

1 Microxanox: an R package for simulating an aquatic
2 *MIC*Robial ecosystem that can occupy *OX*ic or
3 *ANOX*ic 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, to find stable states. 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, stable 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.

Code metadata		
Nr.	description	Please fill in this column
C1	Current code version	v0.9.0
C2	Permanent links to code/repository used for this code version	https://github.com/UZH-PEG/microxanox 10.5281/zenodo.6624125
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 ($\geq 4.1.0$) 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
13 models and the understanding about how ecosystems work and respond to

14 environmental changes [1, 2, 3]. They are also important for developing
15 hypotheses to test in empirical studies. One area of ecology in which models
16 have been influential is the understanding of their response to a gradual change
17 in an environmental driver [4]. An environmental driver is an environmental
18 condition that affects an ecosystem. Still, it is assumed not to be affected by
19 the ecosystem, such as the rate of nutrient input into a lake.

20 It is conceivable that an ecosystem state, such as the total biomass of a
21 particular type of bacteria, may remain unchanged when an environmental
22 driver changes. It is also possible that the ecosystem state changes gradually.
23 It is also possible that the ecosystem state changes abruptly to a new state
24 that is difficult to recover from [4]. This possibility for abrupt, perhaps
25 catastrophic changes that are difficult to reverse causes considerable concern
26 [5, 6, 7].

27 An example where a gradual change of an environmental variable causes
28 an abrupt change in the system is the switch from an aerobic (oxygen is
29 available for metabolism) to an anaerobic (oxygen generally unavailable) state
30 in a microbial ecosystem. This system has been investigated by (author?) [8]
31 in a simulation study of a mathematical model. Three types of microbes occur
32 in the model: cyanobacteria (CB) dominating the oxic state, and two types
33 of sulfur bacteria that dominate the anoxic state (sulfate reducing bacteria
34 (SB) and phototrophic sulfur bacteria (PB)). The model shows that gradual
35 change in the rate at which oxygen could diffuse into the ecosystem (termed
36 oxygen diffusivity) could cause catastrophic changes in the ecosystem state
37 that would be difficult to reverse.

38 One feature of the study by (author?) [8] was limited biodiversity. Specif-
39 ically, there was no biodiversity within each of the three types of bacteria.
40 This leaves the question of if and how bacteria's biodiversity within these
41 types (i.e., functional groups) affects the ecosystem response to environmental
42 change. This limitation is not specific to the study of (author?) [8]. There
43 are few if any studies of the effects of biodiversity on abrupt transitions
44 between ecosystem states.

45 We decided to fill this research gap by making a simulation study of how
46 within functional group biodiversity affects ecosystem responses to environ-
47 mental change (author?) [9], and to base our work on the work and model of
48 (author?) [8]. With this goal in mind, we developed the microxanox package
49 [10]. The first stage of development was to write code from scratch (as there
50 was no available code to start from) and to confirm that this new imple-
51 mentation would reproduce the previously published results. The resulting
52 reproduction is available as one of the package vignettes: [vignette Partial
53 reproduction of Bush et al.](#)

54 The second stage was to add functionality that would be necessary to

answer our research question. Most importantly, we made it possible to have multiple species of bacteria within each of the three functional groups, for the multiple species to differ in their characteristics, and to vary the number of species and amount of variability among them. We also added functionality that allowed: temporally varying environmental conditions, the addition of random noise to state variables, and immigration. In addition to the model itself, the package includes some functions to analyse the results and visualize these to provide a starting point for customized visualizations based on own requirements. The basic and additional functionality is described in the package [User Guide](#).

3. Software description

The *microxanoz* package is for simulating a three functional group system (*CB*: cyanobacteria, *PB*: phototrophic sulfur bacteria, and *SB*: sulfate-reducing bacteria) with four chemical substrates (*P*: phosphorus, *O*: oxygen, *SR*: reduced sulfur, *SO*: oxidized sulfur). It includes feedback between organisms and biogeochemical processes and is based on (author?) [8] (See (author?) [8] for a detailed discussion of the model). At the core of the simulations is a set of ordinary differential equations (specified in the function `bushplus_dynamic_model()`, though this function need not be directly called). There are functions for running individual simulations and a set of simulations across, for example, a range of environmental conditions.

