On the right panel, the simulation grid size can be defined (2). If you create a new landscape, all cells will start with a zero capacity!

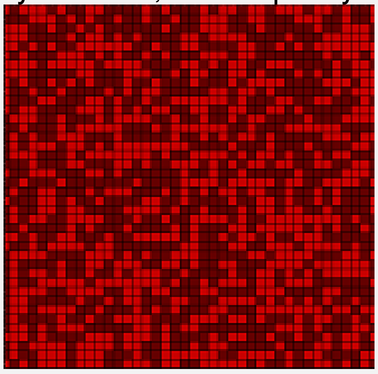
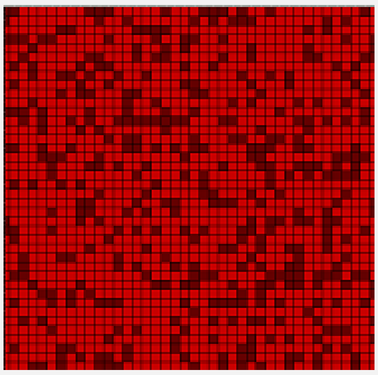
! **NOTE:** **!**

Changing the simulation grid clears all disturbance event definitions!

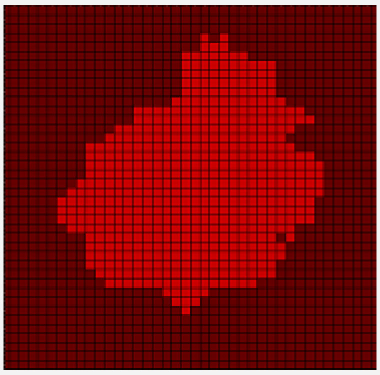
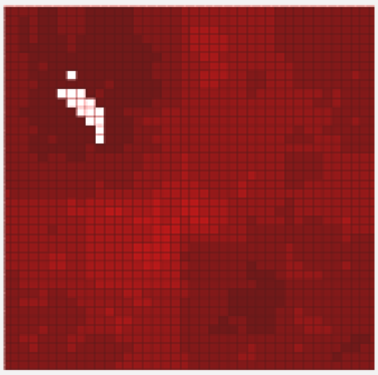
You can now create a random distribution of the cell capacity (3). SIMBAA supports "high capacity" sites and "low capacity" sites. These are distributed according to the given probabilities. Note: if you change a probability, the other is automatically updated! You can also define the capacity of a rectangular area by hand (4). Enter the desired capacity and click on “set” Then select the area be pressing the left mouse and dragging while keeping the mouse button pressed. Use the “edit flow grid” button (5) to invoke the flow grid editor (see below).

As experimental feature you can use a midpoint displacement algorithm (commonly known as “fractal” landscape generation algorithm) to create spatial correlated distribution of the cell capacities. When “discrete levels” is enabled, only “high capacity” and “low capacity” cells are created. When disabled, all integer values between “low” and “high” are used.
You can also load a bitmap with the capacity information encoded as grey scales. The picture is internally converted into grey scale (if coloured) and scaled to fit the simulation grid. The cell capacity is set between “low cap” and “high cap” according to the grey value. As darker the grey value, as higher the resulting cell capacity. A white colour in the image will result in a matrix cell with zero capacity. See the example below.

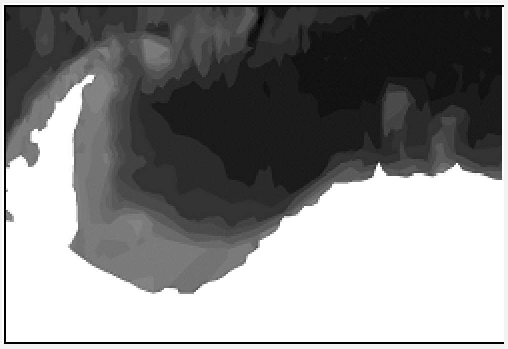
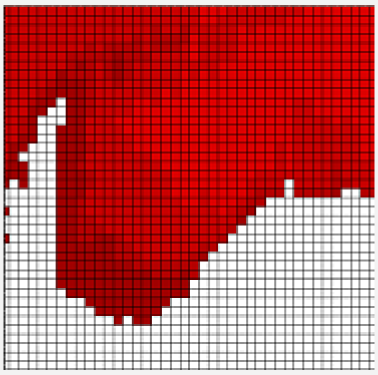
Examples of capacity distribution maps created by different approaches. “Hi capacity” was 10, “low capacity”= 5 in all cases.

Random capacity maps with different probabilities. On the left, both cell types have the same probability (0.5 each). On the right, the high capacity sites (light red) have a probability of 0.75 and the low capacity sites accordingly 1-0.75 = 0.25.

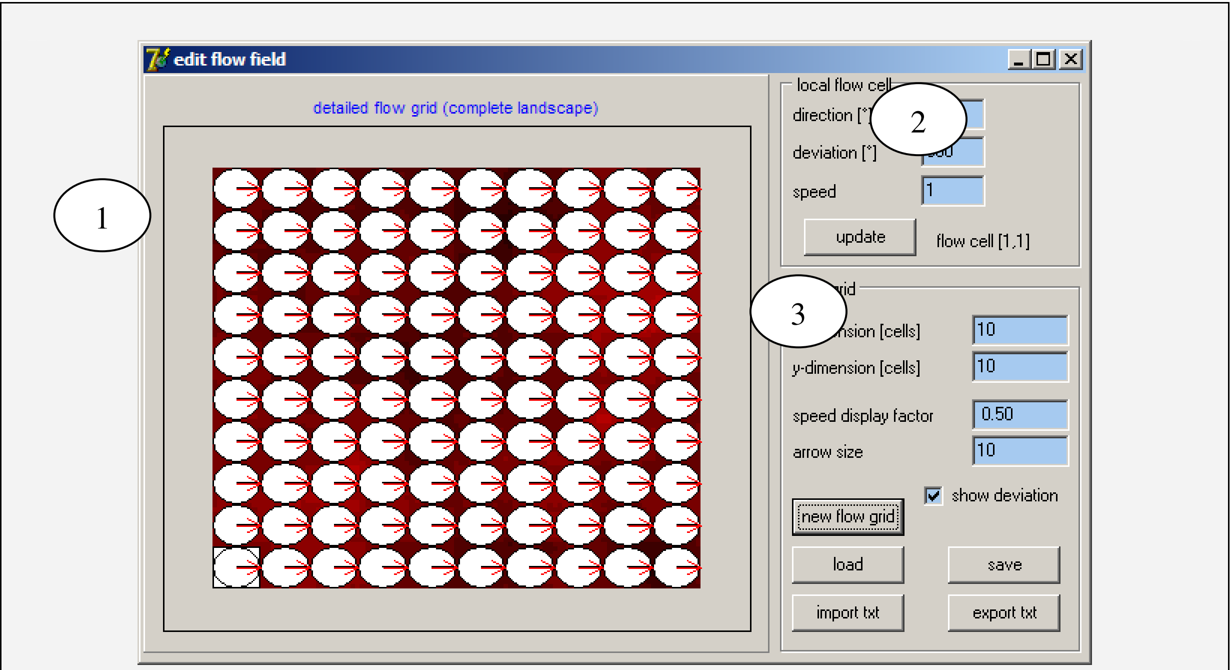



Example of “fractal” capacity maps. On the left “discrete level” was used, whereas on the right, all levels are used.

This is an example for a grey scale image loaded as information source for the capacity of the landscape. The image is a depth map of the Weddell Sea, thus the depth information will be encoded as capacity. Note the stretching as the image is rectangular but the simulation grid is quadratic. The white area of the image (shelf ice and land region) is translated in matrix cells with zero capacity, the darker grey colours in higher capacities.

1.5.5 The flow editor



direction [°]
deviation [°]
speed 1
flow cell [1,1]
x-dimension [cells] 10
y-dimension [cells] 10
speed display factor 0.50
arrow size 10
[✓] show deviation

Figure 11, The flow grid editor

This editor window (Figure 11) is displayed when the "**edit flow grid**"-button of the landscape editor (Figure 10) is pressed. The left panel (1) shows a picture of the simulation grid (cell capacities as shades of red) and the flow direction, represented by an arrow. The length and orientation of this arrow are proportional to the flow speed and direction of the flow in that cell. If the "show deviation" checkbox is checked, the deviation is displayed as sector (see above).

To select a flow cell, click on the corresponding arrow. A white frame highlights the selected flow cell and in the upper right panel (2) the flow properties of the selected cell is shown and accessible.

In the lower right panel (3) you can define the size of the flow grid. You can also load and save the flow grid and export/ import as text file. If you create a new flow field, all flow cells will contain the flow properties defined in the upper panel.

The "speed display factor" and "arrow size" are just used to display the flow field. If you have different flow speeds, you can use the first to scale the length of the arrow, the second parameter determines the size of the arrows point.

1.5.6 The visualise tool window

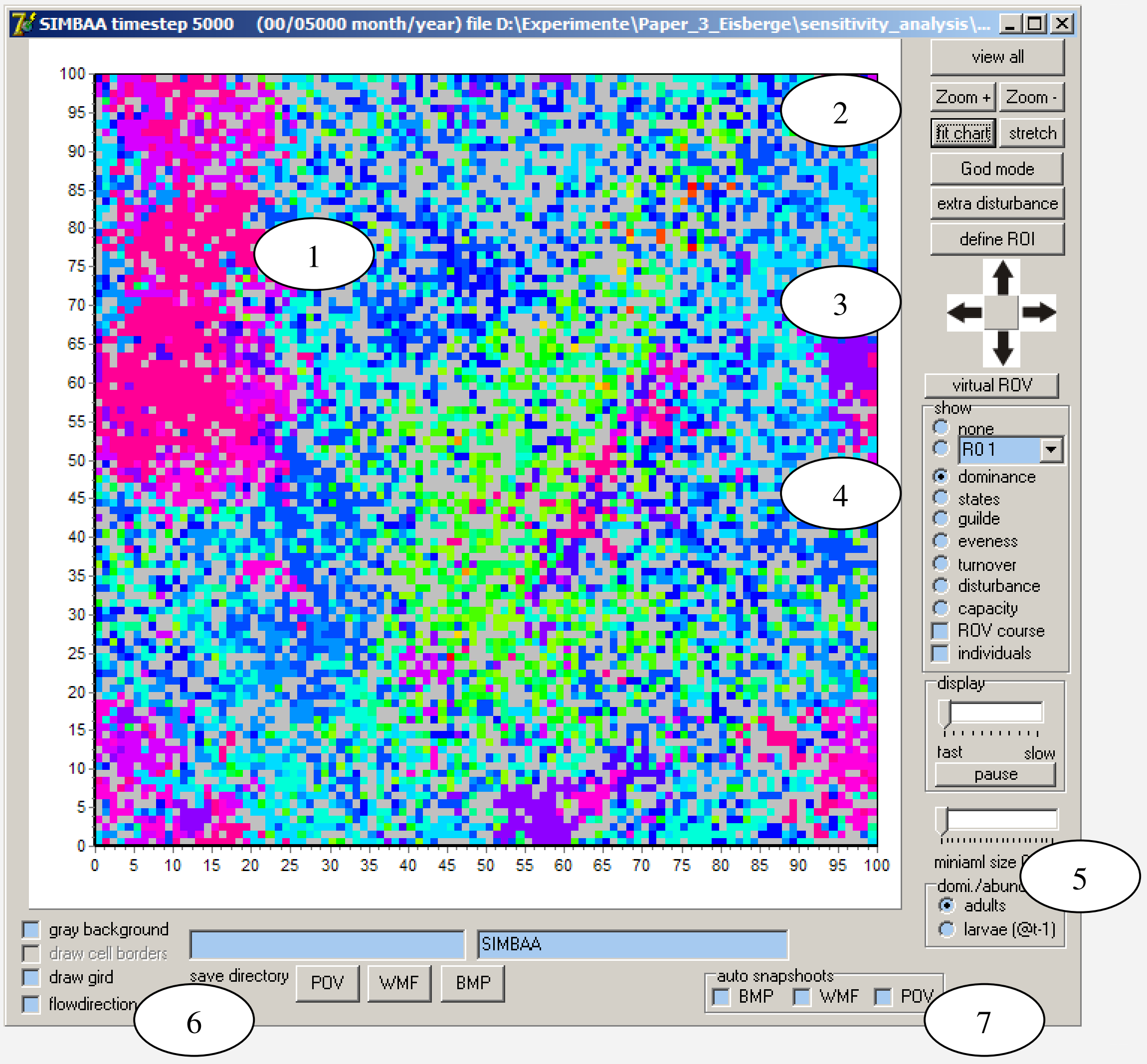


Figure 12, The visualisation tool window

This is the main visual inspection tool. You can check the simulation grid and display various aspects of a simulation run. The main panel (1) shows a graphical representation of the grid. It is possible to zoom (left-button) and pan (right-button) the image by the mouse or the cursor buttons. With the buttons in the upper right (2) the user can reset the viewport to the whole simulation. The button “fit chart” tries to rescale the left panel to archive an aspect ration of 1:1 whereas “stretch” rescales the panel to use all available window space. You can also move the visual part by the cursor buttons (3). The view is updated on a regular interval (see main panel). However, if the simulation is not running or the update interval is too sparse, the middle button of (3) redraws the current view. If the simulation is too fast, the middle panel “display” can be used to slow down the redraw or even pause the simulation.

