

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

4             Rainer M Krug<sup>a,1</sup>, Owen L. Petchey<sup>a</sup>

<sup>a</sup>*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, steady state, final state, ordinary  
7     differential equations

---

---

\*Corresponding author

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

<sup>1</sup>Corresponding Author

<sup>2</sup>Equal 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	<a href="https://github.com/UZH-PEG/microxanox">https://github.com/UZH-PEG/microxanox</a> <a href="https://doi.org/10.5281/zenodo.7148667">10.5281/zenodo.7148667</a>
C3	Code Ocean compute capsule	
C4	Legal Code License	MIT
C5	Code versioning system used	<a href="#">git</a>
C6	Software code languages, tools, and services used	<a href="#">R</a>
C7	Compilation requirements, operating environments	<a href="#">R (&gt;= 4.1.0)</a>  magrittr tibble ggplot2 patchwork grDevices stats mgcv deSolve dplyr tidyr stringr multidplyr
C8	If available Link to developer documentation/manual	<a href="#">User Guide</a>
C9	Support email for questions	<a href="mailto:Rainer.Krug@uzh.ch">Rainer.Krug@uzh.ch</a> ; <a href="mailto:Owen.Petchey@ieu.uzh">Owen.Petchey@ieu.uzh</a>

## 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, 16]. 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, 15].

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 to include the vectors and  
77 matrices and used matrix mathematics. In this way, we made it possible to  
78 run simulations with different numbers of species without having to change the  
79 underlying code.

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

#### 86 3.1. Software architecture

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

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

### 102 3.2. Running one simulation

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

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

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

### 119 3.3. Examining ecosystem responses to environmental change

120 One approach to finding how the ecosystem responds to environmental driver  
121 change is to examine the relationship between the environmental driver value and  
122 the numerical value of a system's steady state (if it exists). The first component  
123 of this approach is to run the simulation for a long time and record the final  
124 state (i.e., the state of the system at the end of a simulation). When one does  
125 this across a range of environmental conditions, one discovers how the final  
126 state of the system changes with the environmental conditions. The package  
127 contains functionality for finding final states that correspond to values of one  
128 environmental driver, namely the value of oxygen diffusivity.

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

139 Two numerical strategies for finding final states and their sensitivity to  
140 parameters are implemented. Two strategies are implemented in order to allow  
141 comparison of their results. The first method runs a independent simulation for  
142 each combination of initial conditions and oxygen diffusivity (we term this the  
143 *Replication method*). This is the method used in the Bush et al. [2] study and  
144 was used to obtain the results in figures 3 and 4 of that article.

145 The second method runs two simulations, one with step-wise and slowly  
146 temporally *increasing* oxygen diffusivity, and the other with step-wise and slowly

