

1 Microxanox: an R package for simulating an aquatic
2 *MICRobial* ecosystem that can occupy *OXic* or
3 *ANOXic* states.

4 Rainer M Krug^{a,1}, Owen L. Petchey^a

^a*Department of Evolutionary Biology and Environmental Studies Winterthurerstrasse 190
8057 Zurich*

5 **Abstract**

Microxanox is an R package to simulate a three functional group ecosystem (cyanobacteria, phototrophic sulfur bacteria, and sulfate-reducing bacteria) with four chemical substrates (phosphorus, oxygen, reduced sulfur, and oxidized sulfur) using a set of ordinary differential equations. Simulations can be run individually or over a parameter range. The model can be implemented with different numbers of species per functional group. The package is constructed in such a way that the results contain the input parameter used, so that a saved results can be loaded again and the simulation be repeated. Furthermore, the package framework and code should serve as a useful starting point for making simulation models of other types of ecosystems.

6 *Keywords:* reproducibility, regime shift, final state, ordinary differential
7 equations

*Corresponding author

Email addresses: Rainer.Krug@uzh.ch; Rainer@krugs.de (Rainer M Krug),
Owen.Petchey@ieu.uzh.ch (Owen L. Petchey)

¹Corresponding Author

²Equal contribution

8 1. Required Metadata

9 1.1. Current code version

10 Ancillary data table required for subversion of the codebase.

Nr.	Code metadata description	Please fill in this column
C1	Current code version	v0.9.1
C2	Permanent links to code/repository used for this code version	https://github.com/UZH-PEG/microxanox 10.5281/zenodo.7148667
C3	Code Ocean compute capsule	
C4	Legal Code License	MIT
C5	Code versioning system used	git
C6	Software code languages, tools, and services used	R
C7	Compilation requirements, operating environments	R (>= 4.1.0) magrittr tibble ggplot2 patchwork grDevices stats mgcv deSolve dplyr tidyr stringr multidplyr
C8	If available Link to developer documentation/manual	User Guide
C9	Support email for questions	Rainer.Krug@uzh.ch ; Owen.Petchey@ieu.uzh

11 2. Motivation and significance

12 Mathematical models play a key role in the development of ecosystem models
13 and in increasing the understanding of how and why ecosystems change when the
14 environment changes [1, 3, 17]. They are also important for developing hypotheses

15 to test in empirical studies. One area of ecology in which models have been
16 influential is the understanding of ecosystem responses to gradual change in an
17 environmental driver [12]. An environmental driver is an environmental condition
18 that affects an ecosystem, but which is not itself affected by the ecosystem. An
19 example driver would be the amount of nutrients entering a lake from run-off
20 from surrounding agricultural land.

21 It is conceivable that an ecosystem state, such as the total biomass of a
22 particular type of bacteria, may remain unchanged when an environmental driver
23 changes. It is also possible that the ecosystem state changes gradually. It is
24 also possible that the ecosystem state changes abruptly to a new state that is
25 difficult to recover from [12]. This possibility for abrupt, perhaps catastrophic
26 changes, which are difficult to reverse, causes considerable concern [4, 9, 16].

27 An example where a gradual change of an environmental variable causes an
28 abrupt change in the system is the switch from an aerobic (oxygen is available for
29 metabolism) to an anaerobic (oxygen generally unavailable) state in a microbial
30 ecosystem. This system has been investigated by Bush et al. [2] in a simulation
31 study of a mathematical model. Three types of microbes occur in the model:
32 cyanobacteria (CB) dominating the oxic state, and two types of sulfur bacteria
33 that dominate the anoxic state (sulfate reducing bacteria (SB) and phototrophic
34 sulfur bacteria (PB)). The model shows that gradual change in the rate at which
35 oxygen can diffuse into the ecosystem (termed oxygen diffusivity) could cause
36 catastrophic changes in the ecosystem state that would be difficult to reverse.

37 One feature of the study by Bush et al. [2] was limited biodiversity. Specifically,
38 there was no biodiversity within each of the three types (i.e., functional groups)
39 of bacteria. This leaves the question of if and how biodiversity within these
40 functional groups might affect the ecosystem response to environmental change.
41 This limitation is not specific to the study of Bush et al. [2]. There are few if any
42 studies of the effects of biodiversity on abrupt transitions between ecosystem
43 states.