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

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

3.1. Software architecture

The framework used when writing this package aimed to maximise simplicity for the user and to make it straightforward to reproduce results (see the supplement [11] to (author?) [9] for an example of how this is used). As such, all the parameters needed to run a simulation or find a stable state

93 (i.e. the final state of the ecosystem) are contained in a single object (which
94 can easily be created using included functions). This parameter object is given
95 to a function that runs the simulations and returns the results. The returned
96 results object is identical to the parameter object but with an additional slot
97 named **results**, which contains the simulation results. Thus the returned
98 results object contains the simulation conditions (parameters) as well as
99 the results, and can be used to run the simulation again. This promotes
100 reproducibility and makes incremental changes of individual parameters with
101 a consecutive re-running of the simulations straightforward.

102 In the following sections we describe how to use the package to run one
103 simulation and to find steady states across an environmental gradient.

104 3.2. Running one simulation

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

106 A simulation is run using the `run_simulation()` function. In this func-
107 tion, the ODEs are solved using the function `ode()` in the package *deSolve*
108 package [12]. The `run_simulation()` function needs only one argument -
109 an object as created by the function `new_runsim_parameter()`. The pa-
110 rameter object returned by `new_runsim_parameter()` contains, among other
111 things, the `strain_parameter` object, which can be created by the func-
112 tion `new_strain_parameter()`. For a detailed description of the parameter
113 objects, their meaning and how they are created and have values set and
114 changed, please see the *User Guide* which accompanies the package or is
115 available at [User Guide](#).

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

122 3.3. Finding Stable States

123 The general approach to finding the numerical value of a system's stable
124 state (if it exists) with a specific parameter set is to run the simulation for a
125 long time and record the final state. When one does this across a range of
126 environmental conditions, one discovers how the steady state of the system
127 responds to the environmental conditions. The package contains functionality
128 for finding steady states that correspond to values of one environmental driver,
129 namely the value of oxygen diffusivity.

130 Two methods for finding steady states are implemented. The first runs a
131 separate simulation for each combination of starting conditions and oxygen