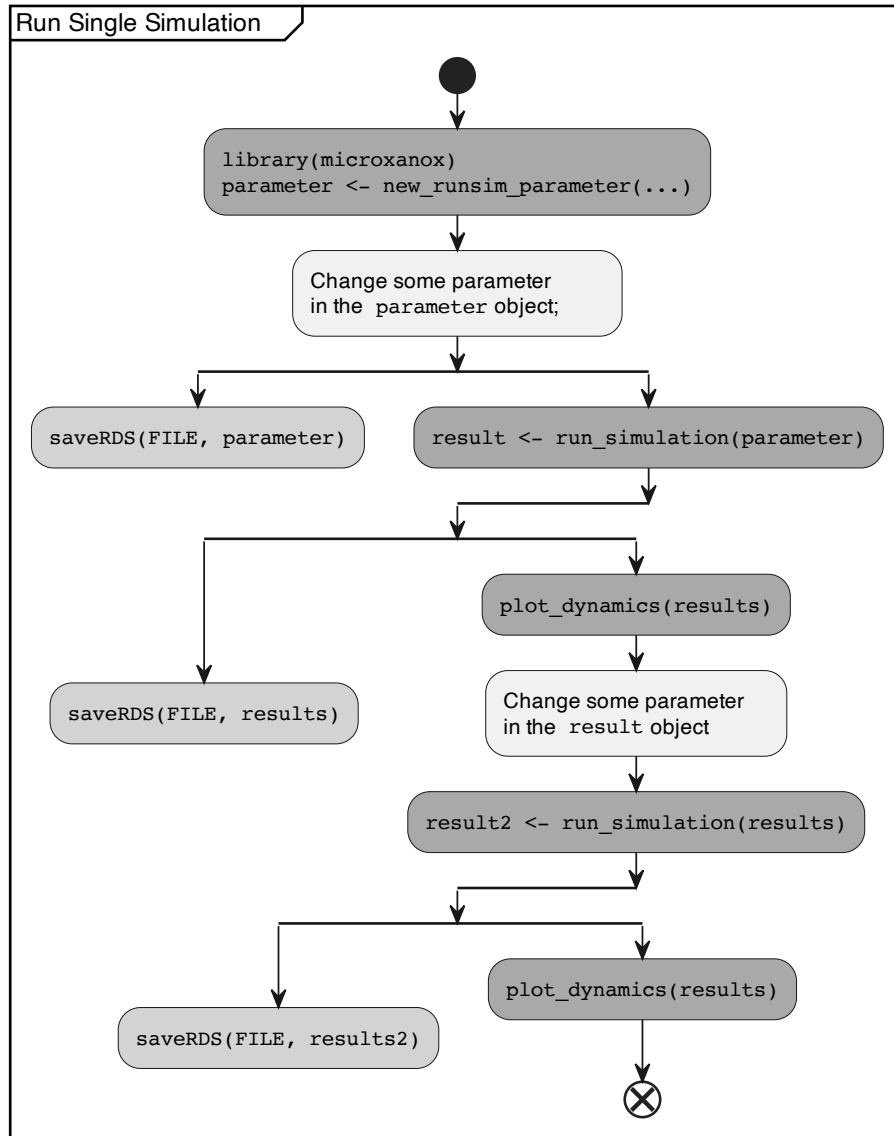


Figure 1: Typical flow of a simulation. Dark Grey boxes: commands necessary for simulation; Light Grey: Saving of parameter and results; Lightest Grey: Different non specified commands.

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

153 The replication method is implemented in the function `run_replication_ssfind()`  
154 which takes a parameter object as returned by the function `new_replication_ssfind_parameter()`  
155 and the number of cores for multithreading the simulation. As the multithreading  
156 uses the package function `mclapply()` from the package `parallel` [11], the mul-  
157 tithreading only works on Linux and Mac. It is planned to move to `parLapply()`  
158 [11] in a future release.

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

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

### 164 3.4. Analysing and visualising results

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

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

## 173 4. Impact

174 The open source implementation and extension of the model used in Bush  
175 et al. [2] provides the means of reproducing the results published while at the  
176 same time provides the means of making unique, innovative, and important  
177 investigations of how ecosystems change as the environmental changes, and how  
178 biodiversity may modulate this.

179 The design of the package code and functionality is with reproducibility in  
180 mind (for other example of reproducibility directed software, see e.g. Correndo  
181 et al. [5], ? ] Fox et al. [6]): the combination of all parameters being in a  
182 single parameter object and the return of the simulation as a result object which  
183 inherits from the parameter object, provides a relatively easy-to-use framework  
184 to implement reproducible experiments.

185 Here we evidence the impact of the *microxanox* package by describing three  
186 use-cases and then by describing how the package can be a starting point for  
187 models of other ecosystems. The first two use cases are described in detail  
188 (including the code for reproducing them) in the [User Guide](#) and the *Partial*

