

1 MicroxanoX: an R package for simulating an aquatic
2 *MICRO*bial 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 steady 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, steady 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 the understanding about how and why ecosystems change when the environ-
 14 ment changes [1, 3, 14]. They are also important for developing hypotheses to test
 15 in empirical studies. One area of ecology in which models have been influential
 16 is the understanding of their response to a gradual change in an environmental
 17 driver [10]. An environmental driver is an environmental condition that affects
 18 an ecosystem. Still, it is assumed not to be affected by the ecosystem, such as
 19 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 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 in the system is the switch from an aerobic (oxygen is available for
28 metabolism) to an 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 oxygen diffusivity) could cause
35 catastrophic changes in the ecosystem state that would be difficult to reverse.

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

43 We decided to fill this research gap by making a simulation study of how
44 within functional group biodiversity affects ecosystem responses to environmental
45 change Limberger et al. [6], and to base our work on the work and model of
46 Bush et al. [2]. With this goal in mind, we developed the `microxanox` package
47 [5]. The first stage of development was to write code from scratch (as there was
48 no available code to start from) and to confirm that this new implementation
49 would reproduce the previously published results. The resulting reproduction is
50 available as one of the package vignettes: [vignette Partial reproduction of Bush](#)
51 [et al.](#)

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

62 3. Software description

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

73 To make the simulation run with multiple species per functional group, we
74 expressed different species characteristics in the elements of vectors and matrices.
75 We also coded the ordinary differential equations to include the vectors and
76 matrices and used matrix mathematics. In this way, we made it possible to
77 run simulations with different numbers of species without having to change the
78 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 [8] to Limberger et al. [6] for an example of how this is used). As such,
89 all the parameters needed to run a simulation or find a steady state (i.e. the
90 final state of the ecosystem) are contained in a single object (which can easily be
91 created using included functions). This parameter object is given to a function
92 that runs the simulations and returns the results. The returned results object is
93 identical to the parameter object but with an additional slot named `results`,
94 which contains the simulation results. Thus the returned results object contains
95 the simulation conditions (parameters) as well as the results, and can be used to
96 run the simulation again. This promotes reproducibility and makes incremental
97 changes of individual parameters with a consecutive re-running of the simulations
98 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].



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.

105 The `run_simulation()` function needs only one argument - an object as created
 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 identical to the parameter object, ex-
 114 cept of an additional slot containing the results. This design produces a fully
 115 reproducible object as it can be used as a parameter object to be fed back into the
 116 `run_simulation()` function to rerun the simulation again from the parameter
 117 used to generate the results before.

118 3.3. Finding Steady States

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

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

132 The second method runs two simulations, one with step-wise and slowly
 133 temporally *increasing* oxygen diffusivity, and the other with step-wise and slowly
 134 *decreasing* oxygen diffusivity. (Put another way, to explore the sensitivity of
 135 steady states of the dynamic model under oxygen diffusivity variation, subset
 136 values of this parameter are numerically evaluated. Two value vectors of oxygen
 137 diffusivity are presented step-wise, gradually increasing and decreasing.) During
 138 this temporal environmental change, the state of the system is recorded just
 139 before change to a new oxygen diffusivity (we term this the *Temporal method*).

140 The package does not include methods for a formal analysis of the stability
 141 of the system and users should take care to assess if steady states are unique,
 142 and indeed if a steady state has been achieved.

143 The replication method is implemented in the function `run_replication_ssfind()`
 144 which takes a parameter object as returned by the function `new_replication_ssfind_parameter()`
 145 and the number of cores for multithreading the simulation. As the multithreading
 146 uses the package function `mclapply()` from the package `parallel` [9], the multi-
 147 threading only works on Linux and Mac. It is planned to move to `parLapply()`
 148 [9] in a future release.

149 The temporal method is implemented in the function `run_temporal_ssfind()`,
 150 which takes a parameter object as created by the function `new_temporal_ssfind_parameter()`.
 151 It is planned for a later release, to run these two simulations in parallel.
 152 For a more detailed walk-through of these two approaches and explanation
 153 please see the [User Guide](#).

154 3.4. Analysing and visualising results

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

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

163 4. Impact

164 The open source implementation and extension of the model used in Bush
 165 et al. [2] provides the means of reproducing the results published while at
 166 the same time provides the means of doing unique, innovative, and important
 167 investigations of how ecosystems change as the environmental changes, and how
 168 biodiversity may modulate this.