44 We decided to fill this research gap by making a simulation study of how
45 within functional group biodiversity affects ecosystem responses to environmental
46 change Limberger et al. [8], and to base our work on the work and model of
47 Bush et al. [2]. With this goal in mind, we developed the `microxanox` package
48 [7]. The first stage of development was to write code from scratch (as there was
49 no available code to start from) and to confirm that this new implementation
50 would reproduce the previously published results. The resulting reproduction is
51 available as one of the package vignettes: [vignette Partial reproduction of Bush](#)
52 [et al.](#)

53 The second stage was to add functionality that would be necessary to answer
54 our research question. Most importantly, we made it possible to have multiple
55 species of bacteria within each of the three functional groups, for the multiple
56 species to differ in their characteristics, and to vary the number of species and
57 amount of variability among them. We also added functionality that allowed:
58 temporally varying environmental conditions, the addition of random noise to
59 state variables, and immigration. In addition to the model itself, the package
60 includes some functions to analyse the results and visualize these to provide a

61 starting point for customized visualizations based on own requirements. The
62 basic and additional functionality is described in the package [User Guide](#).

63 3. Software description

64 The *microxanox* package is for simulating a three functional group system
65 (*CB*: cyanobacteria, *PB*: phototrophic sulfur bacteria, and *SB*: sulfate-reducing
66 bacteria) with four chemical substrates (*P*: phosphorus, *O*: oxygen, *SR*: reduced
67 sulfur, *SO*: oxidized sulfur). It includes feedback between organisms and biogeo-
68 chemical processes and is based on Bush et al. [2] (See Bush et al. [2] for a detailed
69 discussion of the model). At the core of the simulations is a set of ordinary
70 differential equations (specified in the function `bushplus_dynamic_model()`,
71 though this function need not be directly called). There are functions for run-
72 ning individual simulations and a set of simulations across, for example, a range
73 of environmental conditions.

74 To make the simulation run with multiple species per functional group, we
75 expressed different species' characteristics in the elements of vectors and matrices.
76 We also coded the ordinary differential equations in matrix form to use matrix
77 mathematics. In this way, we made it possible to run simulations with different
78 numbers of species without having to change the underlying code.

79 The package functions and code have a modular structure so that new
80 functionality can be easily added. E.g., temporally defined events of any type
81 could be specified. Further, all parameter values required to run a simulation
82 are stored in one object. Lastly, the general structure of the code should make it
83 straightforward to adapt the model to other similar systems (described in more
84 detail in the Impact section).

85 3.1. Software architecture

86 The framework used when writing this package aimed to maximise simplicity
87 for the user and to make it straightforward to reproduce results (see the supple-
88 ment [10] to Limberger et al. [8] for an example of how this is used). As such, all
89 the parameters needed to run a simulation or a set of simulations are contained
90 in a single object (which can easily be created using included functions). This
91 parameter object is given to a function that runs the simulations and returns the
92 results. The returned results object is identical to the parameter object but with
93 an additional slot named `results`, which contains the simulation results. Thus
94 the returned results object contains the simulation conditions (parameters) as
95 well as the results, and can be used to run the simulation again. This promotes
96 reproducibility and makes incremental changes of individual parameters with a
97 consecutive re-running of the simulations straightforward.

98 In the following sections we describe how to use the package to run one
99 simulation and to find final states across an environmental gradient.



Figure 1: Typical flow of a simulation. Dark Grey boxes: commands necessary for simulation; Light Grey: Saving of parameter and results; Lightest Grey: Different non specified commands.

100 3.2. Running one simulation

101 A typical simulation would look as shown in Figure 1.

102 A simulation is run using the `run_simulation()` function. In this function,
103 the ODEs are solved using the function `ode()` in the package *deSolve* [14]. The
104 `run_simulation()` function needs only one argument - an object as created
105 by the function `new_runsim_parameter()`. The parameter object returned by
106 `new_runsim_parameter()` contains, among other things, the `strain_parameter`
107 object, which can be created by the function `new_strain_parameter()`. For
108 a detailed description of the parameter objects, their meaning and how they
109 are created and have values set and changed, please see the *User Guide* which
110 accompanies the package or is available at [User Guide](#).

