

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, 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 ecosystem.

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.

Nr.	Code metadata description	Please fill in this column
C1	Current code version	v0.9.0
C2	Permanent link 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 understanding
 13 about how ecosystems work and how they respond to environmental changes
 14 [1, 3, 14]. They are also important for developing hypotheses to test in empirical
 15 studies. One area of ecology in which models have played a influential role is
 16 how ecosystems respond to gradual change in an environmental driver [10]. An
 17 environmental driver is a environmental condition that affects an ecosystem, but
 18 is assumed to not be affected by the ecosystem, such as the rate of nutrient input
 19 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 driver
22 changes. It is also possible that the ecosystem state changes gradually. It is
23 also possible that the ecosystem state changes abruptly to a new state that is
24 difficult to recover from [10]. This possibility for abrupt, perhaps catastrophic
25 changes that are difficult to reverse causes considerable concern [4, 7, 13].

26 An example where a gradual change of an environmental variable causes an
27 abrupt change of the system is the switch from an aerobic (oxygen is available
28 for metabolism) to anaerobic (oxygen generally unavailable) state in a microbial
29 ecosystem. This system has been investigated by Bush et al. [2] in a simulation
30 study of a mathematical model. Three types of microbes occur in the model:
31 cyanobacteria (CB) dominating the oxic state, and two types of sulfur bacteria
32 that dominate the anoxic state (sulfate reducing bacteria (SB) and phototrophic
33 sulfur bacteria (PB)). The model shows that gradual change in the rate at which
34 oxygen could diffuse into the ecosystem (termed the oxygen diffusivity) could
35 cause catastrophic changes in the ecosystem state that would be difficult to
36 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 of bacteria. This leaves
39 open the question of if and how biodiversity within these types (i.e. functional
40 groups) of bacteria affects the ecosystem response to environmental change. This
41 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. [6], and to base our work on the work and model of Bush
47 et al. [2]. It was with this goal in mind that we developed the *microxanox* package
48 [5]. 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, addition of random noise to state
59 variables, and immigration. In addition to the model itself, the package includes
60 some functions to analyse the results as well as to visualize the results to provide
61 a 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 *microxanoz* 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 differ-
70 ential equations (specified in the function `bushplus_dynamic_model()`, though
71 this function need not be directly called). There are functions for running
72 individual simulations and for running a set of simulations across, for example,
73 a range 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 to use 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 modular structure so that new func-
81 tionality can be easily added. E.g. temporally defined events of any type could
82 be specified. Further, all parameter values required to run a simulation are
83 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 simplicity
88 for the user, and to make it straightforward to reproduce results (see the
89 supplement [8] to Limberger et al. [6] for an example of how this is used). As
90 such, all the parameters needed to run a simulation or find a stable state (i.e. the
91 final state of the ecosystem) are contained in a single object (which can easily be
92 created using included functions). This parameter object is given to a function
93 that runs the simulations and returns the results. The returned results object is
94 identical to the parameter object but with an additional slot named `results`,
95 which contains the results of the run. Thus the returned results object contains
96 the 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 and re-running the simulations straightforward.

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

101 3.2. Running one simulation

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

103 A simulation is run using the `run_simulation()` function. In this function,
104 the ODEs are solved using the function `ode()` in the package *deSolve* package [12].
105 The `run_simulation()` function needs only one argument - an object as created



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.

106 by the function `new_runsim_parameter()`. The parameter object returned by
107 `new_runsim_parameter()` contains among other things the `strain_parameter`
108 object, which can be created by the function `new_strain_parameter()`. For
109 a detailed description of the parameter objects, their meaning and how they
110 are created and have values set and changed please see the *User Guide* which
111 accompanies the package or is available at [User Guide](#).

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

118 3.3. Finding Stable States

119 The general approach used to find the stable state of the system with a
120 specific parameter set is to run the simulation for a long time and record the
121 final state. When one does this across a range of environmental conditions, one
122 discovers how the steady state of the system responds to the environmental
123 conditions. The package contains functionality for finding steady states that
124 correspond to values of one environmental driver, namely the value of oxygen
125 diffusivity.