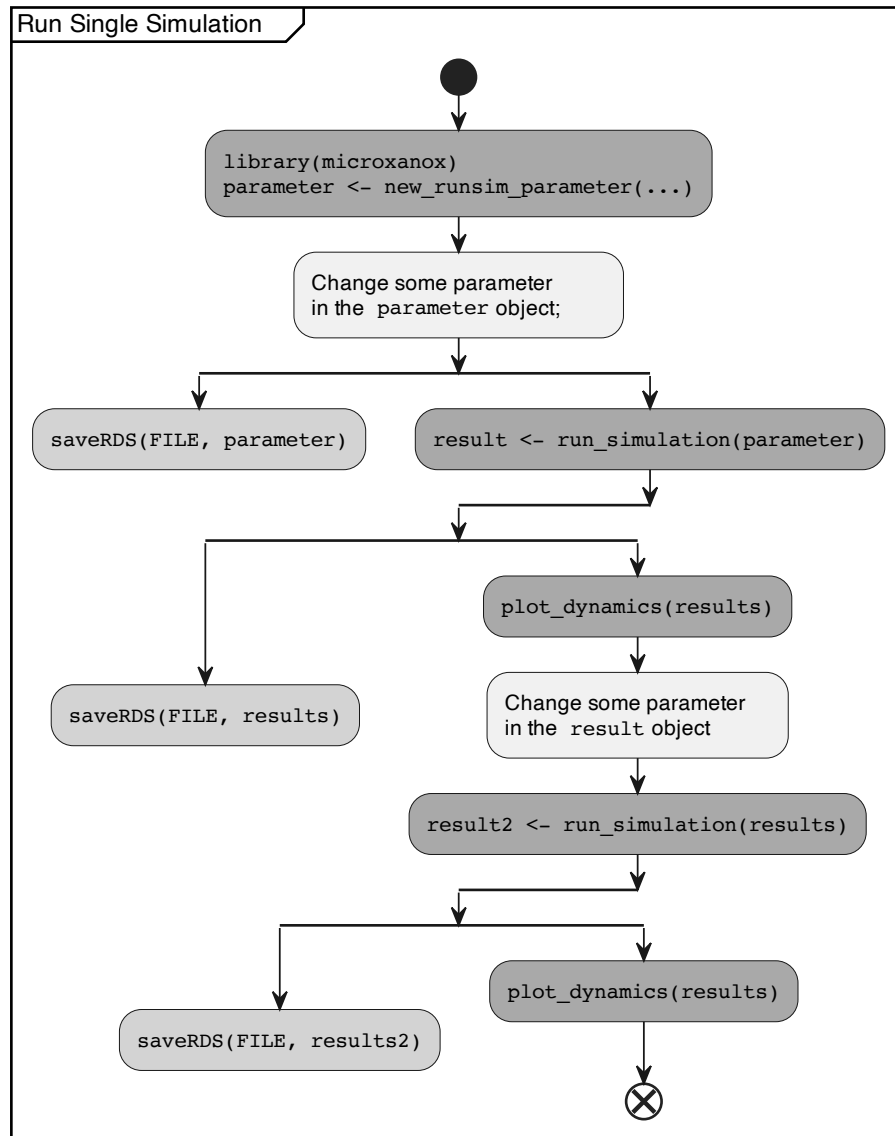


Figure 1: Typical flow of a simulation. Dark Grey boxes: commands necessary for simulation; Light Grey: Saving of parameter and results; Lightest Grey: Different non specified commands.

diffusivity (we term this the *Replication method*). This is the method used in the (author?) [8] study. The second runs two simulations, one with step-wise and slowly temporally *increasing* oxygen diffusivity, and the other with step-wise and slowly *decreasing* oxygen diffusivity. During this temporal environmental change, the state of the system is recorded just before change to a new oxygen diffusivity (we term this the *Temporal method*). We implemented two methods since there is no definitive best method, and in order to check if results were sensitive to choice of method.

The replication method is implemented in the function `run_replication_ssfind()` which takes a parameter object as returned by the function `new_replication_ssfind_parameter()` and the number of cores for multithreading the simulation. As the multithreading uses the package function `mclapply()` from the package `parallel` [13], the multithreading only works on Linux and Mac. It is planned to move to `parLapply()` [13] in a future release.

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

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

3.4. Analysing and visualising results

From the results returned, summary measures about how the ecosystem stable states respond to environmental change can be extracted. The function `get_stability_measures()` returns quantities such as the amount of environmental change required for the system to abruptly change to a different state.

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

4. Impact

The open source implementation and extension of the model used in (author?) [8] provides the means of reproducing the results published while at the same time provides the means of doing unique, innovative, and important investigations of how ecosystems respond to environmental change, and how biodiversity may modulate this response.

The design of the package code and functionality is with reproducibility in mind: the combination of all parameters being in a single parameter object as well as the return of the simulation as a result object which inherits from the