You can display various aspects using the middle right panel (4).

Display options:

None:	No cell information is displayed, only (when enabled) virtual ROV transect and individuals
Species:	Select a species of the drop-down list to display its spatial distribution. The intensity of the colour is proportional to the number of individuals of the species in a cell.
States:	Succession states according to the current state definition. **R0** is red, **R1** yellow, **R2** green and **UD** blue.
Guild:	Displays the dominant guild of a cell. Same colour-schema as above: R0 is red, R1 yellow, R2 green and UD blue.
Evenness:	Displays Pielou’s evenness *J’* based on the species composition of the cell. Colour scheme: like a spectrum with low evenness (few species) represented by cold (blue) and a high evenness (many species) as hot (red) colours.
Turnover:	Different shades of blue show the overall disturbance history of a cell. As darker as fewer disturbances events occurred at a cell.
Disturbances:	Show a detailed map of the disturbance history within a certain time period. As brighter the red colour as younger the disturbance is. Cells with have not been disturbed since a defined time (not the changing slider “minimal size” respectively “max age”!) are shaded grey.
Capacity:	Shows the cell capacity in different shades of red like in the landscape editor. Bright shades represent higher capacities.

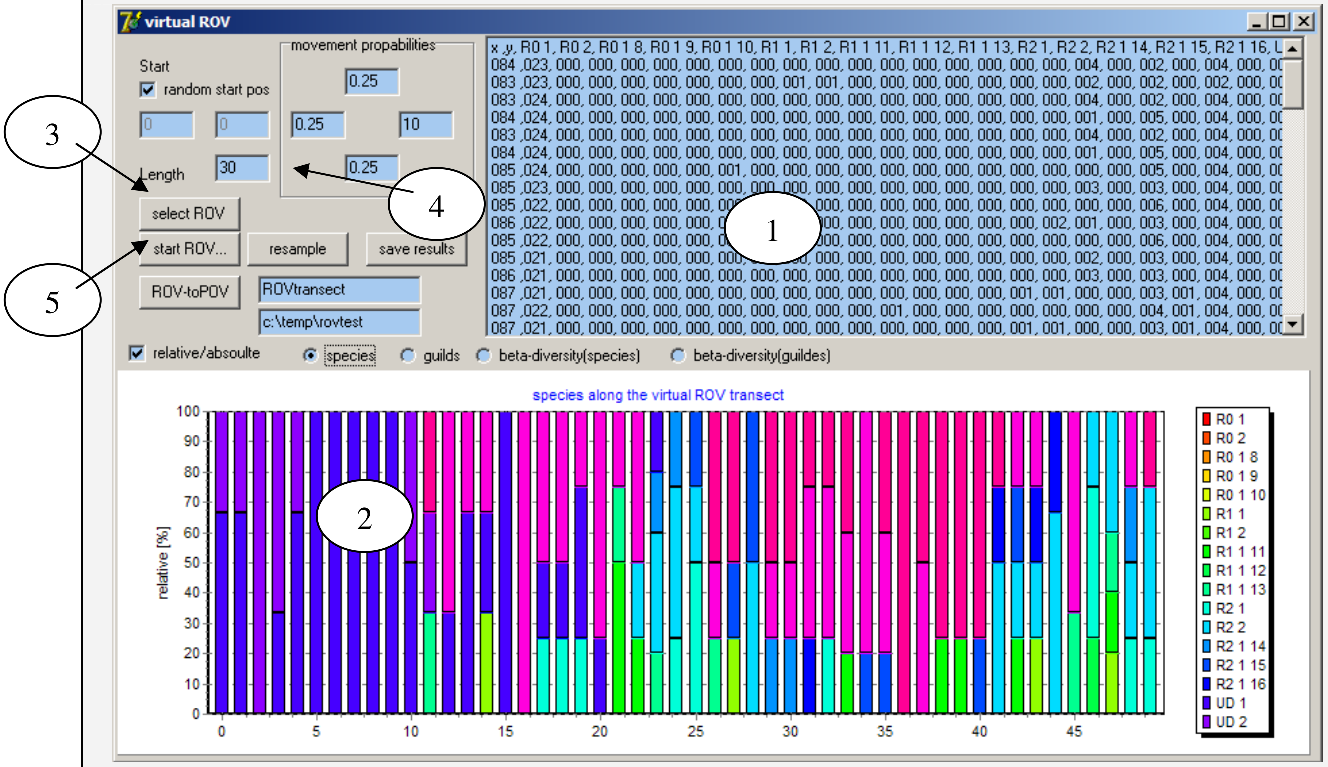
Switch “**ROV course**”: displays the current virtual ROV track as light blue line on top of the map

Switch “**Individuals**”: draw each individual as a circle. The colour of the circle represents the species and the diameter is proportional to the size of the individual (see growth-function). NOTE: drawing individuals can be quit time consuming and slows down the whole simulation!

It is possible to use either the abundance of established individuals over a certain size (slider “min size for visual”) or the larvae distribution **of the previous time step** (“larvae @t-1”) when displaying dominance and species pattern. **NOTE: the value “minimal size for visual” determines the size of any individual to be considered by several functions of the model and is used throughout the simulation. E.g. all individuals below this size are not considered for diversity measurements and not displayed as individuals and so!**

The lower panel (6) allows toggling various display options (e.g. cell border etc.). It is also possible to generate snapshots in various formats on demand or automatically, each time the map is redrawn. These are stored in the directory given by the first entry field “save directory” and with a filename consisting of the given name appended by the time step. Example: “save directory= “c:\simulation\snapshot\“, “name=TEST” and “BMP” will generate “test00000.bmp”, test00001.bmp” in the directory “c:\simulation\snapshot” and so forth. **Note the last slash in the save director!**

1.5.7 The virtual ROV



virtual ROV

Start
[✓] random start pos
0 0
Length 30

movement propabilities
0.25
0.25 10
0.25

select ROV
start ROV... resample save results
ROV-toPOV ROVtransect
c:\temp\rovtest

[✓] relative/absoulte [✓ species] guilds beta-diversity(species) beta-diversity(guildes)

```
x ,y, R0 1, R0 2, R0 1 8, R0 1 9, R0 1 10, R1 1, R1 2, R1 1 11, R1 1 12, R1 1 13, R2 1, R2 2, R2 1 14, R2 1 15, R2 1 16, L
084 ,023, 000, 000, 000, 000, 000, 000, 000, 000, 000, 000, 000, 000, 000, 000, 000, 000, 004, 000, 002, 000, 004, 000, 00
083 ,023, 000, 000, 000, 000, 000, 000, 000, 001, 001, 000, 000, 000, 000, 000, 000, 000, 002, 000, 002, 000, 002, 000, 00
083 ,024, 000, 000, 000, 000, 000, 000, 000, 000, 000, 000, 000, 000, 000, 000, 000, 000, 004, 000, 002, 000, 004, 000, 00
084 ,024, 000, 000, 000, 000, 000, 000, 000, 000, 000, 000, 000, 000, 000, 000, 000, 000, 001, 000, 005, 000, 004, 000, 00
083 ,024, 000, 000, 000, 000, 000, 000, 000, 000, 000, 000, 000, 000, 000, 000, 000, 000, 004, 000, 002, 000, 004, 000, 00
084 ,024, 000, 000, 000, 000, 000, 000, 000, 000, 000, 000, 000, 000, 000, 000, 000, 000, 001, 000, 005, 000, 004, 000, 00
085 ,024, 000, 000, 000, 000, 000, 001, 000, 000, 000, 000, 000, 000, 000, 000, 000, 000, 000, 000, 005, 000, 004, 000, 00
085 ,023, 000, 000, 000, 000, 000, 000, 000, 000, 000, 000, 000, 000, 000, 000, 000, 000, 003, 000, 003, 000, 004, 000, 00
085 ,022, 000, 000, 000, 000, 000, 000, [illegible], [illegible], 000, 000, 000, 000, 000, 000, 000, 000, 000, 000, 006, 000, 004, 000, 00
086 ,022, 000, 000, 000, 000, 000, [illegible], [illegible], [illegible], 000, 000, 000, 000, 000, 000, 000, 002, 001, 000, 003, 000, 004, 000, 00
085 ,022, 000, 000, 000, 000, 000, [illegible], [illegible], [illegible], 000, 000, 000, 000, 000, 000, 000, 000, 000, 000, 006, 000, 004, 000, 00
085 ,021, 000, 000, 000, 000, 000, 000, 000, 000, 000, 000, 000, 000, 000, 000, 000, 000, 002, 000, 003, 000, 004, 000, 00
086 ,021, 000, 000, 000, 000, 000, 000, 000, 000, 000, 000, 000, 000, 000, 000, 000, 000, 003, 000, 003, 000, 004, 000, 00
087 ,021, 000, 000, 000, 000, 000, 000, 000, 000, 000, 000, 000, 000, 000, 000, 001, 001, 000, 000, 003, 001, 004, 000, 00
087 ,022, 000, 000, 000, 000, 000, 000, 000, 000, 000, 000, 001, 000, 000, 000, 000, 000, 000, 000, 004, 001, 004, 000, 00
087 ,021, 000, 000, 000, 000, 000, 000, 000, 000, 000, 000, 000, 000, 000, 000, 001, 001, 000, 000, 003, 001, 004, 000, 00
```