111 After the parameter object has been defined, it can be used in the `run_simulation()`
112 function. The function returns an object identical to the parameter object, ex-
113 cept of an additional slot containing the results. This design produces a fully
114 reproducible object as it can be used as a parameter object to be fed back into the
115 `run_simulation()` function to rerun the simulation again from the parameter
116 used to generate the results before.

117 3.3. Examining ecosystem responses to environmental change

118 One approach to finding how the ecosystem responds to environmental driver
119 change is to examine the relationship between the environmental driver value and
120 the numerical value of a system's state. The first component of this approach is
121 to run the simulation for a long time and record the final state (i.e., the state of
122 the system at the end of a simulation). When one does this across a range of
123 environmental conditions, one discovers how the final state of the system changes
124 with the environmental conditions. The package contains functionality for finding
125 final states that correspond to values of one environmental driver, namely the
126 value of oxygen diffusivity. In terms of non-linear systems analysis, this would
127 be termed a *parametric sensitivity analysis* being conducted by running an open
128 loop of the dynamic system under a set of initial conditions and a subset of
129 parameter values (here oxygen diffusivity).

130 When one wishes to be able to make conclusions about how the *steady state*
131 (or an equilibrium point) of the system is affected by the environmental driver,
132 it is very important to note that the *final state* (provided by the simulation)
133 is not guaranteed to be a *steady state*. The software does not provide the user
134 with a steady state. In order to somewhat safely assume that the final state is a
135 steady state, the user must ensure that the simulation is run for sufficiently long
136 time for any transient dynamics to disappear, and must also check the type of
137 long-term dynamics occurring. In the results presented here, and in the paper
138 Limberger et al. [8], this was performed by visual inspection, and by checking
139 the sensitivity of conclusions to the length of the simulation. Furthermore, the
140 package does not include methods for a formal analysis of the stability of the
141 system and users should take care to assess if steady states are unique.

142 Two numerical strategies for finding final states and their sensitivity to
143 parameters are implemented. The first method runs a independent simulation

144 for each combination of initial conditions and oxygen diffusivity (we term this
145 the *Replication method*). This is the method used in the Bush et al. [2] study
146 and was used to obtain the results in figures 3 and 4 of that article.

147 The second method runs two simulations, one with step-wise and slowly
148 temporally *increasing* oxygen diffusivity, and the other with step-wise and slowly
149 *decreasing* oxygen diffusivity. (Put another way, to explore the sensitivity of
150 final states of the dynamic model under oxygen diffusivity variation, subset
151 values of this parameter are numerically evaluated. Two value vectors of oxygen
152 diffusivity are presented step-wise, gradually increasing and decreasing.) During
153 this temporal environmental change, the state of the system is recorded just
154 before change to a new oxygen diffusivity (we term this the *Temporal method*).

155 An potentially important difference between the two methods is in the system
156 state when a new value of oxygen diffusivity is set. In the replication method,
157 the system state when a new value of oxygen diffusivity is set is always the same.
158 Whereas in the temporal method, the system state when a new value of oxygen
159 diffusivity is set is the final state of the system for the previously set value of
160 oxygen diffusivity. Since some modellers prefer one approach and others another,
161 we decided to implement both.

162 The replication method is implemented in the function `run_replication_ssfind()`
163 which takes a parameter object as returned by the function `new_replication_ssfind_parameter()`
164 and the number of cores for multithreading the simulation. As the multithreading
165 uses the package function `mclapply()` from the package `parallel` [11], the mul-
166 titheading only works on Linux and Mac. It is planned to move to `parLapply()`
167 [11] in a future release.

168 The temporal method is implemented in the function `run_temporal_ssfind()`,
169 which takes a parameter object as created by the function `new_temporal_ssfind_parameter()`.
170 It is planned for a later release, to run these two simulations in parallel.