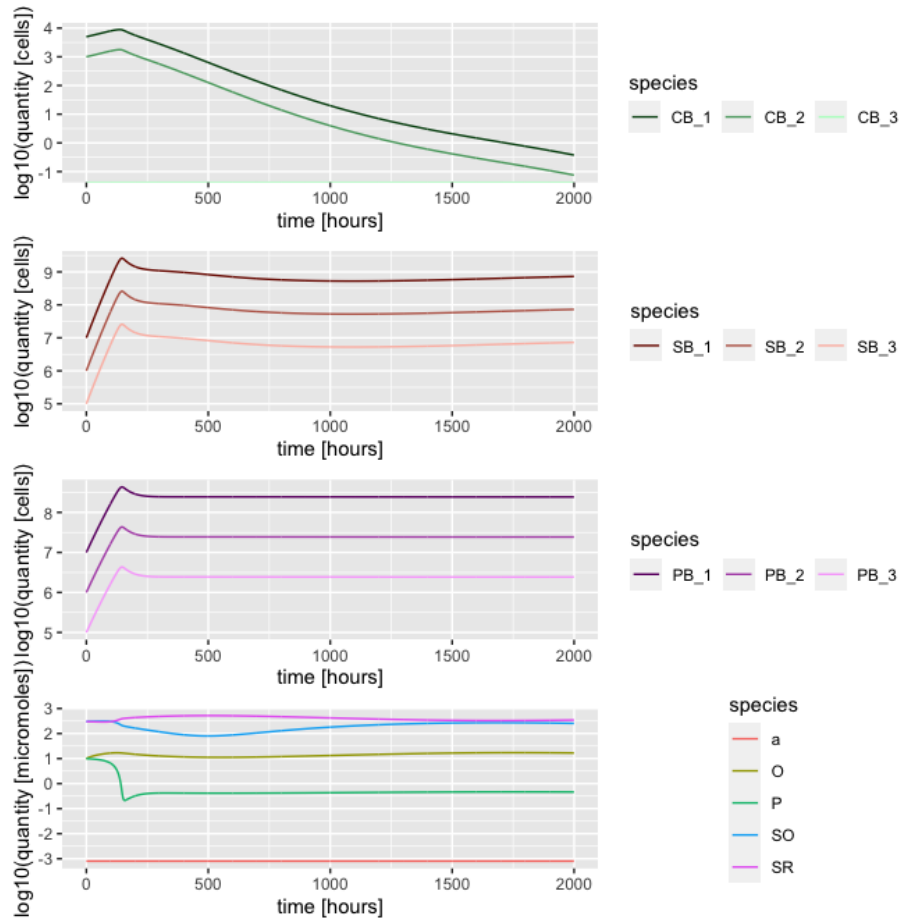


Figure 2: Results of a simulation run shown using the function *plot_dynamics()*. In this case, there were three strains per functional group, though strains within functional groups had identical properties in this example. CB_1 = cyanobacteria strain 1; SB_1 = sulfur reducing bacteria stain 1; PB_1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

170 parameter object provides a relatively easy to use framework to implement
171 reproducible experiments.

172 Here we evidence the impact of the *microxanax* package by describing
173 three use cases and then by describing how the package can be a starting
174 point for models of other ecosystems. The first two use cases are described in
175 detail (including the code for reproducing them) in the *User Guide* and the
176 *Partial Reproduction* vignettes. The third is taken from (author?) [9] and
177 (author?) [11].

178 4.1. Use case 1: Regime shifts during temporal environmental change

179 The study of (author?) [8] includes simulations of the effect of oxygen
180 diffusivity (an environmental driver, in the sense that it affects the ecosystem
181 but is not affected by it) on the ecosystem state (oxic or anoxic). The
182 *microxanax* package contains functionality to make a specific temporal pattern
183 of change in the oxygen diffusivity. As well as allowing individual simulations
184 during which oxygen diffusivity varies, this functionality forms the basis of
185 the temporal method for finding stable states.

186 An example of this functionality is given in the *Partial Reproduction*
187 vignette, which we briefly show here (Figure 3). The example is composed
188 of a single simulation, at the beginning of which the system is in the oxic
189 state with high abundance of cyanobacteria. Oxygen diffusivity is then slowly
190 decreased and eventually, around hour 30'000 the system switches to the
191 anoxic state, with high abundance of both sulfur bacteria types. The oxygen
192 diffusivity is then increased and at around hour 38'000 the system abruptly
193 switches back to the oxic state.

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

200 4.2. Use case 2: The extent of hysteresis depends on community composition

201 The package contains a function to extract summary features of ecosystem
202 responses to environmental change, such as the amount of hysteresis displayed
203 by the ecosystem. Hysteresis is a key feature of ecosystem responses to
204 environmental change, because it is related to how difficult it can be to reverse
205 the effects of environmental change [4]. The amount of hysteresis is measured
206 as the extent of the environmental condition (here oxygen diffusivity) for
207 which there were two stable states. I.e. it is the extent of the environmental