126 Two methods for finding steady states are implemented. The first runs a
127 separate simulation for each combination of starting conditions and oxygen
128 diffusivity (we term this the *Replication method*). This is the method used in
129 the Bush et al. [2] study. The second runs two simulations, one with step-wise
130 and slowly temporally *increasing* oxygen diffusivity, and the other with step-wise
131 and slowly *decreasing* oxygen diffusivity. During this temporal environmental
132 change, the state of the system is recorded just before change to a new oxygen
133 diffusivity (we term this the *Temporal method*). We implemented two methods
134 since there is no definitive best method, and in order to check if results were
135 sensitive to choice of method.

136 The replication method is implemented in the function `run_replication_ssfind()`
137 which takes a parameter object as returned by the function `new_replication_ssfind_parameter()`
138 and the number of cores for multithreading the simulation. As the multithreading
139 uses the package function `mclapply()` from the package `parallel` [9], the multi-
140 threading only works on Linux and Mac. It is planned to move to `parLapply()`
141 [9] in a future release.

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

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

147 3.4. Analysing and visualising results

148 From the results returned, summary measures about how the ecosystem
149 stable states respond to environmental change can be extracted. The function

150 `get_stability_measures()` returns quantities such as the amount of environ-
151 mental change required for the system to abruptly change to a different state.

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

156 4. Impact

157 The open source implementation and extension of the model used in Bush
158 et al. [2] provides the means of reproducing the results published while at
159 the same time provides the means of doing unique, innovative, and important
160 investigations of how ecosystems respond to environmental change, and how
161 biodiversity may modulate this response.

162 The design of the package code and functionality is with reproducibility in
163 mind: the combination of all parameters being in a single parameter object
164 as well as the return of the simulation as a result object which inherits from
165 the parameter object provides a relatively easy to use framework to implement
166 reproducible experiments.

167 Here we evidence the impact of the *microxanox* package by describing three
168 use cases and then by describing how the package can be a starting point for
169 models of other ecosystems. The first two use cases are described in detail
170 (including the code for reproducing them) in the *User Guide* and the *Partial*
171 *Reproduction* vignettes. The third is taken from Limberger et al. [6] and Petchey
172 et al. [8].

173 4.1. Use case 1: Regime shifts during temporal environmental change

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

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

189 Also visible in the results are thick lines showing abundances of bacteria
190 when abundances are low. This is due to the implementation of a function that
191 at regular intervals, increases the abundance to a preset level. This prevents