171 For a more detailed walk-through of these two approaches and explanation
172 please see the [User Guide](#).

173 3.4. Analysing and visualising results

174 From the results returned, summary measures about how the ecosystem
175 final states changes with environmental change can be extracted. The function
176 `get_stability_measures()` returns quantities such as the amount of environ-
177 mental change required for the system to abruptly change to a different state.

178 The function `plot_dynamics()` plots a single simulation run, as returned
179 from the `run_simulation()` function. This function is only provided as a
180 convenience function to provide a way to easily see the results of a simulation
181 run. An example plot resulting from this function is shown in Figure 2.

182 4. Impact

183 The open source implementation and extension of the model used in Bush
184 et al. [2] provides the means of reproducing the results published while at the
185 same time provides the means of making unique, innovative, and important

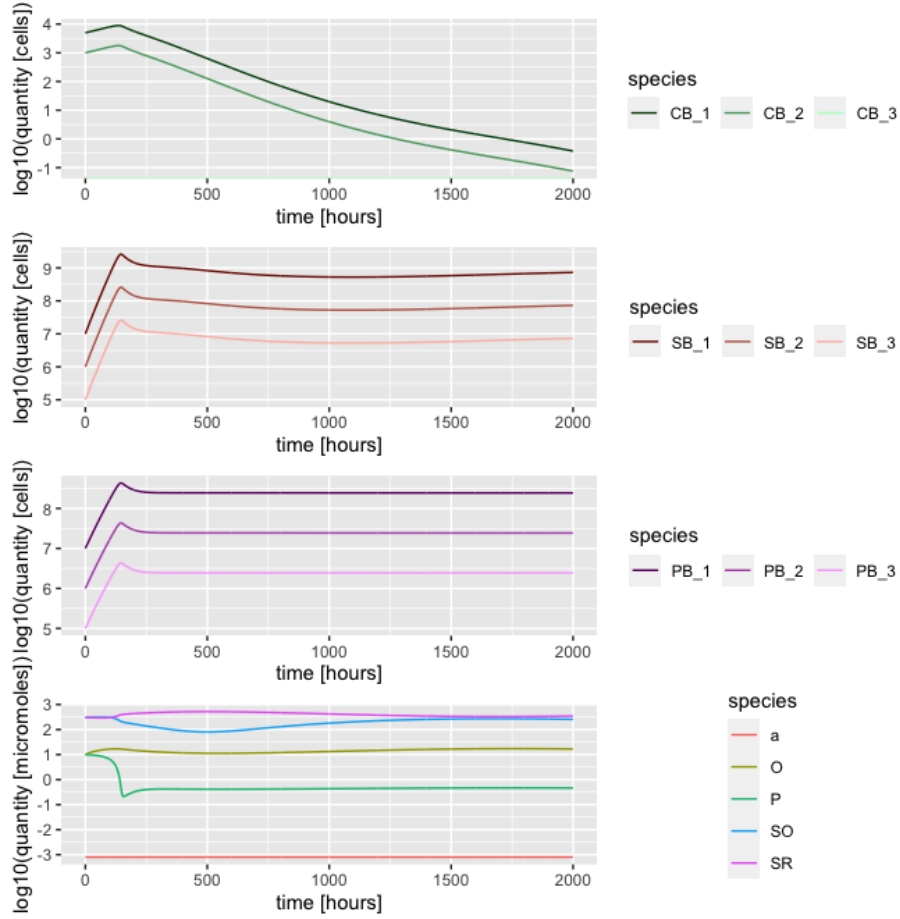


Figure 2: Results of a simulation run shown using the function `plot_dynamics()`. In this case, there were three strains per functional group, though strains within functional groups had identical properties in this example. CB_1 = cyanobacteria strain 1; SB_1 = sulfur reducing bacteria stain 1; PB_1 = phototrophic sulfur bacteria strain 1.

186 investigations of how ecosystems change as the environment changes, and how
187 biodiversity may modulate this.

188 The design of the package code and functionality is with reproducibility in
189 mind (for other example of reproducibility directed software, see e.g. Correndo
190 et al. [5], Spillner et al. [15] or Fox et al. [6]): the combination of all parameters
191 being in a single parameter object and the return of the simulation as a result
192 object which inherits the parameter values from the parameter object, provides
193 a relatively easy-to-use framework to implement reproducible experiments.