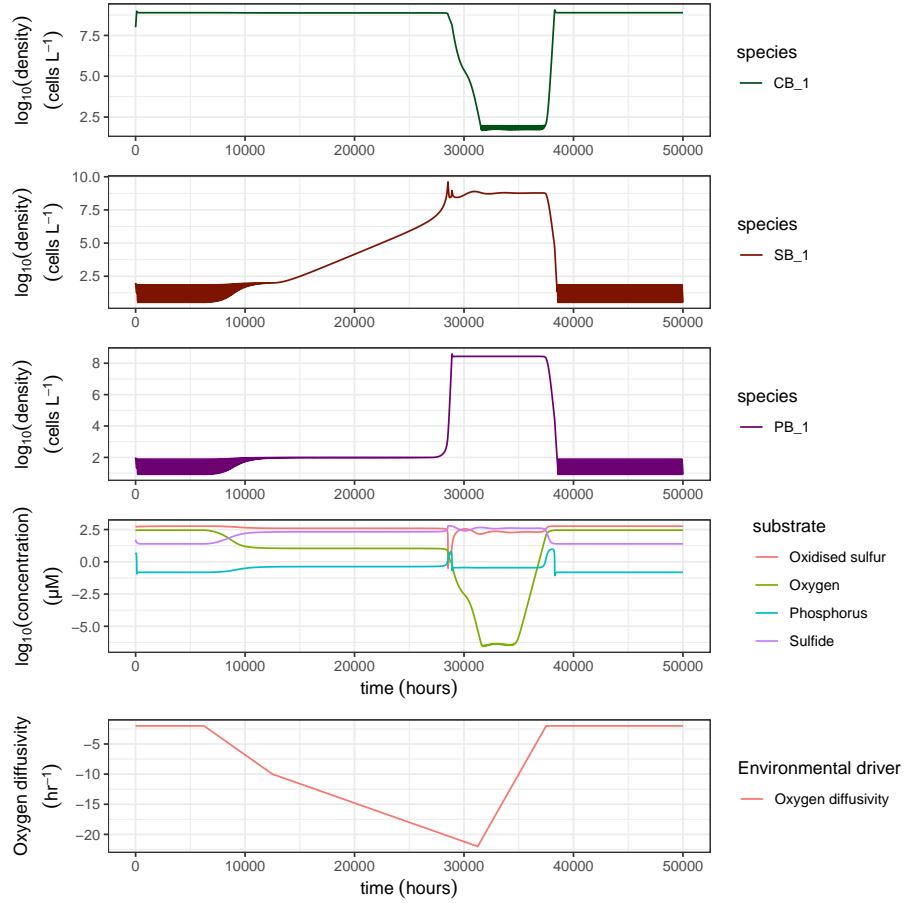


Figure 3: The temporal dynamics of the ecosystem model when an environmental condition (here parameter a , the oxygen diffusivity) changes. Plot of the stable states of the simulation runs under different oxygen diffusivity. In this simulation there is only one strain in each functional group. CB extunderscore1 = cyanobacteria strain 1; SB_1 = 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.

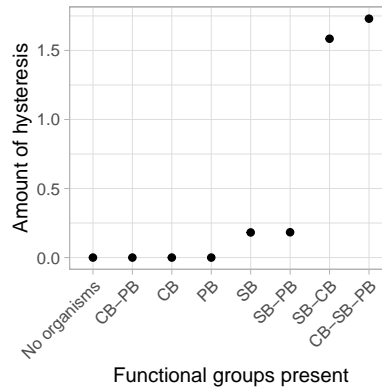


Figure 4: The amount of hysteresis depends on the combination of types of organisms present. The model is entirely deterministic, hence there are no error bars.

conditions for which historical conditions play an important role in determining the current system state (a definition of hysteresis).

Using the package to calculate the extent of hysteresis involves setting ecosystem and simulation parameters, including parameters for the finding of stable states across an environmental gradient, running the stable 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 (author?) [9]. 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.