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

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

180 4.1. Use case 1: Regime shifts during temporal environmental change

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

188 An example of this functionality is given in the *Partial Reproduction* vignette,
 189 which we briefly show here (Figure 3). The example is composed of a single

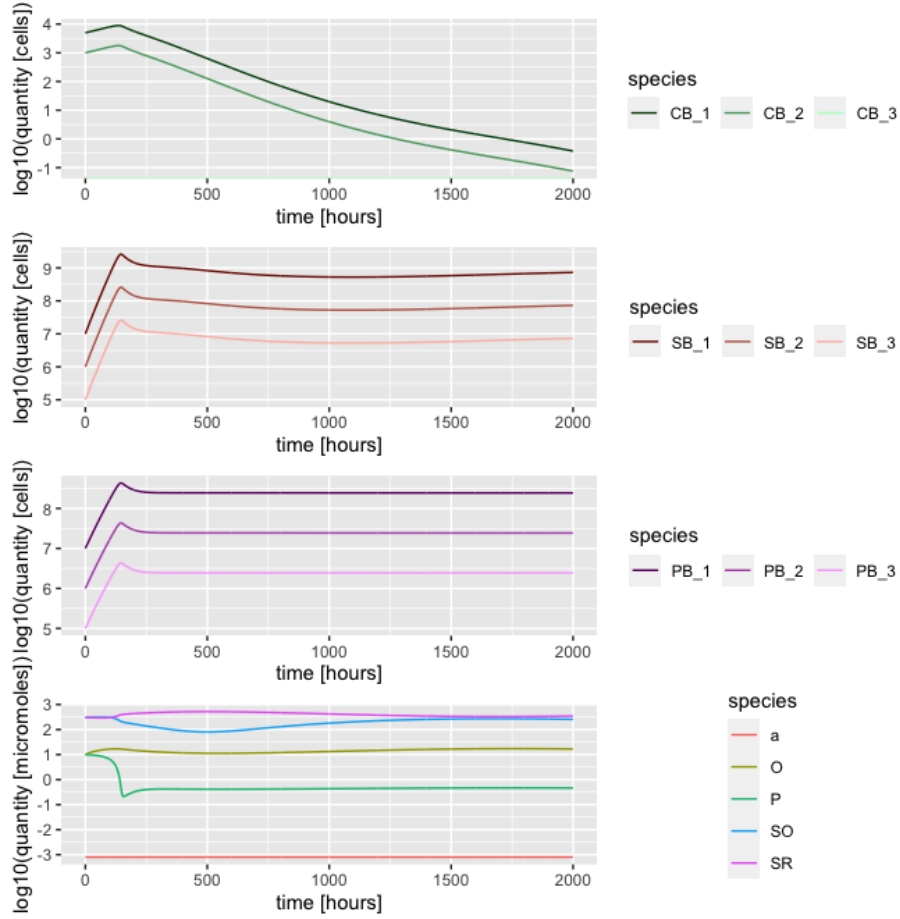


Figure 2: Results of a simulation run shown using the function `plot_dynamics()`. In this case, there were three strains per functional group, though strains within functional groups had identical properties in this example. CB_1 = cyanobacteria strain 1; SB_1 = sulfur reducing bacteria stain 1; PB_1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

simulation, at the beginning of which the system is in the oxic state with high abundance of cyanobacteria. Oxygen diffusivity is then slowly decreased and eventually, around hour 30'000 the system switches to the anoxic state, with high abundance of both sulfur bacteria types. The oxygen diffusivity is then increased and at around hour 38'000 the system abruptly switches back to the oxic state.

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

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

The package contains a function to extract summary features of ecosystem responses to environmental change, such as the amount of hysteresis displayed by the ecosystem. Hysteresis is a key feature of ecosystem responses to environmental change, because it is related to how difficult it can be to reverse the effects of environmental change [10]. The amount of hysteresis is measured as the extent of the environmental condition (here oxygen diffusivity) for which there were two steady 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 (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 steady states across an environmental gradient, running the steady 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. [6]. 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



Figure 3: The temporal dynamics of the ecosystem model when an environmental condition (here parameter a , the oxygen diffusivity) changes. Plot of the steady 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.

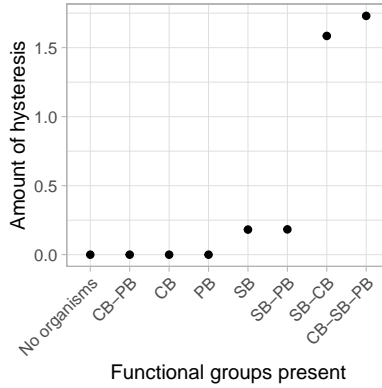


Figure 4: The amount of hysteresis depends on the combination of types of organisms present. The model is entirely deterministic, hence there are no error bars.

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 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.