194 Here we evidence the impact of the *microxanox* package by describing three
195 use-cases and then by describing how the package can be a starting point for
196 models of other ecosystems. The first two use cases are described in detail
197 (including the code for reproducing them) in the [User Guide](#) and the *Partial*
198 *Reproduction* vignettes. The third is taken from Limberger et al. [8] and Petchey
199 et al. [10].

200 4.1. Use case 1: Regime shifts during temporal environmental change

201 The study of Bush et al. [2] includes simulations of the effect of oxygen
202 diffusivity (an environmental driver, in the sense that it affects the ecosystem but
203 is not affected by it) on the ecosystem state (oxic or anoxic). The *microxanox*
204 package contains functionality to make a specific temporal pattern of change
205 in the oxygen diffusivity. As well as allowing individual simulations during
206 which oxygen diffusivity varies, this functionality forms the basis of the temporal
207 method for finding final states.

208 An example of this functionality is given in the *Partial Reproduction* vignette,
209 which we briefly describe and show here (Figure 3). The example is composed of
210 a single simulation, at the beginning of which the system is in the oxic state with
211 high abundance of cyanobacteria. Oxygen diffusivity is then slowly decreased
212 and eventually, around hour 30'000 the system switches to the anoxic state, with
213 high abundance of both sulfur bacteria types. The oxygen diffusivity is then
214 increased and at around hour 38'000 the system abruptly switches back to the
215 oxic state.

216 Also visible in the results are thick lines showing abundances of bacteria
217 when abundances are low. This is due to the implementation of a function that
218 at regular intervals, increases the abundance to a preset level. This prevents
219 abundances reducing to very small numbers. The function that implements this
220 increase abundance can also be made to add a certain abundance to each strain
221 at regular intervals, thus simulating immigration in to the system.

222 4.2. Use case 2: The extent of hysteresis depends on community composition

223 The package contains a function to extract summary features of ecosystem
224 responses to environmental change, such as the amount of hysteresis displayed by
225 the ecosystem. Hysteresis is a key feature of ecosystem responses to environmental
226 change, because it is related to how difficult it can be to reverse the effects of
227 environmental change [12]. The amount of hysteresis is measured as the extent
228 of the environmental condition (here oxygen diffusivity) for which there were



Figure 3: The temporal dynamics of the ecosystem model when an environmental condition (here parameter a , the oxygen diffusivity) changes. Plot of the final states of the simulation runs under different oxygen diffusivity. In this simulation there is only one strain in each functional group. CB extunderscore1 = cyanobacteria strain 1; SB extunderscore1 = sulfur reducing bacteria strain 1; PB = phototrophic sulfur bacteria strain 1. Here we show a figure adapted from the output of the `plot_dynamics()` function.

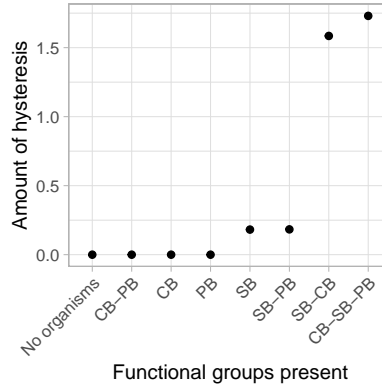


Figure 4: The amount of hysteresis depends on the combination of types of organisms present. The model is entirely deterministic, hence there are no error bars.

two distinct final states. I.e., it is the extent of the environmental conditions for which historical conditions play an important role in determining the current system state.

Using the package to calculate the extent of hysteresis involves setting ecosystem and simulation parameters, including parameters for the finding of final states across an environmental gradient, running the final state finding function, and analysing the results with the function that calculates extent of hysteresis. The code for this is provided in the [User Guide](#).

The results show that the amount of hysteresis depends greatly on the combinations of organisms present (Figure 4). For example, with only the CB (cyanobacteria) present, there was no hysteresis. In contrast, the presence of both CB and SB (sulfate reducing bacteria) led to a large amount of hysteresis. (These results are also given in the *Partial Reproduction* vignette.)