Figure 13, The virtual ROV

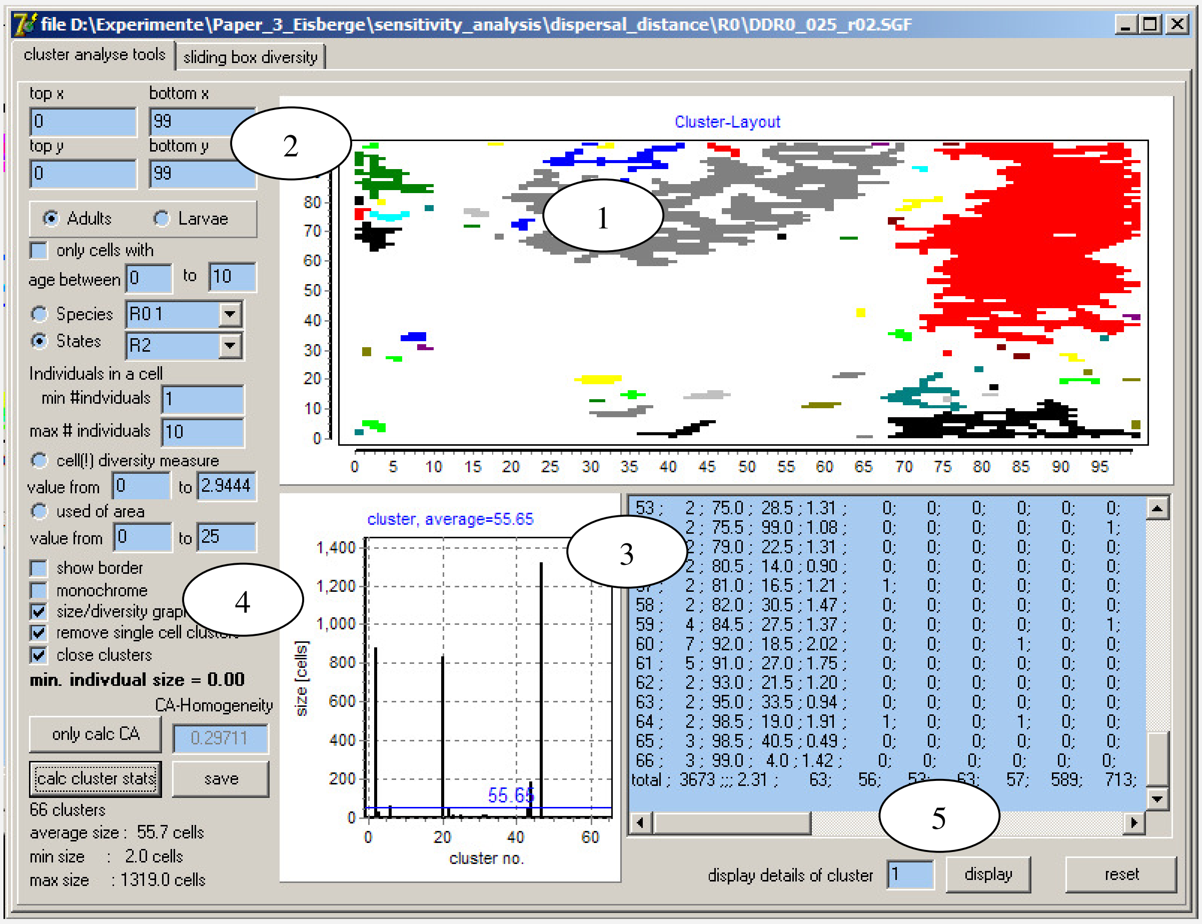
Figure 13 shows the virtual ROV[2] window. With the virtual ROV you can sample the simulation grid similar to a real ROV. The virtual ROV moves along a transect and samples each grid cell on its way and lists its species composition (1). From this a graphical representation is drawn, either on an absolute scale or relative to all cell individuals (2). It is also possible to show the β-diversity based on species respectively guild composition. In this case the first sampled cell is the reference.

The transect can be selected manually (3) or a correlated random walk is performed. When manual selection is desired, the user must select start and endpoint of the transect in the visualiser window by clicking the middle mouse button. Note that the transect is not drawn until selection is complete. When performing a correlated random walk either a random or a defined start position can be used and the length in cells along the transect must be given. Note that a random correlated walk can lead to respectively sample the same cell(s). The probabilities for directional changes

[2] ROV = remote operated vehicle, normally equiped with a camera. Used to sample tansects of the sea floor.

are normalised according to the given numbers (4). In the example the probability to move east is [10/(10+0.25+0.25+0.25)]~0.93 and ~0.02 for each other direction.
Once a transect has been created by starting the virtual ROV (5), it is stored in memory and can be re-sampled. This is very useful when following a temporal development. The results can be saved and an experimental feature allows to create a POV-scene file according to the transect data.

1.5.8 Cluster analysis tool



top x: 0 bottom x: 99
top y: 0 bottom y: 99
[✓ Adults] Larvae
☐ only cells with age between 0 to 10
Species R0 1
[✓ States] R2
Individuals in a cell
min #indviduals 1
max # individuals 10
cell(!) diversity measure value from 0 to 2.9444
used of area value from 0 to 25
☐ show border
☐ monochrome
☑ size/diversity graph
☑ remove single cell cluster
☑ close clusters
min. indivdual size = 0.00
CA-Homogeneity 0.29711
66 clusters
average size : 55.7 cells
min size : 2.0 cells
max size : 1319.0 cells

```
53 ;   2 ; 75.0 ; 28.5 ; 1.31 ;    0;    0;    0;    0;    0;    0;
[illegible]   2 ; 75.5 ; 99.0 ; 1.08 ;    0;    0;    0;    0;    0;    1;
[illegible]   2 ; 79.0 ; 22.5 ; 1.31 ;    0;    0;    0;    0;    0;    0;
[illegible]   2 ; 80.5 ; 14.0 ; 0.90 ;    0;    0;    0;    0;    0;    0;
57 ;   2 ; 81.0 ; 16.5 ; 1.21 ;    1;    0;    0;    0;    0;    0;
58 ;   2 ; 82.0 ; 30.5 ; 1.47 ;    0;    0;    0;    0;    0;    0;
59 ;   4 ; 84.5 ; 27.5 ; 1.37 ;    0;    0;    0;    0;    0;    1;
60 ;   7 ; 92.0 ; 18.5 ; 2.02 ;    0;    0;    0;    1;    0;    0;
61 ;   5 ; 91.0 ; 27.0 ; 1.75 ;    0;    0;    0;    0;    0;    0;
62 ;   2 ; 93.0 ; 21.5 ; 1.20 ;    0;    0;    0;    0;    0;    0;
63 ;   2 ; 95.0 ; 33.5 ; 0.94 ;    0;    0;    0;    0;    0;    0;
64 ;   2 ; 98.5 ; 19.0 ; 1.91 ;    1;    0;    0;    1;    0;    0;
65 ;   3 ; 98.5 ; 40.5 ; 0.49 ;    0;    0;    0;    0;    0;    0;
66 ;   3 ; 99.0 ;  4.0 ; 1.42 ;   0;    0;    0;    0;    0;    0;
total ; 3673 ;;; 2.31 ;    63;    56;    53;    63;    57;   589;   713;
```

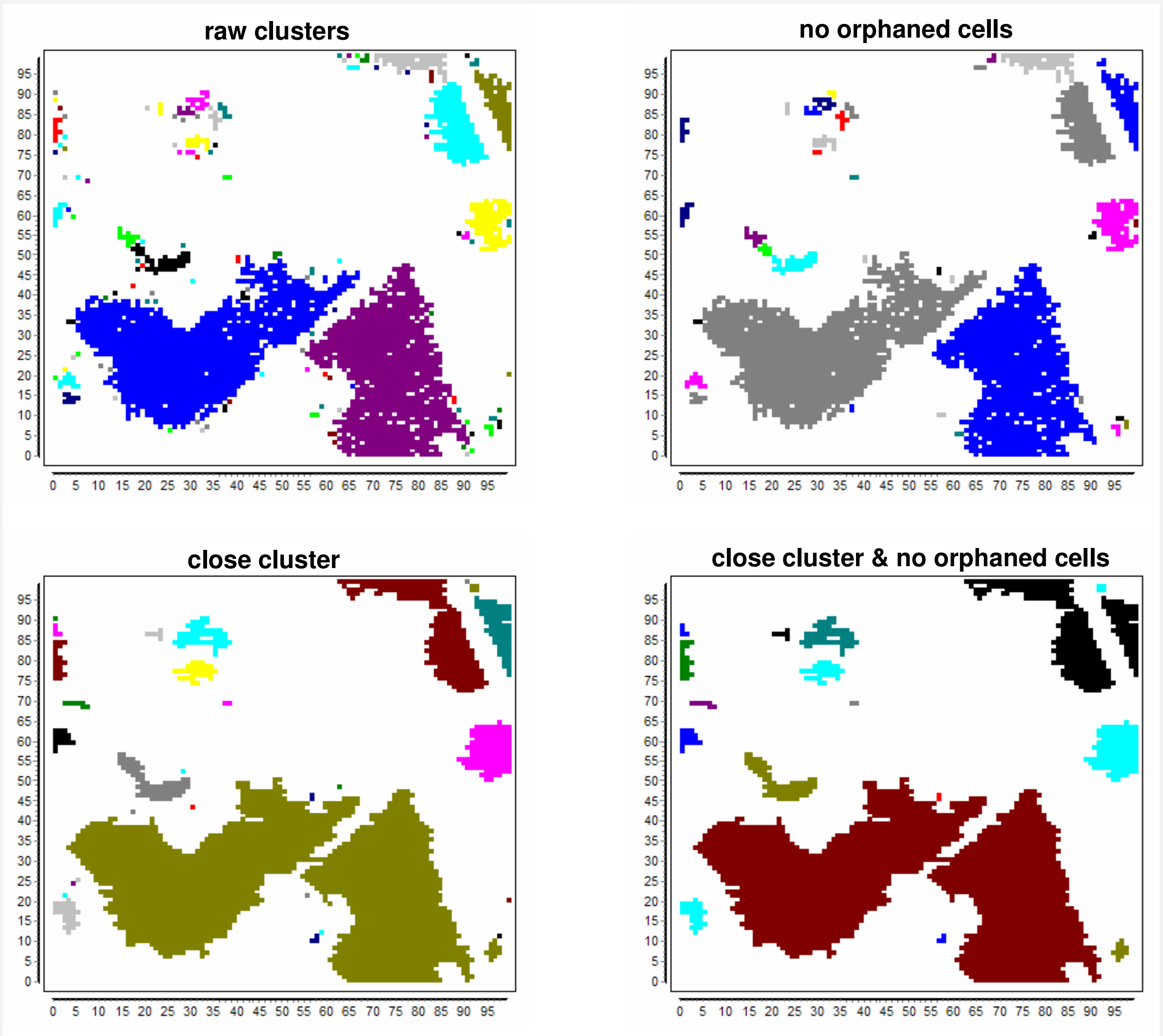
display details of cluster 1

Figure 14, The cluster analysis tool window

This tool allows to estimate and quantify the spatial clustering within a defined area the simulation grid. Based on some criteria (i.e. species, age etc), internally an attribute map of the simulation grid is created. Then clusters of connected regions in this map are identified using a *Hoshen-Kopelmann* (*HK*-) algorithm. As this algorithm internally uses a *von Neumann*-neighbourhood (each cell has four direct neighbours), a cluster may be split into several distinct clusters. This is a well-known phenomena of the algorithm and not avoidable. The tool lists all identified clusters and their properties that can be saved for further analysis. It is possible to apply several manipulations to the feature map to simplify the analysis. The switch “close clusters” applies a “dilate and erode” procedure to the feature map before the cluster identification. This is a common image manipulation filter and results into a “closed” map. The idea is to overcome the constrains of the *HK*-Algorithm and force bigger, cohesive clusters. The second switch “remove single cell cluster” has a similar the effect and removes clusters of a single cell, thus with no direct neighbours, before analysis. (An example can be found below)

The upper panel (1) shows a display of the feature map with the clusters. Identified clusters a randomly coloured when not forced to be monochrome. On the left, the sample area and feature list can be selected (2). Using “calc cluster stats” calculates the clusters and updates the information display (3). On the left part of the summary, a histogram of the size (or diversity) over all identified clusters is shown and on the right side a detailed list of the clusters and their properties. Detailed information on a certain cluster is shown using the number of a cluster and the “display” button on the bottom (5).
Additionally, the so called “*cellular automata (CA) measures for homogeneity*” (*CA*-Homogeneity, {Hütt, 2001 #662}) is computed for the selected feature. This gives the average number of cells with the same nearest neighbourhood configuration.