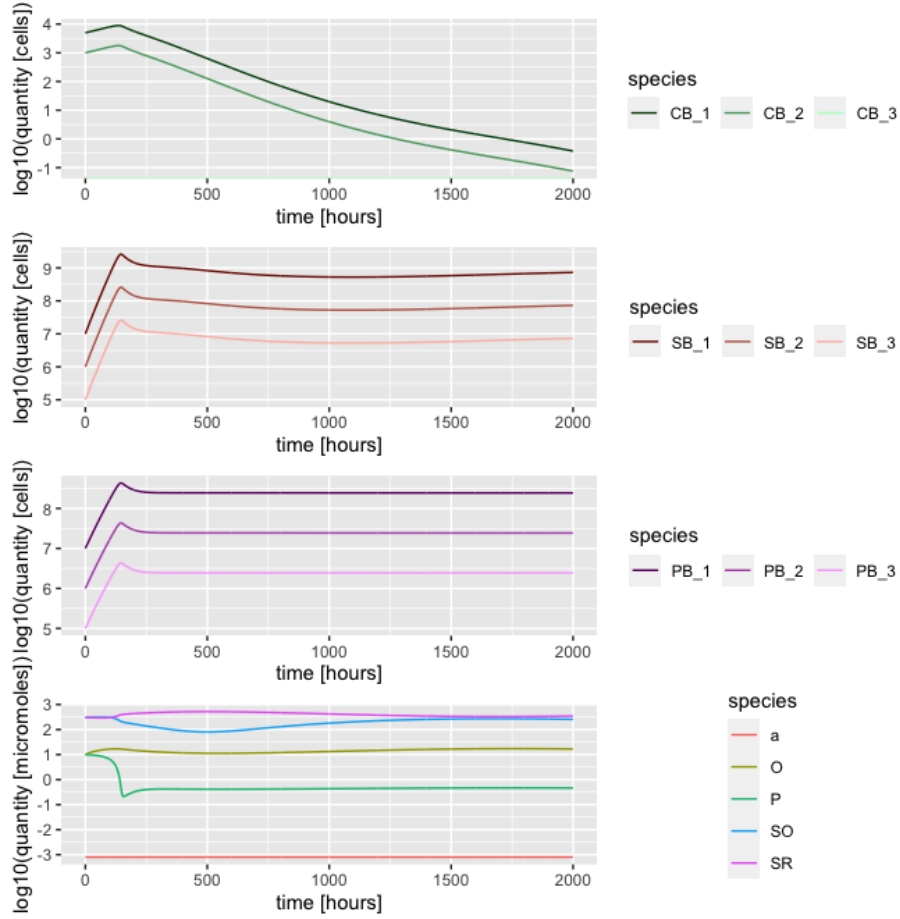


Figure 2: Results of a simulation run shown using the function *plot_dynamics()*. In this case, there were three strains per functional group, though strains within functional groups had identical properties in this example. CB_1 = cyanobacteria strain 1; SB_1 = sulfur reducing bacteria stain 1; PB_1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

192 abundances reducing to very small numbers. The function that implements this
193 increase abundance can also be made to add a certain abundance to each strain
194 at regular intervals, thus simulating immigration in to the system.

195 4.2. Use case 2: The extent of hysteresis depends on community composition

196 The package contains a function to extract summary features of ecosystem
197 responses to environmental change, such as the amount of hysteresis displayed by
198 the ecosystem. Hysteresis is a key feature of ecosystem responses to environmental
199 change, because it is related to how difficult it can be to reverse the effects of
200 environmental change [10]. The amount of hysteresis is measured as the extent
201 of the environmental condition (here oxygen diffusivity) for which there were
202 two stable states. I.e. it is the extent of the environmental conditions for which
203 historical conditions play an important role in determining the current system
204 state (a definition of hysteresis).

205 Using the package to calculate the extent of hysteresis involves setting
206 ecosystem and simulation parameters, including parameters for the finding of
207 stable states across an environmental gradient, running the stable state finding
208 function, and analysing the results with the function that calculates extent of
209 hysteresis. The code for this is provided in the *User Guide*.

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

215 4.3. Use case 3: Effects of functional diversity on regime shifts

216 As discussed in the Introduction section, the package was motivated by the
217 question of how biodiversity influences ecosystem responses to environmental
218 change. Extensive results concerning this question are given in a separate
219 publication Limberger et al. [6]. Here we describe one of the results, which is
220 that having biodiversity in a functional group can allow state changes to occur
221 that otherwise would not have. I.e. biodiversity can qualitatively change the
222 state of the ecosystem.

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

229 Figure 5 shows a simulation with two species (strains) in each of the three
230 functional groups. The ecosystem starts in the oxic state, though with relatively
231 high abundance of each functional group. The strain of SB that is more tolerant
232 to oxygen (SB_1) initially decreases in abundance, but then increases, and the
233 other (SB_2) strain then becomes abundant and SB_1 declines. Furthermore,
234 the cyanobacteria crash in abundance, and the system switches to the anoxic



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.

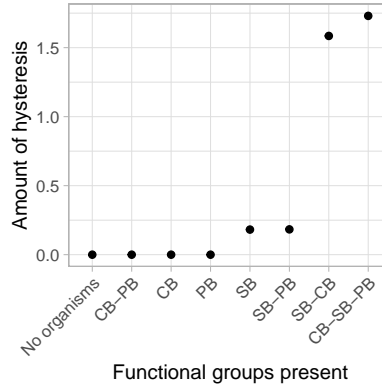


Figure 4: The amount of hysteresis depends on the combination of types of organisms present. The model is entirely deterministic, hence there are no error bars.