4.3. Use case 3: Effects of functional diversity on regime shifts

As discussed in the Introduction section, the package was motivated by the question of how biodiversity influences ecosystem responses to environmental change. Extensive results concerning this question are given in a separate publication Limberger et al. [8]. Here we describe one of the results, which is that having biodiversity in a functional group can allow state changes to occur that otherwise would not have, i.e. biodiversity can qualitatively change the state of the ecosystem.

Biodiversity was added to the functional groups using the `new_strain_parameter()` function to create a parameter set with multiple species per functional group (albeit all with identical features) and then to add variability among the species by calling the `add_strain_var()` function. This function takes an already existing parameter set and adds the specified amount of variation. The new parameter object is then used as before.

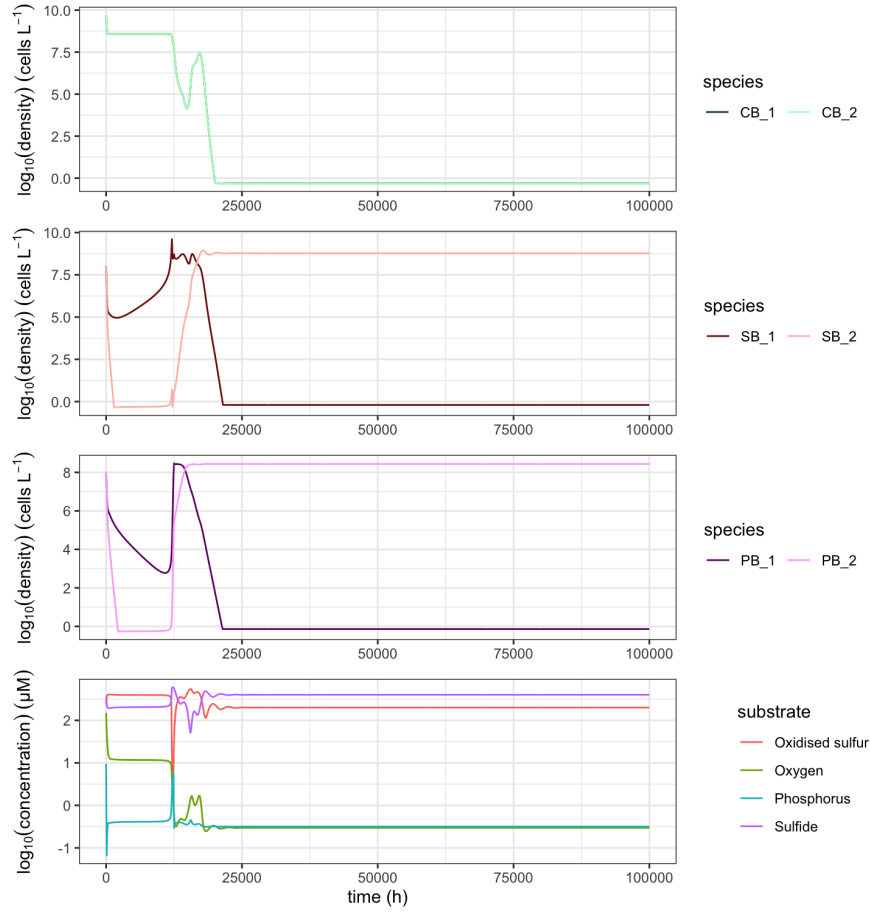


Figure 5: The dynamics of the ecosystem when there are two species in each functional group, and some variation (diversity) in species parameters.

Figure 5 shows a simulation with two species (strains) in each of the three functional groups. The ecosystem starts in the oxic state, though with relatively high abundance of each functional group. The strain of SB that is more tolerant to oxygen (SB_1) initially decreases in abundance, but then increases, and the other (SB_2) strain then becomes abundance and SB_1 declines. Furthermore, the cyanobacteria crash in abundance, and the system switches to the anoxic state. In contrast, if there are two identical strains with tolerance half way between those in Figure 5 the ecosystem remains in the oxic state.