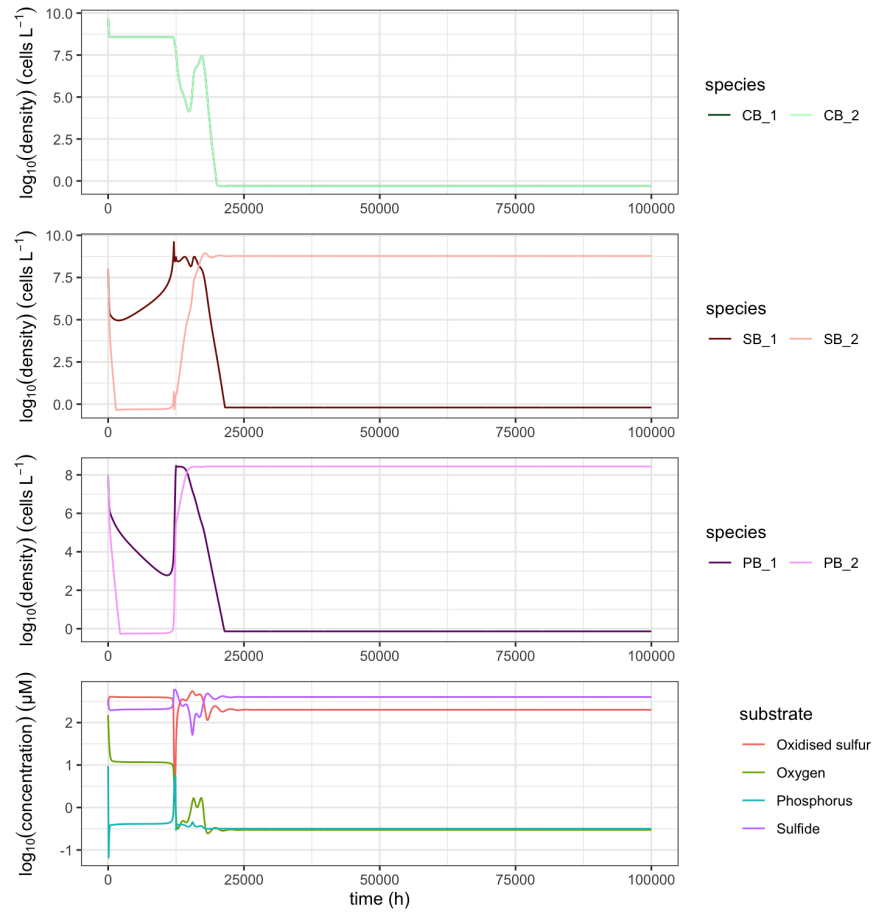


Figure 5: The dynamics of the ecosystem when there are two species in each functional group, and some variation (diversity) in species parameters.

259 5. Conclusions

260 The *microxanox* R package allows the simulation, visualisation, and analysis
261 of a model of a microbial ecosystem while allowing variation in the amount
262 of diversity containing in each of the functional groups of organisms present.
263 It has been used for the research described in another paper that describes
264 one of the first investigations of the effects of diversity on ecosystem resilience
265 Limberger et al. [6]. In that paper, we show that diversity can have large and
266 important effects of ecosystem responses, highlighting the need for models such
267 as ours, with which one can easily manipulate the amount of biodiversity. The
268 *microxanox* package has also been used to reproduce the results of the paper
269 that inspired the package development [2].

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

280 6. Conflict of Interest

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

284 7. Acknowledgements

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

288 References

- 289 [1] Amrei Binzer, Christian Guill, Björn C. Rall, and Ulrich Brose. Interactive
290 effects of warming, eutrophication and size structure: Impacts on biodiversity
291 and food-web structure. *Global Change Biology*, 22(1):220–227, January
292 2016. ISSN 13541013. doi: 10.1111/gcb.13086.
- 293 [2] Timothy Bush, Muhe Diao, Rosalind J. Allen, Ruben Sinnige, Gerard
294 Muyzer, and Jef Huisman. Oxic-anoxic regime shifts mediated by feed-
295 backs between biogeochemical processes and microbial community dynamics.
296 *Nature Communications*, 8(1):789, October 2017. ISSN 2041-1723. doi:
297 10.1038/s41467-017-00912-x.

- [3] P. Catalina Chaparro-Pedraza. Fast environmental change and 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] 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 M. Krug, Kimberley D. Lemmen, Sofia J. Van Moorsel, Marcel Suleiman, Debra Zuppinger-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.
- [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.

- 336 [13] John Vandermeer and Ivette Perfecto. Hysteresis and critical transitions in
337 a coffee agroecosystem. *Proceedings of the National Academy of Sciences*,
338 116(30):15074–15079, July 2019. ISSN 0027-8424, 1091-6490. doi: 10.1073/
339 pnas.1902773116.
- 340 [14] David A. Vasseur and Kevin S. McCann. A Mechanistic Approach for
341 Modeling Temperature-Dependent Consumer-Resource Dynamics. *The*
342 *American Naturalist*, 166(2):184–198, August 2005. ISSN 0003-0147, 1537-
343 5323. doi: 10.1086/431285.