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

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

5. Conclusions

The *microxanox* R package allows the simulation, visualisation, and analysis of a model of a microbial ecosystem while allowing variation in the amount of diversity containing in each of the functional groups of organisms present. It has been used for the research described in another paper that describes one of the first investigations of the effects of diversity on ecosystem resilience Limberger et al. [6]. In that paper, we show that diversity can have large and important effects of ecosystem responses, highlighting the need for models such as ours, with which one can easily manipulate the amount of biodiversity. The

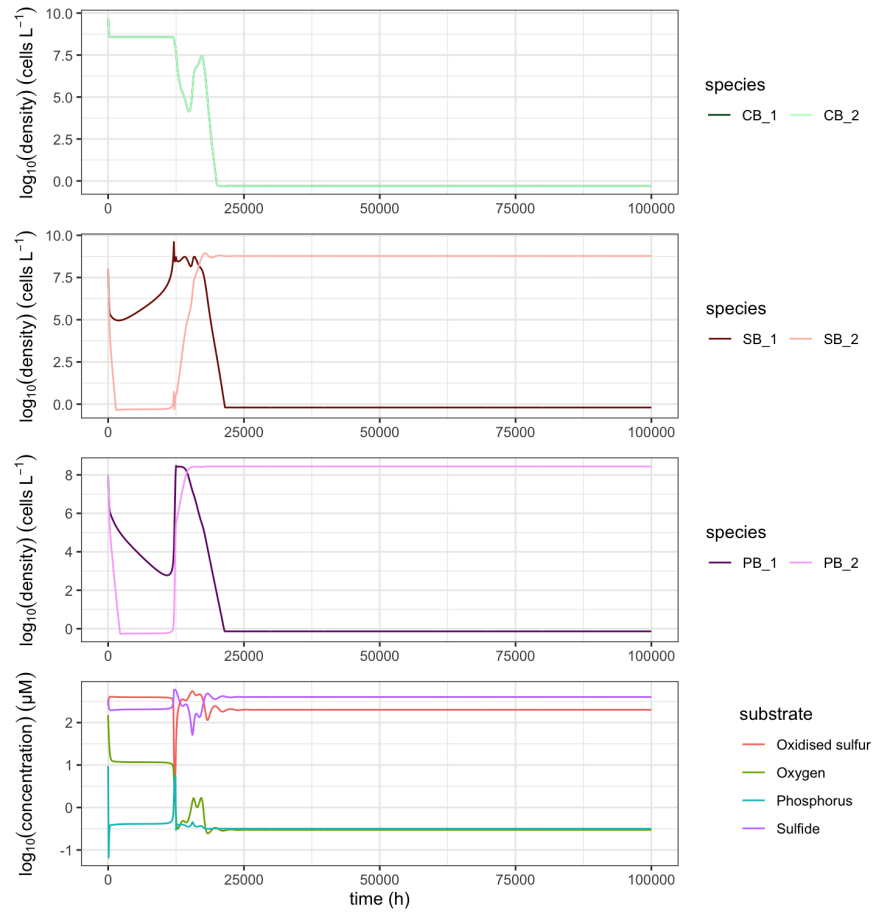


Figure 5: The dynamics of the ecosystem when there are two species in each functional group, and some variation (diversity) in species parameters.

261 *microxanox* package has also been used to reproduce the results of the paper
262 that inspired the package development [2].

263 The package greatly lowers the amount of work required in further investi-
264 gations of the specific ecosystem modelled. There has, for example, been quite
265 limited investigation of how biodiversity influences the short-term responses of
266 the modelled ecosystem to environmental change. Likewise, the package could be
267 used to power an investigation of the effects of biodiversity on the usefulness of
268 early warning signals of abrupt ecosystem change [11]. In addition this package
269 could be used as a template for the implementation for developing models of
270 other types of ecosystems and organism. By doing so, other models can profit
271 from the overall framework used, and the reproducibility aspects as well as the
272 flexibility implemented.