4.4. Adapting for other ecosystems and organisms

We anticipate that the package can be a useful starting point for investigating models of other types of ecosystem and how biodiversity in them affects responses

267 to environmental change. The overall framework of the package, the purpose
268 of each function, and the objects used for storing parameters and results could
269 be retained. E.g., all such models would have parameters that differ among
270 species and need to be described in an object, and studies will often need to run
271 simulations and sets of simulations across environmental conditions.

272 Researchers wanting to model a new ecosystem do not, therefore, have to
273 start from scratch. This will relieve researchers from needing to make software
274 design decisions, and rather focus on appropriately representing their ecosystem
275 and finding the results that interest them. Nevertheless, adaptation of the code
276 in the package will require a person / persons that can take a conceptual model of
277 an ecosystem and then represent that in terms of parameters and rate equations,
278 and a person or persons relatively proficient in R programming.

279 5. Conclusions

280 The *microxanoz* R package allows the simulation, visualisation, and analysis
281 of a model of a microbial ecosystem while allowing variation in the amount of
282 diversity present in each of the functional groups of organisms present. It has
283 been used for the research described in another paper that provides one of the
284 first investigations of the effects of diversity on ecosystem resilience Limberger
285 et al. [8]. In that paper, we show that diversity can have large and important
286 effects on ecosystem responses, highlighting the need for models such as ours,
287 with which one can easily manipulate the amount of biodiversity. The *microxanoz*
288 package has also been used to reproduce the results of the paper that inspired
289 the package development [2].

290 The package greatly lowers the amount of work required in further investi-
291 gations of the specific ecosystem modelled. There has, for example, been quite
292 limited investigation of how biodiversity influences the short-term responses of
293 the modelled ecosystem to environmental change. Likewise, the package could be
294 used to power an investigation of the effects of biodiversity on the usefulness of
295 early warning signals of abrupt ecosystem change [13]. In addition this package
296 could be used as a template for developing models of other types of ecosystems
297 and organism. By doing so, other models can profit from the overall framework
298 used, and the reproducibility aspects as well as the flexibility implemented.

299 6. Conflict of Interest

300 The authors declare no known conflicting or competing interests associated
301 with this publication and there has been no significant financial support for this
302 work that could have influenced its outcome.

303 7. Acknowledgements

304 This project was part of SNF Project 310030_188431. The project was also
305 supported by the University of Zurich Research Priority Programme in Global
306 Change and Biodiversity.

References

- [1] Amrei Binzer, Christian Guill, Björn C. Rall, and Ulrich Brose. Interactive effects of warming, eutrophication and size structure: Impacts on biodiversity and food-web structure. *Global Change Biology*, 22(1):220–227, January 2016. ISSN 13541013. doi: 10.1111/gcb.13086.
- [2] Timothy Bush, Muhe Diao, Rosalind J. Allen, Ruben Sinnige, Gerard Muyzer, and Jef Huisman. Oxidic-anoxic regime shifts mediated by feedbacks between biogeochemical processes and microbial community dynamics. *Nature Communications*, 8(1):789, October 2017. ISSN 2041-1723. doi: 10.1038/s41467-017-00912-x.
- [3] P. Catalina Chaparro-Pedraza. Fast environmental change and eco-evolutionary feedbacks can drive regime shifts in ecosystems before tipping points are crossed. *Proceedings of the Royal Society B: Biological Sciences*, 288(1955):20211192, July 2021. ISSN 0962-8452, 1471-2954. doi: 10.1098/rspb.2021.1192.
- [4] Scott L. Collins, Jesse B. Nippert, John M. Blair, John M. Briggs, Pamela Blackmore, and Zak Ratajczak. Fire frequency, state change and hysteresis in tallgrass prairie. *Ecology Letters*, 24(4):636–647, April 2021. ISSN 1461-023X, 1461-0248. doi: 10.1111/ele.13676.
- [5] Adrian A. Correndo, Austin Pearce, Carl H. Bolster, John T. Spargo, Deanna Osmond, and Ignacio A. Ciampitti. The soiltestcorr R package: An accessible framework for reproducible correlation analysis of crop yield and soil test data. *SoftwareX*, 21:101275, February 2023. ISSN 23527110. doi: 10.1016/j.softx.2022.101275.
- [6] Nathan Fox, Tom August, Francesca Mancini, Katherine E. Parks, Felix Eigenbrod, James M. Bullock, Louis Sutter, and Laura J. Graham. “photosearcher” package in R: An accessible and reproducible method for harvesting large datasets from Flickr. *SoftwareX*, 12:100624, July 2020. ISSN 23527110. doi: 10.1016/j.softx.2020.100624.
- [7] Rainer M Krug and Owen L Petchey. Microxanox: Model to simulate three functional group system. June 2022. doi: 10.5281/ZENODO.6624125.
- [8] Romana Limberger, Uriah Daugaard, Anubhav Gupta, Rainer M. Krug, Kimberley D. Lemmen, Sofia J. Van Moorsel, Marcel Suleiman, Debra Zuppinge-Dingley, and Owen L. Petchey. Functional diversity can facilitate the collapse of an undesirable ecosystem state. *Ecology Letters*, 26(6): 883–895, June 2023. ISSN 1461-023X, 1461-0248. doi: 10.1111/ele.14217.
- [9] Amanda C. Northrop, Vanessa Avalone, Aaron M. Ellison, Bryan A. Ballif, and Nicholas J. Gotelli. Clockwise and counterclockwise hysteresis characterize state changes in the same aquatic ecosystem. *Ecology Letters*, 24(1): 94–101, January 2021. ISSN 1461-023X, 1461-0248. doi: 10.1111/ele.13625.