The following Figure 15 shows the consequences of the operations „no orphaned cells“ and „close clusters“ on a sample data set. Identified cluster are marked with different colours. However, due to technical reasons only 20 different colours were available. Thus adjacent clusters may have the same colour and appear as single cluster.



	identified cluster	average size	standard deviation
raw cluster	125	20.6	124.7
no orphaned cells	41	60.7	213.9
close cluster	32	94.6	402.1
close cluster & no orphaned cells	18	167	531

Figure 15, Example of the cluster identification and resulting cluster properties

1.5.9 Rank analysis tool

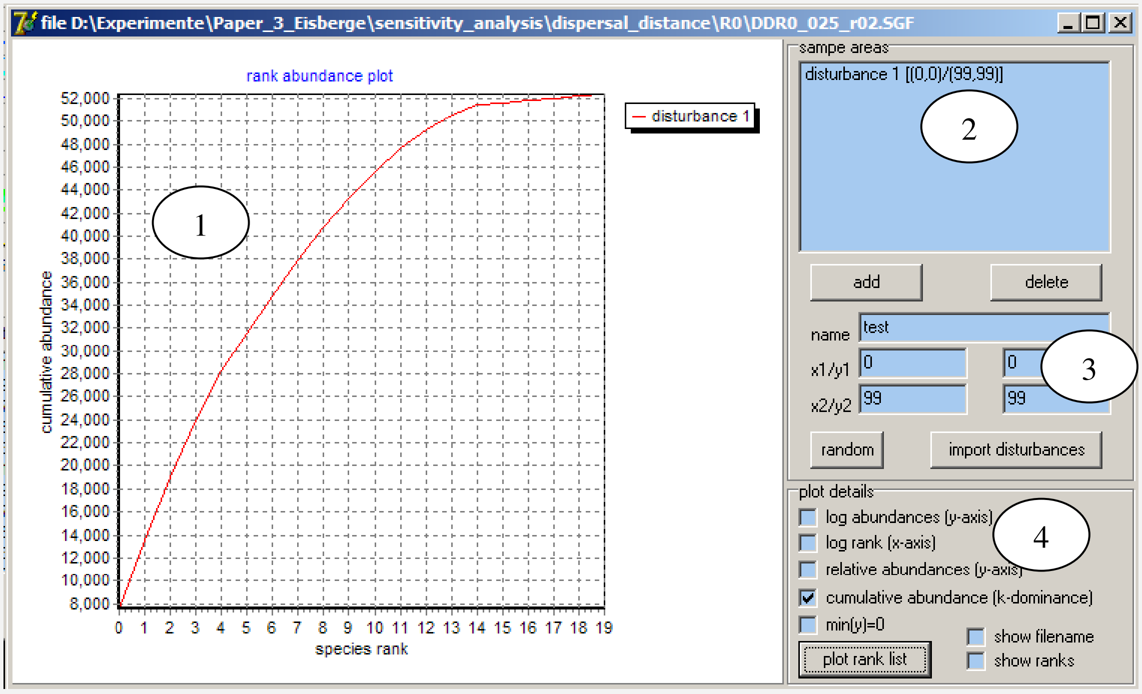
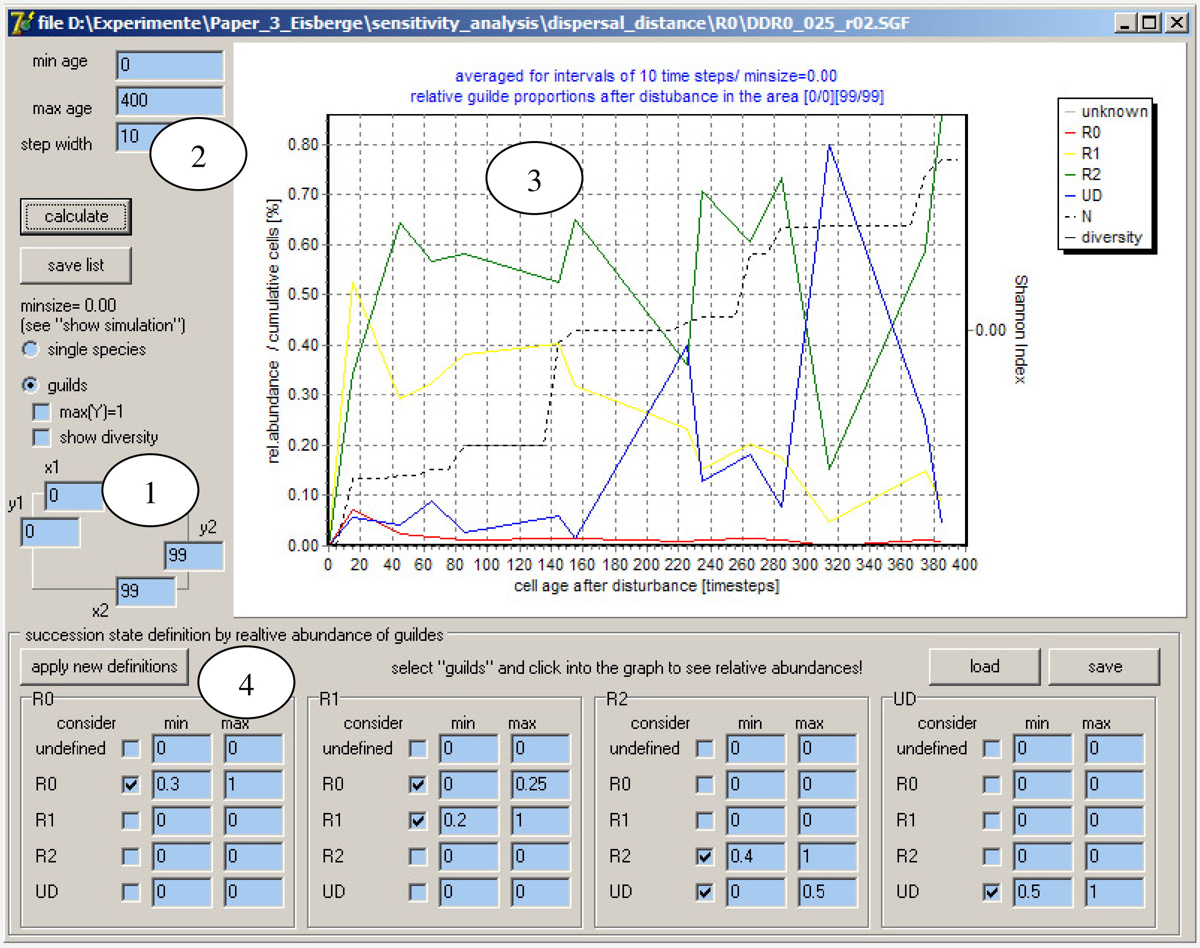


Figure 16, The rank analysis window

This window (Figure 16) allows to sample a defined rectangular area and compute various species-rank plots. The left panel (1) shows the rank plots. On the upper right panel (2) all sampled areas are listed. By default the list is empty. Using “import disturbances” all defined disturbance areas can be imported. The user can define own sample areas or create random sample areas as well (3). On the lower right panel (4) various plot options can be configured. If the switch “show filename” is enabled, the current filename is displayed in the plot caption. “show ranks” adds the species names in order of their rank to the plot.

1.5.10 Succession analysis tool, succession state definition



file D:\Experimente\Paper_3_Eisberge\sensitivity_analysis\dispersal_distance\R0\DDR0_025_r02.SGF

min age: 0
max age: 400
step width: 10

minsize= 0.00
[see "show simulation"]
single species
[✓ guilds]
max(Y)=1
show diversity

x1: 0
y1: 0
y2: 99
x2: 99

averaged for intervals of 10 time steps/ minsize=0.00
relative guilde proportions after distubance in the area [0/0][99/99]

succession state definition by realtive abundance of guildes

select "guilds" and click into the graph to see relative abundances!

R0

	consider	min	max
undefined	☐	0	0
R0	✓	0.3	1
R1	☐	0	0
R2	☐	0	0
UD	☐	0	0

R1

	consider	min	max
undefined	☐	0	0
R0	✓	0	0.25
R1	✓	0.2	1
R2	☐	0	0
UD	☐	0	0

R2

	consider	min	max
undefined	☐	0	0
R0	☐	0	0
R1	☐	0	0
R2	✓	0.4	1
UD	✓	0	0.5

UD

	consider	min	max
undefined	☐	0	0
R0	☐	0	0
R1	☐	0	0
R2	☐	0	0
UD	✓	0.5	1

Figure 17, The succession analysis window

This window (Figure 17) allows to examine the succession of a defined area of the simulation grid (1). By default, the whole grid is sampled. All cells of a given age range (time since last disturbance) are considered and grouped in bins of a given width (2). All individuals in a bin are considered and the average species and guild composition of the bin is computed and displayed (3). Additionally the diversity of a bin can be shown.

In the above example all cells of the area (0/0) - (99/99) that have been disturbed in the last 400 time steps (age between 0 and 400) are considered. These are binned in steps of 10 time steps. Thus the first bin contains all cells with an age of 0-9 time steps, the second all with an age of 10-19 time steps and so on. The average species (or guild) composition is drawn with the average cell age of a bin as x-axis coordinate. If the relative guild composition is displayed, on can click on a data point and the relative guild composition of this point is shown in a popup window.

In the lower panel (4), the guild composition of the succession states R0, R1, R2 and UD can be defined. This information is used to classify a cell to be in a certain succession state according to its species composition, respectively guild composition. For each succession state the (relative) guild composition can be defined. An enabled checkbox of a guild means that this guild is considered and a minimal and maximal range (relative proportion of the guild) must be given. If two ore more guilds are used, all conditions must match to identify a state. The state definition is saved within a simulation. However, you can save and load to separate files as well.

In the above example, the **state R0** is defined solely by a relative proportions of guild R0 to be between 0.3 and 1. This means that any cell in which more than 30% of all individuals are members of the R0-guild is classified the be in the **R0 state**. For a cell to be classified in the **R1 state**, the relative proportion of individuals belonging to the R0 guild must be smaller than 25% and more than 20% must belong to the R1-guild. Analogue, the **R2 state** is defined by more than 40% individuals of the R2 guild and less than 50% UD guild members. Finally, the **UD-state** is characterised by more than 50% of all individuals belonging to the UD - guild.

!

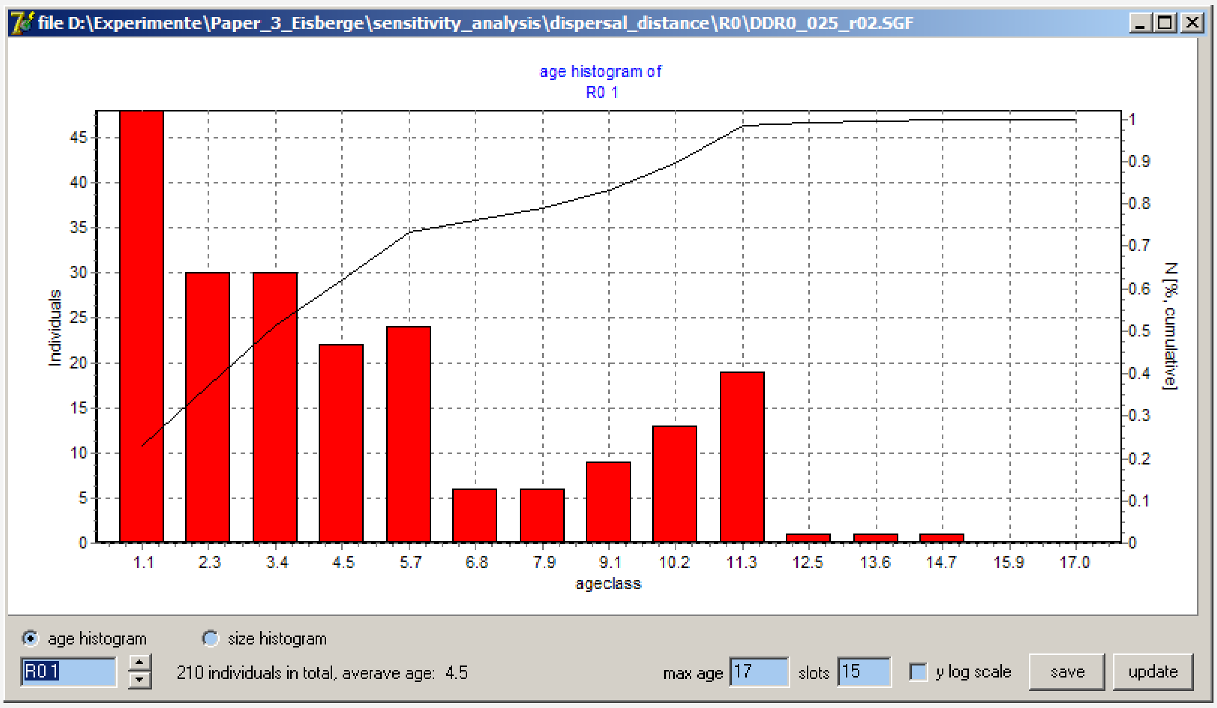
Attention:

Do not confuse guild and state definition! An assemblage of several guilds often defines a state. If you want to say "less than *x*%" follow the above example and use "min = 0, max = *x*". Analogue for defining "more than *z*%" use "min = *z*, max = 1".