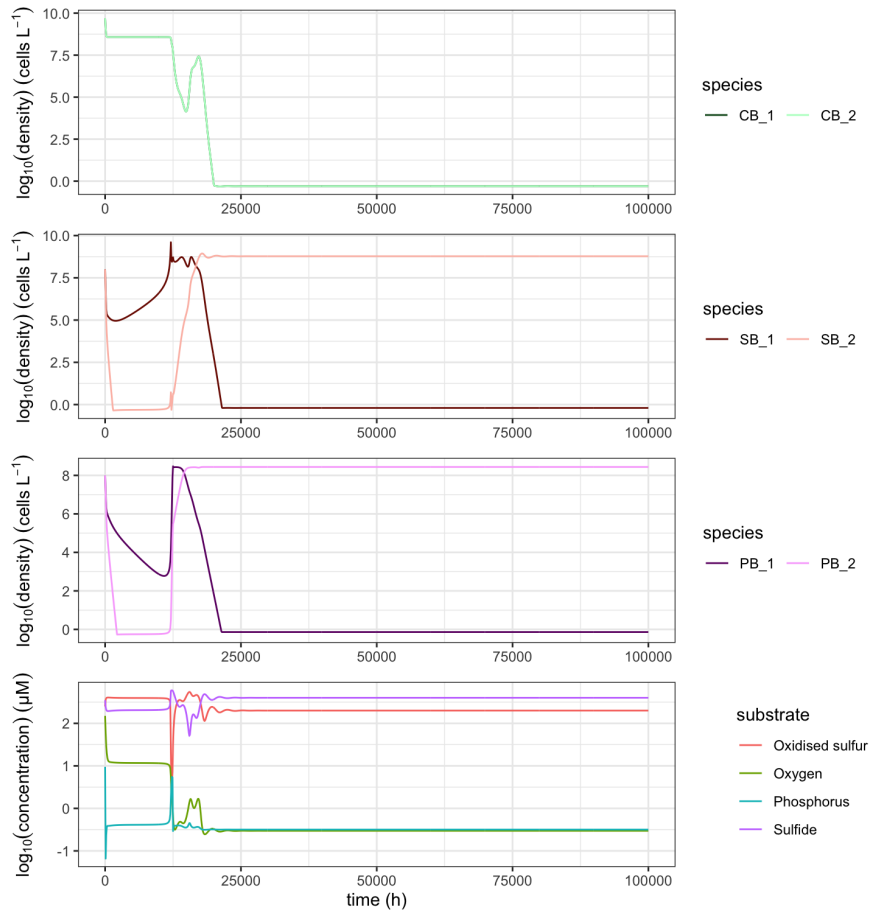


Figure 5: The dynamics of the ecosystem when there are two species in each functional group, and some variation (diversity) in species parameters.

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

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

252 Researchers wanting to model a new ecosystem do not, therefore, have
253 to start from scratch. This will relieve researchers from needing to make
254 software design decisions, and rather focus on appropriately representing
255 their ecosystem and finding the results that interest them. Nevertheless,
256 adaptation of the code in the package will require a person / persons that
257 can take a conceptual model of an ecosystem and then represent that in
258 terms of parameters and rate equations, and that is relatively proficient in R
259 programming.

260 5. Conclusions

261 The *microxanox* R package allows the simulation, visualisation, and
262 analysis of a model of a microbial ecosystem while allowing variation in the
263 amount of diversity containing in each of the functional groups of organisms
264 present. It has been used for the research described in another paper that
265 describes one of the first investigations of the effects of diversity on ecosystem
266 resilience (**author?**) [9]. In that paper, we show that diversity can have
267 large and important effects of ecosystem responses, highlighting the need for
268 models such as ours, with which one can easily manipulate the amount of
269 biodiversity. The *microxanox* package has also been used to reproduce the
270 results of the paper that inspired the package development [8].

271 The package greatly lowers the amount of work required in further investi-
272 gations of the specific ecosystem modelled. There has, for example, been quite
273 limited investigation of how biodiversity influences the short-term responses of
274 the modelled ecosystem to environmental change. Likewise, the package could
275 be used to power an investigation of the effects of biodiversity on the useful-
276 ness of early warning signals of abrupt ecosystem change [14]. In addition this
277 package could be used as a template for the implementation for developing
278 models of other types of ecosystems and organism. By doing so, other models
279 can profit from the overall framework used, and the reproducibility aspects
280 as well as the flexibility implemented.

281 6. Conflict of Interest

282 The authors declare no known conflicting or competing interests associated
283 with this publication and there has been no significant financial support for

284 this work that could have influenced its outcome.