- 347 [10] Owen L Petchey, Uriah Daugaard, Anubhav Gupta, Rainer M Krug, Kimber-
348 ley D. Lemmen, van Moorsel, Sofia J., Marcel Suleiman, Zuppinger-Dingley,
349 Debra, and Romana Limberger. Code package for: "Functional diversity
350 can facilitate the collapse of an undesirable ecosystem state". Zenodo, June
351 2022.
- 352 [11] R Core Team. *R: A Language and Environment for Statistical Computing*.
353 R Foundation for Statistical Computing, Vienna, Austria, 2022.
- 354 [12] Marten Scheffer, Steve Carpenter, Jonathan A. Foley, Carl Folke, and Brian
355 Walker. Catastrophic shifts in ecosystems. *Nature*, 413(6856):591–596,
356 October 2001. ISSN 0028-0836, 1476-4687. doi: 10.1038/35098000.
- 357 [13] Marten Scheffer, Jordi Bascompte, William A. Brock, Victor Brovkin,
358 Stephen R. Carpenter, Vasilis Dakos, Hermann Held, Egbert H. van Nes,
359 Max Rietkerk, and George Sugihara. Early-warning signals for critical
360 transitions. *Nature*, 461(7260):53–59, September 2009. ISSN 0028-0836,
361 1476-4687. doi: 10.1038/nature08227.
- 362 [14] Karline Soetaert, Thomas Petzoldt, and R. Woodrow Setzer. Solving
363 differential equations in R: Package deSolve. *Journal of Statistical Software*,
364 33(9):1–25, 2010. doi: 10.18637/jss.v033.i09.
- 365 [15] Josef Spillner, Panagiotis Gkikopoulos, Pamela Delgado, and Christine
366 Choirat. Towards reproducible software studies with MAO and Renku.
367 *SoftwareX*, 17:100947, January 2022. ISSN 23527110. doi: 10.1016/j.softx.
368 2021.100947.
- 369 [16] John Vandermeer and Ivette Perfecto. Hysteresis and critical transitions in
370 a coffee agroecosystem. *Proceedings of the National Academy of Sciences*,
371 116(30):15074–15079, July 2019. ISSN 0027-8424, 1091-6490. doi: 10.1073/
372 pnas.1902773116.
- 373 [17] David A. Vasseur and Kevin S. McCann. A Mechanistic Approach for
374 Modeling Temperature-Dependent Consumer-Resource Dynamics. *The*
375 *American Naturalist*, 166(2):184–198, August 2005. ISSN 0003-0147, 1537-
376 5323. doi: 10.1086/431285.