Avoid ambiguous state definitions. The states are checked in ascending the order (R0, R1, R2, UD) and the first match is used to identify a state. If no match is found, the state is set to be "undefined".

!

1.5.11 Age structure analysis tool



R0 1

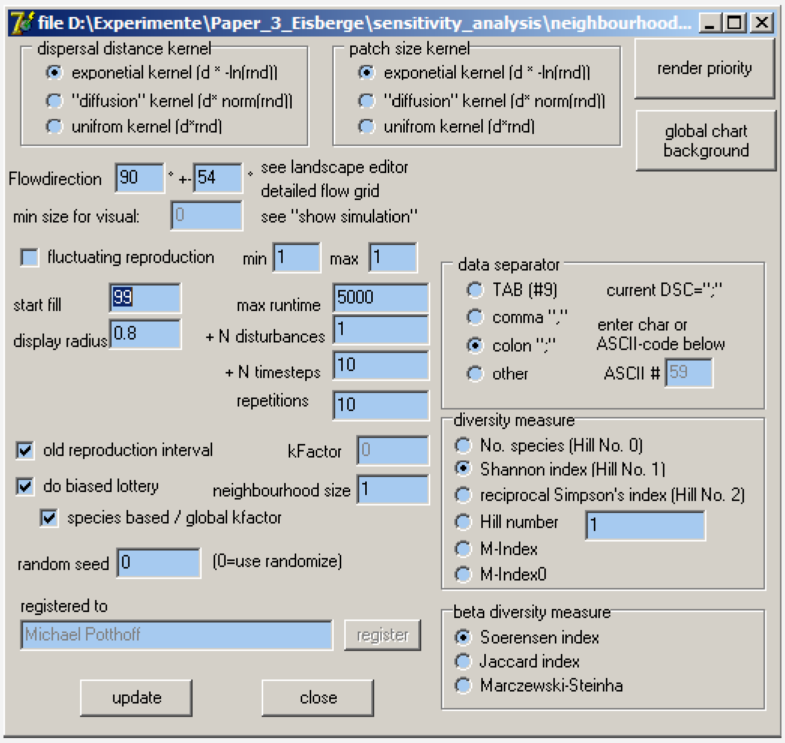
210 individuals in total, averave age: 4.5

max age 17 slots 15 y log scale

Figure 18, The age analysis tool

Figure 18 show the age analysis tool. This tool can be used to create an age or size histogram of a species. Always the whole simulation grid is sampled. You can select the species of interest and define how many bins the histogram should have and the maximal age considered. By default, the maximal age is computed to potentially contain more than 90% of a population according to the mortality. The graph shows the histogram and the cumulative distribution. Additionally, the average age is given as well as how many individuals are counted and, if any, are outside the specified range.

1.6 Additional simulation parameter dialog



file D:\Experimente\Paper_3_Eisberge\sensitivity_analysis\neighbourhood...

dispersal distance kernel
- [✓ exponetial kernel (d * -ln(rnd))]
- "diffusion" kernel (d* norm(rnd))
- unifrom kernel (d*rnd)

patch size kernel
- [✓ exponetial kernel (d * -ln(rnd))]
- "diffusion" kernel (d* norm(rnd))
- unifrom kernel (d*rnd)

render priority

global chart background

Flowdirection 90 ° +- 54 ° see landscape editor detailed flow grid

min size for visual: 0 see "show simulation"

fluctuating reproduction min 1 max 1

start fill 99

display radius 0.8

max runtime 5000

+ N disturbances 1

+ N timesteps 10

repetitions 10

data separator
- TAB (#9)
- comma ","
- [✓ colon ";"]
- other

current DSC=";"

enter char or ASCII-code below

ASCII # 59

[✓] old reproduction interval kFactor 0

[✓] do biased lottery neighbourhood size 1

[✓] species based / global kfactor

random seed 0 (0=use randomize)

registered to

Michael Potthoff register

diversity measure
- No. species (Hill No. 0)
- [✓ Shannon index (Hill No. 1)]
- reciprocal Simpson's index (Hill No. 2)
- Hill number 1
- M-Index
- M-Index0

beta diversity measure
- [✓ Soerensen index]
- Jaccard index
- Marczewski-Steinha

update close

Figure 19, Additional simulation parameter window

The additional simulation parameters (Figure 19) can be invoked by the "additional" button of the main window (Figure 7). This dialog can be used to select features of the SIMBAA core and other model related options.

In the upper panel the **kernels** for "dispersal distance" and "dispersal patch size" can be selected. Note that this selection is global and applies to all species. See chapter "dispersal" for more details.

1.1.1.17 Render priority

The button "**render priority**" on the upper right opens a dialog to change the priority of the SIMBAA GUI. Note that this option influences the multitasking of the host system. If you want to run a simulation while doing other computing tasks, you may select "below normal" or "normal". However, this will result in fewer computation time (CPU time), thus slows down the simulation. To speed up the simulation you may select "above normal" or even "high" but this will degrade the performance of other programs (e.g. word processing) running simultaneous on the same machine.

1.1.1.18 Data separator

The panel "**data separator**" on the middle right allows to select the character used to separate data values when saving any data into text files, thus also the run log. This selection is stored in the SIMBAA grid file. To import SIMBAA data into a spreadsheet or statistical software, you commonly select to import text or "CSV" ("comma/colon separated values")/TAB data.

The both lower right panels are used to select the global used a- and b diversity measure. See chapter "Diversity Measurements available in SIMBAA" for more details and formulas.

The fields "**flow direction**" are remains of an older version and have no meaning (will be removed in a later version)

1.1.1.19 "Fluctuating reproduction"

If this switch is enabled, the number of larvae of a species is multiplied by a random value between "min" and "max".

A little example (a single dispersal event): number of individuals of a species in a cell: 4, fecundity: 2 → disperse 4 * 2 = 8 larvae of this species. Fluctuating reproduction enabled, min = 0.5 max = 2 → draw a random fluctuating factor between 0.5 and 2, let's say 1.4 This will result in 8 * 1.4 = 11.2, rounded 11 larvae to be dispersed. In an other cell, the factor may be 0.65 resulting in 8*0.65=5.2 rounded 5 larvae.

1.1.1.20 "start fill"

Every time you create a new simulation (by "new" in the main window), the whole simulation grid is cleared first. As initial population a random species assemblage is then created. This assemblage fills a strip of the grid from the left side (0/0) to the x-position "startfill" (startfill/ y-dimension). The initial population consists of individuals of random selected species (out off all available species) with their life history traits (e.g. age, last reproduction) randomly spread over their possible range. The cell capacity, e.g. all available space, is completely used.

1.1.1.21 "old reproduction interval"

This switch determines if a possible synchronisation of a species is based on simulation time step (old reproduction interval = **disabled**) or on the individuals age and time of last reproduction (default, old reproduction interval = **enabled**).

1.1.1.22 "Random seed"

This determines the initialisation of the random generator. Without additional hardware it is impossible to generate "real" random numbers with a computer. Thus any random number generated in SIMBAA is generated using a pseudorandom process[3]. Such a process starts with a "seed". Any value other than zero will result in a fix sequence of random numbers, making a simulation repeatable, e.g. the same simulation parameter will finally give the same results. A value of zero will start the random generator with a seed computed of the current time and data each time, thus the same parameter will give different results each run. For detailed information on the topic of random numbers and computers see the Internet (key words: pseudo random numbers")

1.1.1.23 "display radius"

This value is used to scale the size of an individual when displayed in the visualisation window.

1.6.1 Simulation stop conditions:

1.1.1.24 "max runtime"

This determines the runtime of the simulation. Any other value than zero will stop the simulation after "max runtime" time steps have been computed

[3] "A pseudorandom process is a process that appears random but is not. Pseudorandom sequences typically exhibit statistical randomness while being generated by an entirely deterministic computational process." (source: Wikipedia)

1.1.1.25 "+ N disturbances"

This determines how many disturbances must occur **after** the maximal runtime has occurred

1.1.1.26 "+ N time steps"

This is an opportunity to make the simulation run for some final time steps after the above criteria have been fulfilled.

In the shown example, the simulation will run for 5000 time steps, then proceed until (at least) one further disturbance event occurs and finally stop 10 steps after the time step when this happened.

Trick: to run the simulation until ***n*** disturbance events have occurred:

set "**max runtime = 1**" and "**+ N disturbances = *n***"

1.1.1.27 "repetitions"

This value is only used by the SIMBAA tools "Rechenknecht.exe" and "GUIKnecht.exe". It defines how many times the simulation is repeated. (Note: only meaningful when "random seed = 0")

1.1.1.28 “do biased lottery”/”neighbourhood size” and “k-factor”

These switches influence the lottery competition. Normally (do biased lottery=disabled), the lottery is strictly neutral and every larva in the pool has the same chance to win. If “do biased lottery” is enabled, the chance of a larva is influenced by the neighbourhood of the cell (see Formula 5).

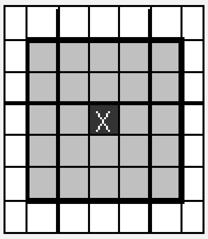
$$biased_S = pool_s * (1 + kfactor * w_{N,S})$$

$$p_s(win) = \frac{biased_S}{\sum biased}$$

Formula 5, Biased lottery description

The chance to win a lottery is determined by the relative proportion of a species in the biased pool. This is computed by the unbiased proportion (*pool$_S$*) weighted by *W$_{N,S}$*, the relative proportion of the species in the neighbourhood of the size *N*. The weight is modified by the *kfactor*.

Example of a biased lottery with 4 species



Neighbourhood size:	2 (grey shaded)
kfactor	2

Species	Unbiased *pool$_S$* *p$_S$(win)*		Neighbourhood (*W$_{N,S}$*)	*biased$_S$* pool
A	0.20	0.20	0.280	0.199
B	0.34	0.00	0.340	0.241
C	0.16	0.54	0.333	0.236
D	0.30	0.26	0.456	0.324
Σ	1.00	1.00	1.409	1.000

Figure 20 shows the influence of a kfactor between -2 and +2 on the winning chance of the species from the above example:

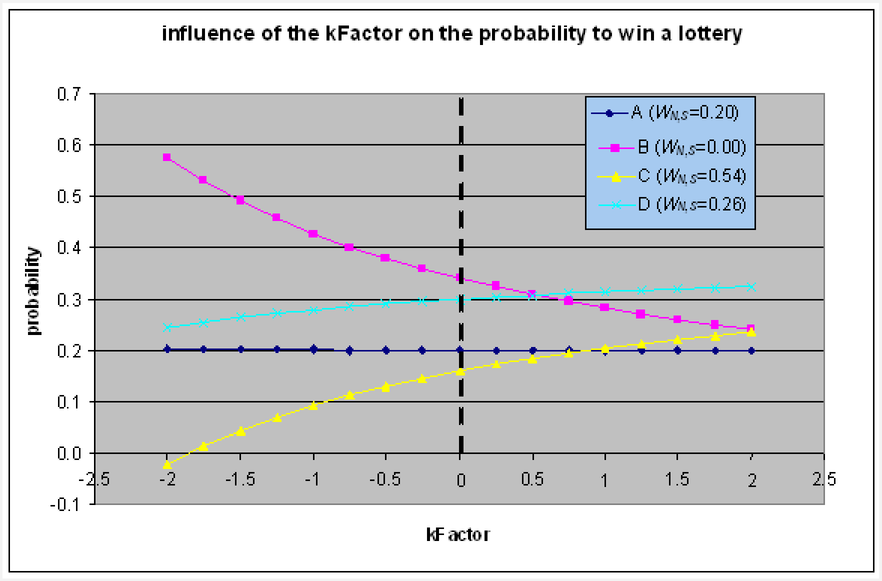


Figure 20, Influence of the kfactor on the lottery

1.7 Diversity Measurements available in SIMBAA

1.7.1 α-Diversity

These indices measure α- or point diversity. In SIMBAA the computation is based on a list of individuals. Typically this list is the inventory of a single sample, e.g. a simulation grid cell or a cluster of cells.