285 7. Acknowledgements

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

289 References

- 290 [1] A. Binzer, C. Guill, B. C. Rall, U. Brose, Interactive effects of warming,
291 eutrophication and size structure: Impacts on biodiversity and food-
292 web structure, *Global Change Biology* 22 (1) (2016) 220–227. doi:
293 [10.1111/gcb.13086](https://doi.org/10.1111/gcb.13086).
- 294 [2] P. C. Chaparro-Pedraza, Fast environmental change and eco-evolutionary
295 feedbacks can drive regime shifts in ecosystems before tipping points
296 are crossed, *Proceedings of the Royal Society B: Biological Sciences*
297 288 (1955) (2021) 20211192. doi:[10.1098/rspb.2021.1192](https://doi.org/10.1098/rspb.2021.1192).
- 298 [3] D. A. Vasseur, K. S. McCann, A Mechanistic Approach for Modeling
299 Temperature-Dependent Consumer-Resource Dynamics, *The American*
300 *Naturalist* 166 (2) (2005) 184–198. doi:[10.1086/431285](https://doi.org/10.1086/431285).
- 301 [4] M. Scheffer, S. Carpenter, J. A. Foley, C. Folke, B. Walker, Catastrophic
302 shifts in ecosystems, *Nature* 413 (6856) (2001) 591–596. doi:[10.1038/](https://doi.org/10.1038/35098000)
303 [35098000](https://doi.org/10.1038/35098000).
- 304 [5] S. L. Collins, J. B. Nippert, J. M. Blair, J. M. Briggs, P. Blackmore,
305 Z. Ratajczak, Fire frequency, state change and hysteresis in tallgrass
306 prairie, *Ecology Letters* 24 (4) (2021) 636–647. doi:[10.1111/ele.13676](https://doi.org/10.1111/ele.13676).
- 307 [6] A. C. Northrop, V. Avalone, A. M. Ellison, B. A. Ballif, N. J. Gotelli,
308 Clockwise and counterclockwise hysteresis characterize state changes
309 in the same aquatic ecosystem, *Ecology Letters* 24 (1) (2021) 94–101.
310 doi:[10.1111/ele.13625](https://doi.org/10.1111/ele.13625).
- 311 [7] J. Vandermeer, I. Perfecto, Hysteresis and critical transitions in a coffee
312 agroecosystem, *Proceedings of the National Academy of Sciences* 116 (30)
313 (2019) 15074–15079. doi:[10.1073/pnas.1902773116](https://doi.org/10.1073/pnas.1902773116).
- 314 [8] T. Bush, M. Diao, R. J. Allen, R. Sinnige, G. Muyzer, J. Huisman,
315 Oxic-anoxic regime shifts mediated by feedbacks between biogeochemical

- 316 processes and microbial community dynamics, *Nature Communications*
317 8 (1) (2017) 789. [doi:10.1038/s41467-017-00912-x](https://doi.org/10.1038/s41467-017-00912-x).
- 318 [9] R. Limberger, U. Daugaard, A. Gupta, R. Krug, K. Lemmen, S. van
319 Moorsel, M. Suleiman, D. Zuppinger-Dingley, O. Petchey, Functional
320 diversity can facilitate the collapse of an undesirable ecosystem state, *Authorea Preprint* (Jun. 2022). [doi:10.22541/au.165513066.66785786/v1](https://doi.org/10.22541/au.165513066.66785786/v1).
322
- 323 [10] R. M. Krug, O. L. Petchey, Microxanox: Model to simulate three func-
324 tional group system. (Jun. 2022). [doi:10.5281/ZENODO.6624125](https://doi.org/10.5281/ZENODO.6624125).
- 325 [11] O. L. Petchey, U. Daugaard, A. Gupta, R. M. Krug, K. D. Lemmen, van
326 Moorsel, Sofia J., M. Suleiman, Zuppinger-Dingley, Debra, R. Limberger,
327 Code package for: "Functional diversity can facilitate the collapse of
328 an undesirable ecosystem state", *Zenodo* (Jun. 2022). [doi:10.5281/ZENODO.6334544](https://doi.org/10.5281/ZENODO.6334544).
329
- 330 [12] K. Soetaert, T. Petzoldt, R. W. Setzer, Solving differential equations in
331 R: Package deSolve, *Journal of Statistical Software* 33 (9) (2010) 1–25.
332 [doi:10.18637/jss.v033.i09](https://doi.org/10.18637/jss.v033.i09).
- 333 [13] R Core Team, *R: A Language and Environment for Statistical Computing*,
334 R Foundation for Statistical Computing, Vienna, Austria (2022).
- 335 [14] M. Scheffer, J. Bascompte, W. A. Brock, V. Brovkin, S. R. Carpenter,
336 V. Dakos, H. Held, E. H. van Nes, M. Rietkerk, G. Sugihara, Early-
337 warning signals for critical transitions, *Nature* 461 (7260) (2009) 53–59.
338 [doi:10.1038/nature08227](https://doi.org/10.1038/nature08227).