273 6. Conflict of Interest

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

277 7. Acknowledgements

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

281 References

- 282 [1] Amrei Binzer, Christian Guill, Björn C. Rall, and Ulrich Brose. Interactive
283 effects of warming, eutrophication and size structure: Impacts on biodiversity
284 and food-web structure. *Global Change Biology*, 22(1):220–227, January
285 2016. ISSN 13541013. doi: 10.1111/gcb.13086.
- 286 [2] Timothy Bush, Muhe Diao, Rosalind J. Allen, Ruben Sinnige, Gerard
287 Muyzer, and Jef Huisman. Oxidic-anoxic regime shifts mediated by feed-
288 backs between biogeochemical processes and microbial community dynamics.
289 *Nature Communications*, 8(1):789, October 2017. ISSN 2041-1723. doi:
290 10.1038/s41467-017-00912-x.
- 291 [3] P. Catalina Chaparro-Pedraza. Fast environmental change and eco-
292 evolutionary feedbacks can drive regime shifts in ecosystems before tip-
293 ping points are crossed. *Proceedings of the Royal Society B: Biological*
294 *Sciences*, 288(1955):20211192, July 2021. ISSN 0962-8452, 1471-2954. doi:
295 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] Rainer M Krug and Owen L Petchey. Microxanox: Model to simulate three functional group system. June 2022. doi: 10.5281/ZENODO.6624125.
- [6] Romana Limberger, Uriah Daugaard, Anubhav Gupta, Rainer Krug, Kimberley Lemmen, Sofia van Moorsel, Marcel Suleiman, Debra Zuppinger-Dingley, and Owen Petchey. Functional diversity can facilitate the collapse of an undesirable ecosystem state. *Authorea Preprint*, June 2022. doi: 10.22541/au.165513066.66785786/v1.
- [7] 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.
- [8] Owen L Petchey, Uriah Daugaard, Anubhav Gupta, Rainer M Krug, Kimberley D. Lemmen, van Moorsel, Sofia J., Marcel Suleiman, Zuppinger-Dingley, Debra, and Romana Limberger. Code package for: "Functional diversity can facilitate the collapse of an undesirable ecosystem state". Zenodo, June 2022.
- [9] R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2022.
- [10] Marten Scheffer, Steve Carpenter, Jonathan A. Foley, Carl Folke, and Brian Walker. Catastrophic shifts in ecosystems. *Nature*, 413(6856):591–596, October 2001. ISSN 0028-0836, 1476-4687. doi: 10.1038/35098000.
- [11] Marten Scheffer, Jordi Bascompte, William A. Brock, Victor Brovkin, Stephen R. Carpenter, Vasilis Dakos, Hermann Held, Egbert H. van Nes, Max Rietkerk, and George Sugihara. Early-warning signals for critical transitions. *Nature*, 461(7260):53–59, September 2009. ISSN 0028-0836, 1476-4687. doi: 10.1038/nature08227.
- [12] Karline Soetaert, Thomas Petzoldt, and R. Woodrow Setzer. Solving differential equations in R: Package deSolve. *Journal of Statistical Software*, 33(9):1–25, 2010. doi: 10.18637/jss.v033.i09.
- [13] John Vandermeer and Ivette Perfecto. Hysteresis and critical transitions in a coffee agroecosystem. *Proceedings of the National Academy of Sciences*, 116(30):15074–15079, July 2019. ISSN 0027-8424, 1091-6490. doi: 10.1073/pnas.1902773116.
- [14] David A. Vasseur and Kevin S. McCann. A Mechanistic Approach for Modeling Temperature-Dependent Consumer-Resource Dynamics. *The American Naturalist*, 166(2):184–198, August 2005. ISSN 0003-0147, 1537-5323. doi: 10.1086/431285.