In SIMBAA the following α-diversity are available:

1.1.1.29 Shannon Index (Shannon Entropy)

$$H' = -\sum p_i * \log(p_i)$$

1.1.1.30 Simpson's Index (reciprocal)

$$D = \sum p_i^2 \qquad or \qquad \frac{1}{D}$$

1.1.1.31 Hill Numbers

The Hill Numbers form a group of diversity measurements. They are based on the *Rényi* entropy where α is the order, S is the sample size i.e. species count, p_i the relative proportion of the ith species:

Rényi entropy $$H_\alpha = \frac{1}{1-\alpha}\log\sum_{i=1}^{S} p_i^\alpha$$

Mark Hill proposed using $N_a = \exp(H_a)$. Thus N_a is the "Hill number". Although *a* (respectively α) can be any number (and SIMBAA allows to compute it), some have a common interpretation:

N_0 = number of species
N_1= exponential Shannon Index
N_2= inverse Simpson Index

1.1.1.32 M-Index

The M-Index is a special measurement and was designed during this thesis. It does not compute traditional diversity but allows to order an assemblage according to the dominance of its members, e.g. “pioneer dominated” or “climax dominated”. It is descript in an own chapter.

1.7.2 β-Diversity

The concept of *β*-diversity is sometimes not well defined. In SIMBAA *β*-diversity measures the similarity (respectively complementarily) of two assemblages. Following *β*-diversity functions are available:

(In the following examples let *a* be the total number of species in both samples, *b* number of species in the first sample and *c* the number of species in the second sample)

1.1.1.33 Sørensen Index

$$C_S = \frac{2a}{2a+b+c}$$

1.1.1.34 Jaccard Index

$$C_J = \frac{a}{a+b+c}$$

1.1.1.35 Marczewski-Steinhaus Distance

$$C_{MS} = 1 - \frac{a}{a+b+c}$$

1.8 The M-Index

The M-Index is a simple, dimension-less index for analysing the configuration of hierarchical species communities. If no hierarchy is detected, the M-Index is equal to 0. Other extreme values (M-Index = -1 or M-Index = 1) occur when the composition consists only of members of the respective end of the hierarchy. For other configurations the M-Index takes values between -1 and +1 that reflect the skewness of the community, e.g. it gives an indication to which end of the hierarchy the community is more developed.

For each possible species within the community a hierarchy value *OR* (Objective Rank) must be defined first. This value *OR* characterises the position or relevance of a species in the hierarchy. It can be simply defined by the rank order relation of the species as:

$$OR_i = -1 + \left[(2 * \frac{(r_i - 1)}{(R - 1)} \right]$$

where O*R* is the hierarchy-value of the ith species and r_i its rank among the *R* possible hierarchy ranks. Using the about formula the species hierarchy-value OR_i is ranked from -1 to +1 with equal distances from rank to rank. However, OR_i -values may be also assigned „by hand“, enabling a defined hierarchy if desired. In general, a value of $OR_I = 0$ means that the (ith) species is not relevant for the hierarchy whereas $OR_I = -1$ and $OR_i = +1$ represent the lower, respectively upper end of the hierarchy.

Species abundances A_i may be log+1 transformed and normalised by the sum of the log+1 transformed abundances. This log+1 transformation is used to down-weight high abundances. Of course, any other transformation may be used or the transformation can be omitted completely, using only the relative species proportions. In this case (using untransformed relative species abundances) the final index should be called “M-Index$_0$”). This gives the relative species quotient q_i for each species.

$$qi = \frac{\log(1+A_i)}{\sum \log(1+A_i)}$$

or simply $$qi = \frac{A_i}{\sum A_i}$$

The M-Index can be computed as the sum of q_i weighted by OR_i:

$$M - Index = \sum OR_i * q_i$$

1.8.1 M-Index, an example:

Assume a collection of 8 species. This 8 species can be grouped into 5 bins, e.g. based on their occurrence in a succession after disturbance events where the rank 1 attributes a pioneer species and 5 a climax member:

species	A	B	C	D	E	F	G	H
rank	1	1	1	2	2	3	3	4
OR_i	-1.00	-1.00	-1.00	-0.33	-0.33	0.33	0.33	1.00

Now these communities are sampled at 5 stations:

sample	absolute abundance							
sample 1	100	25	0	0	0	0	0	0
sample 2	0	0	0	0	0	0	0	100
sample 3	100	100	100	100	100	100	100	100
sample 4	50	10	100	0	12	137	10	5
sample 5	5	50	0	10	10	100	137	12

Applying the above transformations [2] this results in the following q_i-values

sample	q_i							
sample 1	0.586	0.414	0.000	0.000	0.000	0.000	0.000	0.000
sample 2	0.000	0.000	0.000	0.000	0.000	0.000	0.000	1.000
sample 3	0.125	0.125	0.125	0.125	0.125	0.125	0.125	0.125
sample 4	0.174	0.106	0.204	0.000	0.113	0.218	0.106	0.079
sample 5	0.079	0.174	0.000	0.106	0.106	0.204	0.218	0.113

The relative abundances ([2a]) are

sampe	q_i							
sample 1	0.80	0.20	0.00	0.00	0.00	0.00	0.00	0.00
sample 2	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00
sample 3	0.13	0.13	0.13	0.13	0.13	0.13	0.13	0.13
sample 4	0.15	0.03	0.31	0.00	0.04	0.42	0.03	0.02
sample 5	0.02	0.15	0.00	0.03	0.03	0.31	0.42	0.04

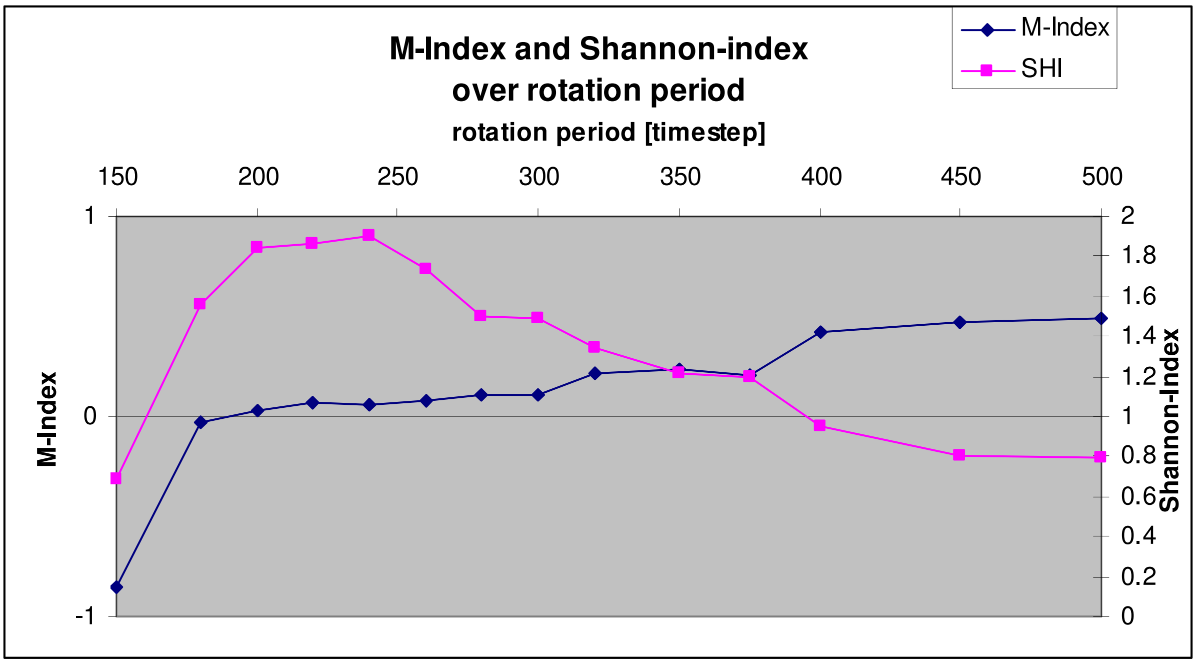
Calculating the M-Index finally gives:

sample	M-Index	M-Index$_0$	Shannon
sample 1	**-1.00**	-1.00	0.50
sample 2	**1.00**	1.00	0.00
sample 3	**-0.25**	-0.25	2.08
sample 4	**-0.33**	-0.34	1.42
sample 5	**-0.07**	0.09	1.42

As expected, the sample 1 and 2, consisting only of pioneer –respectively climax-species are just mapped on their corresponding ends of the given hierarchy. Together with the Shannon-Index a more complete picture of the community state can be drawn: e.g. the Shannon-index of sample 1 (SHI = 0.50) indicates that sample 1 consists of several species whereas sample 2 (SHI = 0) has only one counting species. Although the Shannon-index of sample 3 represents the maximal expectable Shannon-index (all species have the same relative proportions), thus indicating the highest diversity, the M-Index clearly indicates a skewed community, in this case (because pioneer species or early successional stages, have been assigned the negative end of the rank), towards early succession. However, a value of M-Index = -0.25 represents an advanced succession state rather than the very first beginning. Sample 4 and sample 5 share the same Shannon-index of SHI = 1.42 but their M-Index clearly separates them into an early (sample 4: M-Index = -0.33) and a rather balanced (sample 5: M-Index = -0.07) stage.

An other example from a simulation study about the influence of disturbance events on the community of a model system. The species where again grouped according to their succession potential, ranging from -1 for pioneer species (species group R0) to +1 for climax species (species group UD). Aim of this study was to qualify the influence of different disturbance regimes towards the final community stage. The disturbance regime can be characterised by the rotation period RT, which is the time needed to statistically disturb the whole simulation area once. It is depending on disturbance area and disturbance frequency. As disturbance area was kept constant, the ascending RT represents a descending disturbance frequency. Simulations started with a random community and ran for 10000 time steps. The following table summarises the simulation results. Given are species abundance and both Shannon- and M-Index.

RT	**Abundance**								**SHI**	M-Index
	R01	**R02**	**R11**	**R12**	**R21**	**R22**	**UD1**	**UD2**		
150	76238	74253	0	0	0	14	0	0	0.69	-0.86
180	23246	23972	1853	1995	51852	42243	2908	2436	1.56	-0.03
200	5779	7195	10775	14215	41934	41242	11158	18207	1.84	0.03
220	2842	2852	15074	17571	25262	29033	19672	38199	1.86	0.06
240	3358	6030	16174	17116	23362	15010	33814	35637	1.90	0.05
260	2159	3456	9629	17067	15206	13788	41282	47759	1.74	0.08
280	1108	1749	9223	6375	11861	9162	58580	52358	1.50	0.10
300	1188	1195	6776	9073	9104	11913	54354	56709	1.49	0.11
320	2072	0	6339	8774	6173	5531	58177	63411	1.34	0.22
350	947	0	6979	7803	6680	781	62078	65072	1.22	0.23
375	5588	0	3530	4810	2641	3436	68092	62378	1.20	0.21
400	0	0	6406	4583	0	0	68365	70917	0.95	0.42
450	0	0	2249	1232	0	0	75074	71561	0.80	0.47
500	0	0	2682	413	0	0	70614	76526	0.79	0.49



These pictures represent the final model state after 10000 time steps. Drawn is the succession stage (red = pioneer / R0-group, yellow = early settler / R1-group, green= late settler/R2-group, blue = climax / UD-group) based on the dominant succession potential at a particular location.