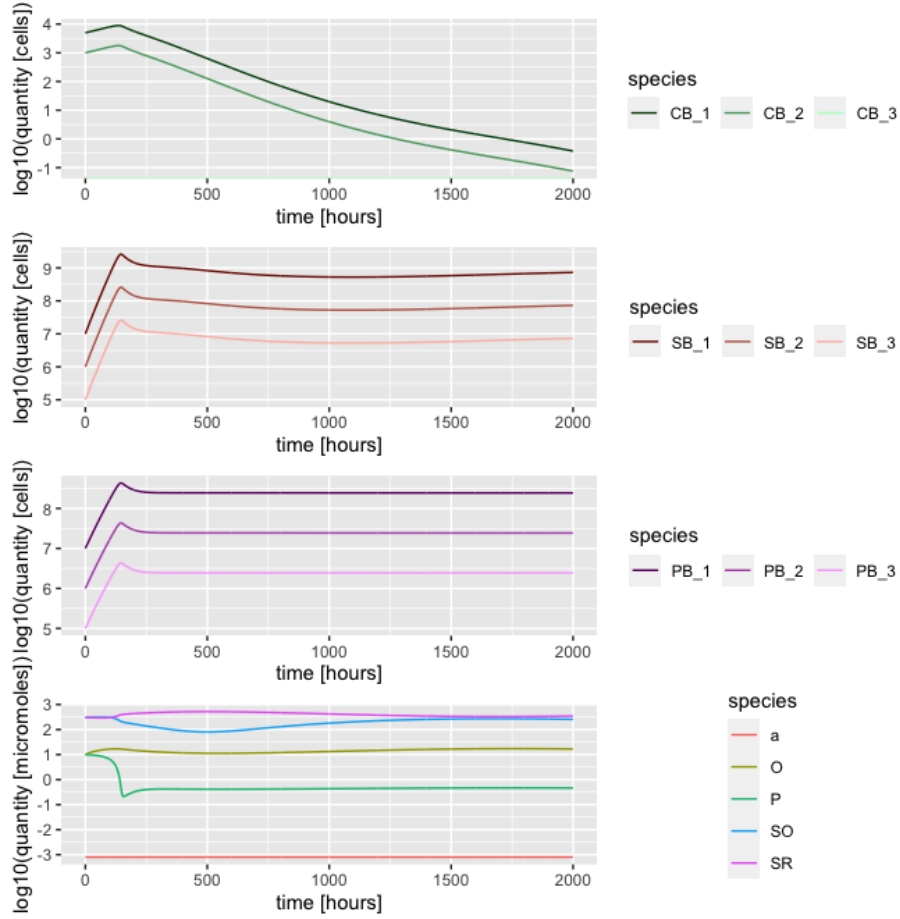


Figure 2: Results of a simulation run shown using the function `plot_dynamics()`. In this case, there were three strains per functional group, though strains within functional groups had identical properties in this example. CB\_1 = cyanobacteria strain 1; SB\_1 = sulfur reducing bacteria stain 1; PB\_1 = phototrophic sulfur bacteria strain 1.



189 *Reproduction* vignettes. The third is taken from Limberger et al. [8] and Petchey  
190 et al. [10].

#### 191 4.1. Use case 1: Regime shifts during temporal environmental change

192 The study of Bush et al. [2] includes simulations of the effect of oxygen  
193 diffusivity (an environmental driver, in the sense that it affects the ecosystem but  
194 is not affected by it) on the ecosystem state (oxic or anoxic). The `microxonox`  
195 package contains functionality to make a specific temporal pattern of change  
196 in the oxygen diffusivity. As well as allowing individual simulations during  
197 which oxygen diffusivity varies, this functionality forms the basis of the temporal  
198 method for finding final states.

199 An example of this functionality is given in the *Partial Reproduction* vignette,  
200 which we briefly describe and show here (Figure 3). The example is composed of  
201 a single simulation, at the beginning of which the system is in the oxic state with  
202 high abundance of cyanobacteria. Oxygen diffusivity is then slowly decreased  
203 and eventually, around hour 30'000 the system switches to the anoxic state, with  
204 high abundance of both sulfur bacteria types. The oxygen diffusivity is then  
205 increased and at around hour 38'000 the system abruptly switches back to the  
206 oxic state.

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

#### 213 4.2. Use case 2: The extent of hysteresis depends on community composition

214 The package contains a function to extract summary features of ecosystem  
215 responses to environmental change, such as the amount of hysteresis displayed by  
216 the ecosystem. Hysteresis is a key feature of ecosystem responses to environmental  
217 change, because it is related to how difficult it can be to reverse the effects of  
218 environmental change [12]. The amount of hysteresis is measured as the extent  
219 of the environmental condition (here oxygen diffusivity) for which there were  
220 two steady states. I.e., it is the extent of the environmental conditions for which  
221 historical conditions play an important role in determining the current system  
222 state.

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

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



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 stain 1; PB = phototrophic sulfur bacteria strain 1. Here we show a figure adapted from the output of the `plot_dynamics()` function.

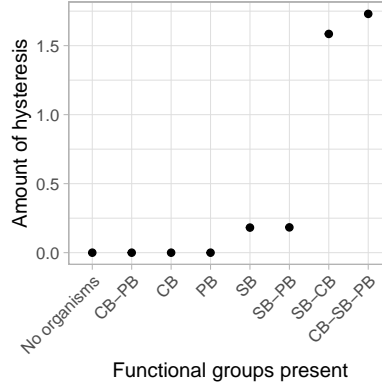


Figure 4: The amount of hysteresis depends on the combination of types of organisms present. The model is entirely deterministic, hence there are no error bars.