You can easily see that with RT = 150 only the R0 (=pioneer) species survive whereas with longer RT the R0-group (red) and, interestingly, the R2-group (late settlers, green) vanish. The visual impression that the model shows between RT = 200 and RT = 260 the most diverse (mixed) cases are confirmed by both a high Shannon-index and a: M-Index near zero. In general, the: M-Index ascends with ascending rotation period, indicating a shift towards dominance by later succession stages, supporting the visual results. The: M-Index can be interpreted as 4 different situations, separated by distinct levels in the curve. The first consists only of RT = 150 and has a: M-Index of: M-Index = -0.86, showing a high dominance of pioneer species. The abundance data reveal that only very few other individuals (14 individuals of R22) prevent the: M-Index from becoming its extreme value. The next level near zero indicates a well mixed or better not clearly dominated community. However, with ascending rotation period there is a slight shift towards more climax-dominated communities. The next level (RT= 320-375) with a: M-Index around +0.2 is clearly climax dominated. The main reason for this is the loss of a complete R0-species ($R0_2$), which seems to coincide with higher abundance in the UD-group (see abundance table). The last level (RT > 375, M-Index around +0.45) is caused by the loss of the complete R0- as well as the R2-group. The: `M`-Index value is underlined by the visual impression of the dominant UD-group.

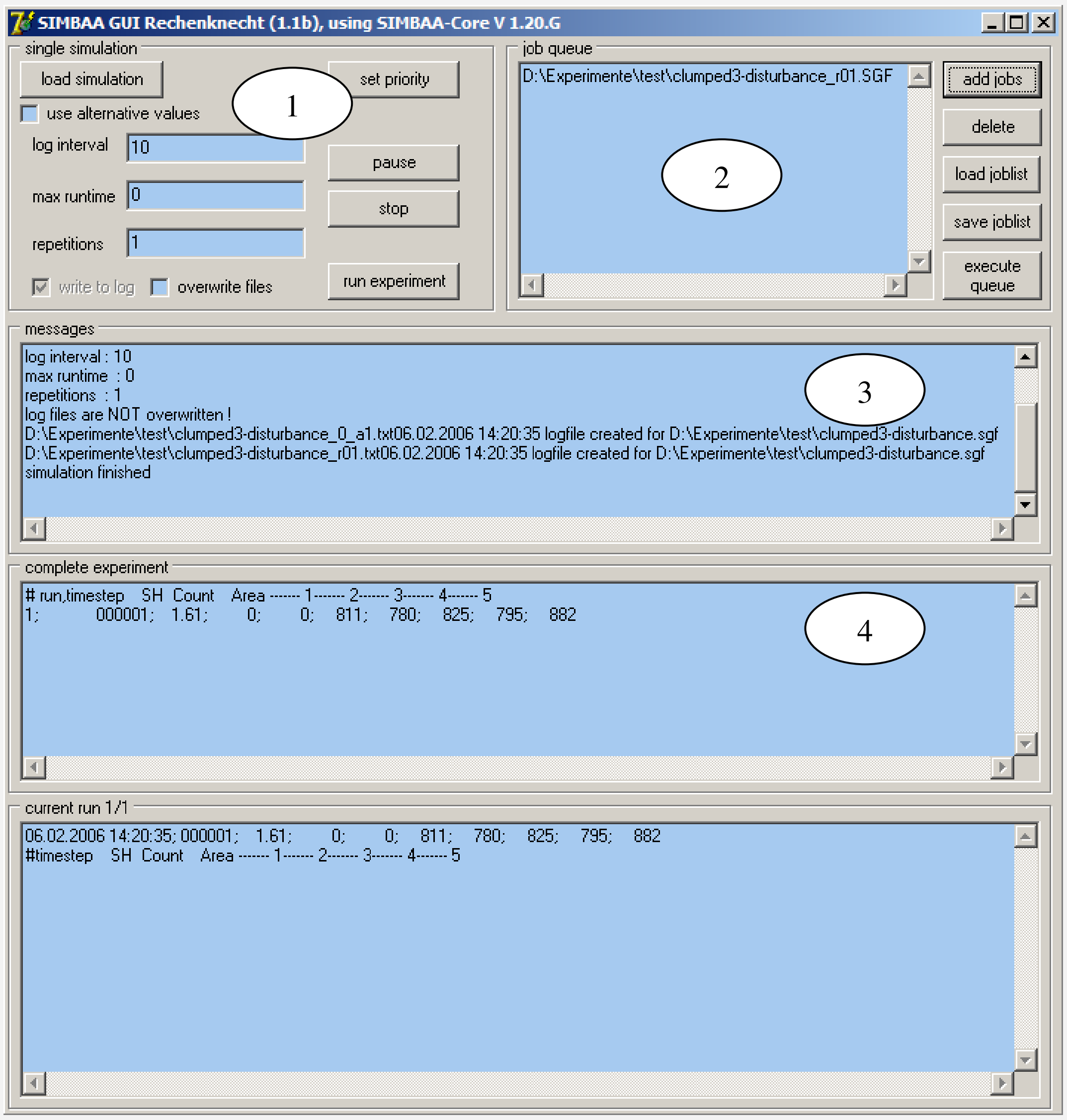
1.9 General SIMBAA tips

All graphs are capable to be saved or copied to the clipboard. A simple double click on the graph will open a save dialog. Graphics are saved by default as enhanced windows meta file (*.emf/ *.wmf). By holding down the left SHIFT-Key, the graphic is copied into the clipboard.

SIMBAA Tools

SIMBAA has some additional tools. Most useful are „Rechenknecht.exe” and “GuiKnecht.exe”. Both programs can load pre-configured simulations and just compute them.

7



SIMBAA GUI Rechenknecht (1.1b), using SIMBAA-Core V 1.20.G

[ ] use alternative values
log interval: 10
max runtime: 0
repetitions: 1
[✓] write to log [ ] overwrite files

```
D:\Experimente\test\clumped3-disturbance_r01.SGF
```

```
log interval : 10
max runtime  : 0
repetitions  : 1
log files are NOT overwritten !
D:\Experimente\test\clumped3-disturbance_0_a1.txt06.02.2006 14:20:35 logfile created for D:\Experimente\test\clumped3-disturbance.sgf
D:\Experimente\test\clumped3-disturbance_r01.txt06.02.2006 14:20:35 logfile created for D:\Experimente\test\clumped3-disturbance.sgf
simulation finished
```

```
# run,timestep   SH  Count    Area ------- 1------- 2------- 3------- 4------- 5
1;          000001;   1.61;       0;       0;    811;    780;    825;    795;    882
```

```
06.02.2006 14:20:35; 000001;   1.61;       0;       0;    811;    780;    825;    795;    882
#timestep   SH  Count   Area ------- 1------- 2------- 3------- 4------- 5
```

Figure 21, The SIMBAA tool “GuiKnecht.exe”

Figure 21 shows the window of the “GuiKnecht”. You can either load a single simulation of a list of simulations (2) or process this list in a batch mode. All options are normally read from the simulation but it is possible to override some options by changing the values in (1). “execute queue” tries to load and execute every entry in the job list. In the upper panel (3) status information are listed, the middle panel (2) lists the final simulation result or the final results of an each repetition. The bottom panel (3) list the status of the current running simulation.

Rechenknecht.exe is a command line tool that can be used to run a simulation from a command line. This is mainly useful when simulation is done on a remote machine.

An other very useful tool is “Replaceproject.exe”. This is also a command line utility for manipulating various aspects of a simulation. The following screen appears when started with no parameters:

```
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\>replaceproject
SIMBAA ReplaceParameter, using SIMBAA-Core V 1.20
no parameter!!!

ReplaceProject [datafile] KEYWORD [newValues]


accepted keywords are:

    FLOWDIR : set flow direction to specified angle [deg]
    FLOWDEV : set flow deviation to specified range [deg]
    RESETEX : reset species extinction times
   REPETITION : set repetition count to N
    RUNTIME : set max runtime to N
  SPECIESPOOL : replace species pool with pool from file
    STATEDEF : replace sucession state definition with definition from file
    MINSIZE : replace the "min-size-for-visual" with new value
   FLOWSPEED : set flow speed to specified range
    RANDSEED : pseudo random generator seed (0=use randomize)
```

A very useful batch file to call program with a set of parameters where the first parameter is always a filename out of a list of all files matching a given filemask is:

“forall.bat”

```
@Echo OFF
ECHO rekursiv durch alle directorys
IF "%1"=="" goto using
IF "%2"=="" goto using
FOR /R %%d in (%1) do %2 "%%d" "%3" "%4" "%5" "%6" "%7" "%8"
GOTO ende
:using
ECHO USAGE:
ECHO.
ECHO %0 FILEMASK PROGRAM
ECHO.
:ende
ECHO goodbye
```

An example to use this batchfile would be “c:\>forall *.sgf ReplaceProject runtime 1000”. This would result replacing the “runtime” of all SIMBAA grid files (*.sgf) in the current (and deeper directory(s) to be replaced with the value “1000”.

1.9.1 Appendix

1.1.1.36 File Format of the SIMBAA GRID FILE *.SGF

This is the file format for a simulation. It contains all information. The SGF is a binary file format. It is organised in different sections.

1.) HEADER
2.) STATE DEFINITIONS
3.) FLOW GRID DEFINITION
4.) DISTURBANCE DEFINITON
5.) SPECIES DEFINITION
6.) SIMULATION GRID DEFINITION

The first section in the file is a header with relevant information. All sections are descript below in detail. For easier access outside the SIMBAA environment, both type and size of the data field is listed along with its offset in bytes from the beginning of the structure. Sometimes the data fields are aligned by the compiler in a way that there are spare bytes. This is indicated by the real size of this field in brackets. Thus e.g. a size of 1(4) means that the data field just uses the first byte but covers 4 bytes in total. The reason for this and the order of the fields is the growth and change of the structures during development of SIMBAA. Also the Delphi-style type definition is given.

1.9.2 SGF-Header

This header contains most information on the simulation.

Name	type	size [byte]	offset	explanation
ID	char	16	0	This field must contains `'SIMBAA GRID FILE'` and is used to identify a valid SGF file
Version	char	6	16	The version string (e.g. `'V 1.20'`)
DEXB1	byte	2	22	Reserved 2 byte
Species count	integer	4	24	Number of defined species
Xdim	integer	4	28	Grid dimension on x-axis
YDim	integer	4	32	Grid dimension on y axis
Disturbances	integer	4	36	Number of defined disturbances
DisturbedArea	integer	4	40	Cummulative amount of disturbed area
Timestep	integer	4	44	Current time step
DisturbanceCount	integer	4	48	Cummulative number of occurred disturbances
ExternalDriftCount	integer	4	52	Max Number of larvae for a external drift event
ExternalDriftEvent	double	8	56	probability for an external drift event
HiC	integer	4	64	capacity of a high capacity cell
LoC	integer	4	68	capacity of a low capacity cell
pHiC	double	8	72	probability for a high capacity cell
Flow dir	double	8	80	flow direction (unused)
Flow dev	double	8	88	flow deviation (unused)
Dietime	51 integers	204	96	List of the extinction time of all species (0=not extinct jet)
lasttAb	51 integers	204	300	abundance of all species in the last time step
Maxtime	integer	4	504	maximal runtime of simulation (0=unlimited)
Repetitions	integer	4	508	repeat simulation N times
Periodic	boolean	1	512	periodic boundary condition state
OnlyDrift	boolean	1	513	disable SIMBAA's explicit dispersal model
ProportionalDrift	boolean	1	514	dirft is proportional to speces abundance
DSCChar	char	1	515	data separating charachter, ASCII char used to separate data values
SeedSyncAgeClasses	boolean	1	516	synchronise the age of all individuals of a species based on reproduction interval