#### 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 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 abundant 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 to environmental change. The overall framework of the package, the purpose of each function, and the objects used for storing parameters and results could be retained. E.g., all such models would have parameters that differ among

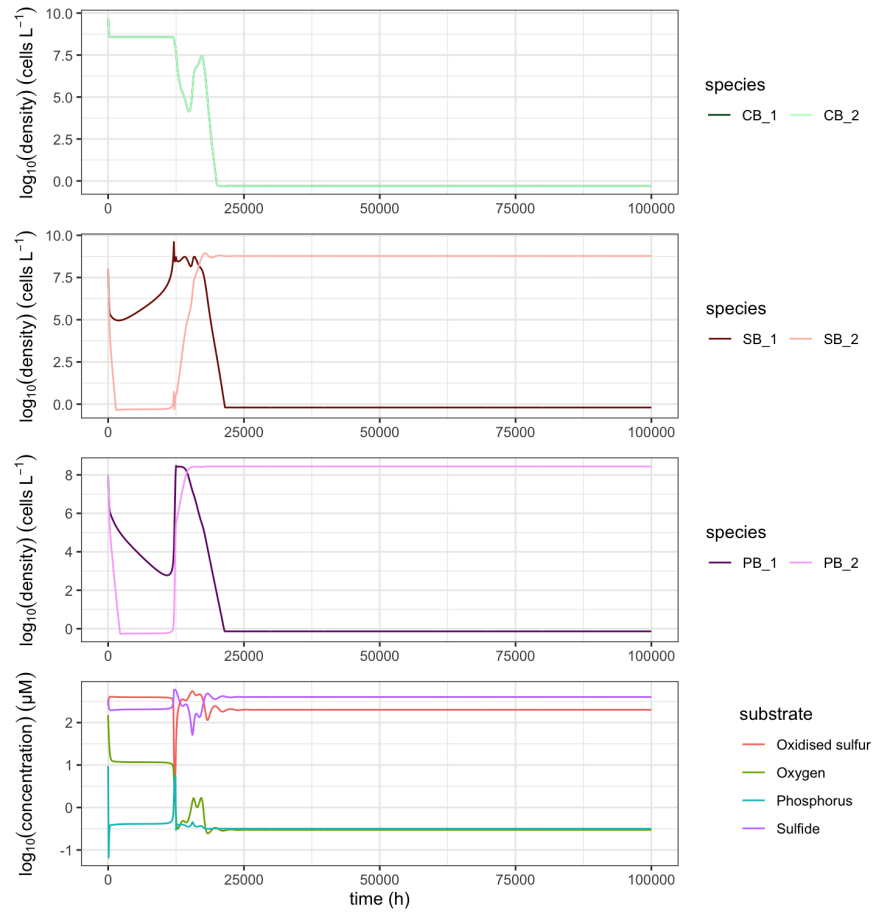


Figure 5: The dynamics of the ecosystem when there are two species in each functional group, and some variation (diversity) in species parameters.

261 species and need to be described in an object, and studies will often need to run  
262 simulations and sets of simulations across environmental conditions.

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

## 270 5. Conclusions

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

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

## 290 6. Conflict of Interest

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

## 294 7. Acknowledgements

295 This project was part of SNF Project 310030\_188431. The project was also  
296 supported by the University of Zurich Research Priority Programme in Global  
297 Change and Biodiversity.

## 298 References

- 299 [1] Amrei Binzer, Christian Guill, Björn C. Rall, and Ulrich Brose. Interactive  
300 effects of warming, eutrophication and size structure: Impacts on biodiversity  
301 and food-web structure. *Global Change Biology*, 22(1):220–227, January  
302 2016. ISSN 13541013. doi: 10.1111/gcb.13086.
- 303 [2] Timothy Bush, Muhe Diao, Rosalind J. Allen, Ruben Sinnige, Gerard  
304 Muyzer, and Jef Huisman. Oxidic-anoxic regime shifts mediated by feed-  
305 backs between biogeochemical processes and microbial community dynamics.  
306 *Nature Communications*, 8(1):789, October 2017. ISSN 2041-1723. doi:  
307 10.1038/s41467-017-00912-x.
- 308 [3] P. Catalina Chaparro-Pedraza. Fast environmental change and eco-  
309 evolutionary feedbacks can drive regime shifts in ecosystems before tip-  
310 ping points are crossed. *Proceedings of the Royal Society B: Biological*  
311 *Sciences*, 288(1955):20211192, July 2021. ISSN 0962-8452, 1471-2954. doi:  
312 10.1098/rspb.2021.1192.
- 313 [4] Scott L. Collins, Jesse B. Nippert, John M. Blair, John M. Briggs, Pamela  
314 Blackmore, and Zak Ratajczak. Fire frequency, state change and hysteresis  
315 in tallgrass prairie. *Ecology Letters*, 24(4):636–647, April 2021. ISSN  
316 1461-023X, 1461-0248. doi: 10.1111/ele.13676.
- 317 [5] Adrian A. Correndo, Austin Pearce, Carl H. Bolster, John T. Spargo,  
318 Deanna Osmond, and Ignacio A. Ciampitti. The soiltestcorr R package:  
319 An accessible framework for reproducible correlation analysis of crop yield  
320 and soil test data. *SoftwareX*, 21:101275, February 2023. ISSN 23527110.  
321 doi: 10.1016/j.softx.2022.101275.
- 322 [6] Nathan Fox, Tom August, Francesca Mancini, Katherine E. Parks, Fel-  
323 ix Eigenbrod, James M. Bullock, Louis Sutter, and Laura J. Graham.  
324 “photosearcher” package in R: An accessible and reproducible method for  
325 harvesting large datasets from Flickr. *SoftwareX*, 12:100624, July 2020.  
326 ISSN 23527110. doi: 10.1016/j.softx.2020.100624.
- 327 [7] Rainer M Krug and Owen L Petchey. Microxanox: Model to simulate three  
328 functional group system. June 2022. doi: 10.5281/ZENODO.6624125.
- 329 [8] Romana Limberger, Uriah Daugaard, Anubhav Gupta, Rainer M. Krug,  
330 Kimberley D. Lemmen, Sofia J. Van Moorsel, Marcel Suleiman, Debra  
331 Zuppingier-Dingley, and Owen L. Petchey. Functional diversity can facilitate  
332 the collapse of an undesirable ecosystem state. *Ecology Letters*, 26(6):  
333 883–895, June 2023. ISSN 1461-023X, 1461-0248. doi: 10.1111/ele.14217.
- 334 [9] Amanda C. Northrop, Vanessa Avalone, Aaron M. Ellison, Bryan A. Ballif,  
335 and Nicholas J. Gotelli. Clockwise and counterclockwise hysteresis charac-  
336 terize state changes in the same aquatic ecosystem. *Ecology Letters*, 24(1):  
337 94–101, January 2021. ISSN 1461-023X, 1461-0248. doi: 10.1111/ele.13625.

- 338 [10] Owen L Petchey, Uriah Daugaard, Anubhav Gupta, Rainer M Krug, Kimber-  
339 ley D. Lemmen, van Moorsel, Sofia J., Marcel Suleiman, Zuppinger-Dingley,  
340 Debra, and Romana Limberger. Code package for: "Functional diversity  
341 can facilitate the collapse of an undesirable ecosystem state". Zenodo, June  
342 2022.
- 343 [11] R Core Team. *R: A Language and Environment for Statistical Computing*.  
344 R Foundation for Statistical Computing, Vienna, Austria, 2022.
- 345 [12] Marten Scheffer, Steve Carpenter, Jonathan A. Foley, Carl Folke, and Brian  
346 Walker. Catastrophic shifts in ecosystems. *Nature*, 413(6856):591–596,  
347 October 2001. ISSN 0028-0836, 1476-4687. doi: 10.1038/35098000.
- 348 [13] Marten Scheffer, Jordi Bascompte, William A. Brock, Victor Brovkin,  
349 Stephen R. Carpenter, Vasilis Dakos, Hermann Held, Egbert H. van Nes,  
350 Max Rietkerk, and George Sugihara. Early-warning signals for critical  
351 transitions. *Nature*, 461(7260):53–59, September 2009. ISSN 0028-0836,  
352 1476-4687. doi: 10.1038/nature08227.
- 353 [14] Karline Soetaert, Thomas Petzoldt, and R. Woodrow Setzer. Solving  
354 differential equations in R: Package deSolve. *Journal of Statistical Software*,  
355 33(9):1–25, 2010. doi: 10.18637/jss.v033.i09.
- 356 [15] John Vandermeer and Ivette Perfecto. Hysteresis and critical transitions in  
357 a coffee agroecosystem. *Proceedings of the National Academy of Sciences*,  
358 116(30):15074–15079, July 2019. ISSN 0027-8424, 1091-6490. doi: 10.1073/  
359 pnas.1902773116.
- 360 [16] David A. Vasseur and Kevin S. McCann. A Mechanistic Approach for  
361 Modeling Temperature-Dependent Consumer-Resource Dynamics. *The*  
362 *American Naturalist*, 166(2):184–198, August 2005. ISSN 0003-0147, 1537-  
363 5323. doi: 10.1086/431285.