DEXB2	byte	3	517	reserved 3 byte
Vminsize	double	8	520	minimal size for visual
SubVersionChar	char	1	528	subversion identifier (e.g. `'D'`)
DispersalKernelF	byte	1	529	Bit-based dispersal kernel flag, upper nibble for "patch size kernel", lower nibble for "dispersal distance" \$01=exponentioal kernel, \$02=diffusion kernel, \$04=uniform kernel, \$08 reserved
DoBiasedLottery	byte	1	530	flag for biased lottery
kFactor	shortint	1	531	unused
SizeNeighbourhood	byte	1	532	neighbourhood size for biased lottery
DEXB3	byte	3	533	reserved 3 byte
kFaktor2	double	8	536	koppel faktor for biased lottery
RepFluc	boolean	1	544	flag for fluctuating reproduction
RepFlucMin	integer	4	545	flucutating reproduction minimal
RepFlucMax	integer	4	549	flucutating reproduction maximal
RandSeedValue	integer	4	553	random number seed
Reserved	byte	51	557	reserved

Delphi-type definition:

```
TSimFileHeader = packed RECORD
          ID : Array[1..16] of Char;
          version : Array [1..6] of Char;
          DEXB1 : ARRAY[0..1] of byte; // dummy extra bytes 1
          SpeciesCount, Xdim, Ydim, Disturbances,DisturbedArea,
          Timestep, DisturbanceCount,
          externalDriftcount : integer;
          externalDriftEvent : double;
          HiC,LoC : integer;
          pHiC : double;
          flowDir, FlowDev : double;
          dietime,lasttAb : TSpeciesList;
          maxtime, repetitions : longint;
          periodic : boolean;
          onlyGDrift,
          proporionalDrift : boolean;
          DSCChar : char;
          SeedSyncAgeClasses : boolean;
          DEXB2: ARRAY[0..2]of byte; // dumme extra bytes 2
          VminSize : double;
          SubVersionChar : char;
          DispersalKernelF : byte; // 0000-0000 patch & distance kernel
          DoBiasedLottery : byte;
          KFaktor : shortInt;
          SizeNeighbourHood : byte;
          DEXB3 : ARRAY[0..2] of byte; // dummy extra bytes 3
          kFaktor2 : double;
```

```
     RepFluc : boolean;
     RepFlucMin, RepFlucMax : integer;
     RandSeedValue : integer;
     RESERVED : ARRAY[0..100-sumofChange] of byte;
    END;
```

1.9.3 State definitions

The state definitions is a four element list holding the definitions for each of the states R0,R1,R2 and UD. Each definition itself is a 5 element list. Each element of this list holds the information on a particular guild (if used, min, max proportions)

TTransStateDef

Name	type	size [byte]	offset	explanation
R0	TStateDef	120	0	state definition
R1	TStateDef	120	120	
R2	TStateDef	120	240	
UD	TStateDef	120	360	
total size		**480**		

TStateDef

Name	type	size [byte]	offset	explanation
undefined	TStateGuildeDef	24	0	definition of the min/max and if used
R0	TStateGuildeDef	24	24	
R1	TStateGuildeDef	24	48	
R2	TStateGuildeDef	24	72	
UD	TStateGuildeDef	24	96	
total size		**120**		

TStateGuildeDef

Name	type	size [byte]	offset	explanation
useThisGuilde	boolean	1(8)	0	if this guild is essential for the state
relativeMin	double	8	8	min. rel. proportion of individuals
relativeMax	double	8	16	max rel. proportion of individuals
total size		**24**		

Delphi-type definition:

```
TTransState = (undefined,R0,R1,R2,UD);

TStateGuildeDef = RECORD
           UseThisGuilde : boolean;
           relativeMin,relativeMax : double;
         END;
TStateDef = ARRAY[0..nGuilds] of TStateGuildeDef;
TTransStateDef = ARRAY [R0..UD] of TStateDef;
```

1.9.4 Flow Grid definition

The flow grid contains all information about the flow grid.

Name	type	size [byte]	offset	explanation
FlowGridXDim	integer	4	0	flow cells on x-axis
FlowGridYDim	integer	4	4	flow cells on y-axis
FlowData	array of TlocalFlowDef	X*Y*24	8	map of local flow definition savin scheme: (0/0),(0/1),(0/2).... (1/0),(1/1),(1/2).... .... (x/0),(x/1).....(x/y)
total size		**various**		

TLocalFlowDef

Name	type	size [byte]	offset	explanation
lFlowDirection	double	8	0	flow direction
lFlowDeviation	double	8	8	flow deviation
lFlowSpeed	double	8	16	flow speed
total size		**24**		

```
// definition of flow grid
TLocalFlowDev = RECORD
         lFlowDirection,
         lFlowDeviation,
         lFlowSpeed : float;
        END;
TFlowGrid = RECORD
       FlowGridXDim,FlowGridYDim : integer;
       FlowData : ARRAY of ARRAY of TLocalFlowDev;
      END;
```

1.9.5 Disturbance definitions

Name	type	size [byte]	offset	explanation
Disturbances	TDisturbaceDef	N*112		a list of all disturbance definitions
total size		**various**		

TDisturbanceDef

Name	type	size [byte]	offset	explanation
xleft	integer	4	0	Disturbance area definition
ytop	integer	4	4	
xright	integer	4	8	
ybottom	integer	4	12	
xmean	double	8	16	Disturbance size definition
xstd	double	8	24	
ymean	double	8	32	
ystd	double	8	40	
clipping	**boolean**	1(8)	48	Clipping enabled
probabiltiy	double	8	56	probability per time step
lasttime	integer	4	64	Last occurence
serverity	double	8	68	0-100 %
changeSubstrate	**byte**	1(8)	**80**	Binary switch S_1-S_8
DisturbanceDirection	double	8	88	Direction
disturbanceDirDeviation	double	8	96	Deviation
subDisturbanceEvents	integer	4(8)	104	Number of sub events
total size		**112**		

Delphi type definition

```
TDisturbance = Record
         xleft, ytop,         // area def
         xright, ybottom : integer;

         xmean, xstd,      // mean size and standard deviation
         ymean, ystd : float;

         clipping : boolean; // clip at area border
         probability : TSubTimeValues;
         Lasttime : integer;

         // changed in v1.2
         severity : float;        // 0 - 1.0 ==0-100%
         ChangeSubstrate : bytE;      // reset subtrate to
         DisturbanceDirection,       // direction 0..2Pi
         DisturbanceDirDeviation : float; // deviation 0..2Pi
```

```
  SubDisturbanceEvents : integer; // do x sunbevents...

End;
```

1.9.6 Species definitions

A linear list of all defined species

Name	type	size [byte]	offset	explanation
species	TSpeciesDef	N*496		a list of all species
total size		**various**		

TSpeciesDef

Name	type	size [byte]	offset	explanation
Name	string[20]	20	0	Name
dispersalDistance	double	8	24	Dispersal distance
dispersalPatchSize	double	8	32	Dispersal patch size
MaxSize	double	8	40	Max size
deathProbability	double	8	48	Mortality per time step
Fecundity	double	8	56	Fecundity per reproduction
maxLifeSpan	integer	4	64	Maximal life span
meanLifeSpan	integer	4	68	„normal“ life span
firstReproduction	integer	4	72	Age of marternity
ReproductionInterval	integer	4	76	Reproduction interval
HasGlobalDrift	boolean	1	80	Is capable of migration
belongsToState	byte	1(4)	81	Belongs to guild
GrowsOnSubstrate	**integer**	4	**84**	Binary substrate S_1-S_8
ChangesToSubstrate	integer	4(8)	88	Binary substrate S_1-S_8
degTimeSync	double	8	**96**	Proability to be in reproduction syncronisation
DisplayColor	integer	4	104	Display color, RGB-value
GrowthModel	integer	4	108	0=linear 1=exponential
growthK	double	8	112	Exponential growth constant
RESERVED	byte	373(376)	120	
total size		**496**		

```
  TSpeciesDef = RECORD
           name : string[20];
           dispersalDistance,
           dispersalPatchSize,
           maxSize      : float;
           deathprobability,
           fecundity     : TSubTimeValues;
           maxLifeSpan,
           meanLifeSpan,
           firstReproduction,
           ReproductionInterval : integer;
           HasGlobalDrift : boolean;
           belongsToState : TState;
           GrowsOnSubstrate,
           ChangesToSubstrate : integer;
           degTimeSync : float;
           DisplayColor : integer; // 4 byte = TColor;
           GrowthModel : integer;
           growthK : float;
           RESERVED    :    ARRAY[0..4*99-(2*sizeOf(float)+2*sizeOf(integer))]    of    byte;
//integer=4Byte!
         END;
```

1.9.7 Simulation grid definition

This is a map of all simulation grid cells. The cells are stored consecutive in the following order:

```
(0/0),(0/1),(0/2)....
(1/0),(1/1),(1/2)....
....
(x/0),(x/1).....(x/y)
```

Each cell has a cell is stored with a header, containing the cell info and a list of all individuals in this cell. The data types are listed below:

Name	type	size [byte]	offset	explanation
Cell Header	TCellHeader	20	0	cell definition
Individuals	TIndividual	N*32	20	list of all individuals in the cell
total size		**various**		

TCellHeader

Name	type	size [byte]	offset	explanation
capacity	integer	4	0	cell capacity
totalDist	integer	4	4	total number of disturbances
lastDisturbance	integer	4	8	time since last disturbance
SubstrateType	integer	4	12	substrate S1-S8
Individuals	integer	4	16	number of individuals in the cell
total size		**20**		

TIndividual

Name	type	size [byte]	offset	explanation
isSpecies	integer	4	0	is of species Nr.
age	integer	4	4	age of the individual
size	double	8	8	current size
lastReproductio n	integer	4	16	time steps since last reproduction
xPos	integer	4	20	sub-grid x-position
yPos	integer	4(8)	24	sub-grid y-position
total size		**32**		

Delphi-style defintions:

```
TFileCellHeader = RECORD
        capacity,
        totalDist,
        lastDisturbance,
        SubstrateType,
        Individuals : integer;
       END;

 TIndividual = RECORD
         isSpecies    : integer;
         age       : integer;
         size       : float;
         lastreproduction : integer;
         xPos,yPos    : integer; // pos in subgrid;
        END;
```

1.10 Acknowledgement, external code

SIMBAA was completely written from the scratch using Borland Delphi 7 Professional. It makes extensive use of some packages supplied by Borland. However, some code was taken and modified from other sources:

The pseudo random generator procedures in the unit MyRandom.pas were taken form "Numerical Recipes in Pascal: The Art of Scientific Computing" (Press, Teukolsky, Vetterling and Flannery, Cambridge University Press, ISBN 0-521-37516-9). This was done in order to use a defined portable pseudo random generator.

The colour routines are based on source of *Grahame Marsh*, released as freeware:

```
//----------------------------------------------------------------------------
//
// HSL - RGB colour model conversions
//
// These four functions can be used to convert between the RGB and HSL colour
// models. RGB values are represented using the 0-255 Windows convention and
// always encapsulated in a TColor 32 bit value. HSL values are available as
// either 0 to 1 floating point (double) values or as a 0 to a defined integer
// value. The colour common dialog box uses 0 to 240 by example.
//
// The code is based on that found (in C) on:
//
//  http:/www.r2m.com/win-developer-faq/graphics/8.html
//
// Grahame Marsh 12 October 1997
//
// Freeware - you get it for free, I take nothing, I make no promises!
//
// Please feel free to contact me: grahame.s.marsh@corp.courtaulds.co.uk
//
// Revison History:
//  Version 1.00 - initial release 12-10-1997
//
//----------------------------------------------------------------------------
```

(copyright notice of the HSL-RGB source code in unit HSLUtils.pas)

The function to read the compile time of the executable (About-Dialog) was taken from:
http://www.delphipraxis.net/topic13233_datum+und+uhrzeit+der+kompilierung+compile+date+time.html
SIMBAA may contain other code parts inspired by information found at various places of the internet. In particular these parts address one specific problems and solutions for these, such as the above mentioned colour conversion and portable random generator. However, these are not essential for the simulation itself but made the work much easier (Man muss das Rad nicht mehrfach erfinden!).