

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 system (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 features multiple species per functional group, where the number is only limited by computational constraints. The R 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.

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

8 **1. TODO**

- 9 • **update links**
10 ☒ ~~update cross references~~
11 ☒ ~~add abstract~~
12 ☒ ~~check keywords~~
13 ☒ ~~add more references~~
14 ☒ ~~expand on Impact section~~
15 ☒ ~~expand on Conclusion section~~

*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

16 **2. Required Metadata**

17 *2.1. Current code version*

18 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
C3	Code Ocean compute capsule	
C4	Legal Code License	CC BY 4.0
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	TO BE ADDED
C9	Support email for questions	Rainer.Krug@uzh.ch; Rainer@krugs.de

19 **3. Motivation and significance**

20 Mathematical models play a key role in the development of understanding
 21 about how ecosystems work and how they respond to environmental changes
 22 [1, 3, 13]. They are also important for developing hypotheses that can then be
 23 tested in empirical studies. One area of ecology in which simple models have
 24 played a influential role is the area of how ecosystems respond to gradual change
 25 in an environmental driver [9]. An environmental driver is a environmental
 26 condition that affects an ecosystem, but is assumed to not be affected by the
 27 ecosystem, such as the rate of nutrient input into a lake.

28 It is conceivable that an ecosystem state, such as the total biomass of a
 29 particular type of bacteria, may remain unchanged when an environmental driver

30 changes. It is also possible that the ecosystem state changes gradually. It is
31 also possible that the ecosystem state changes abruptly to a new state that is
32 difficult to recover from [9]. This possibility for abrupt, perhaps catastrophic
33 changes that are difficult to reverse causes considerable concern [4, 6, 12].

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

45 One feature of the study by Bush et al. [2] was limited biodiversity. Specifi-
46 cally, there was no biodiversity within each of the three types of bacteria. This
47 leaves open the question of if and how biodiversity within these types (i.e. func-
48 tional groups) of bacteria, in their model, affects the ecosystem response to
49 environmental change. This limitation is not specific to the study of Bush et al.
50 [2], there are few if any studies of the effects of biodiversity on abrupt transitions
51 between ecosystem states.

52 We decided to fill this research gap by making a simulation study of how
53 within functional group biodiversity affects ecosystem responses to environmental
54 change Limberger [5], and to base our work on the work and model of Bush et al.
55 [2]. It was with this goal in mind that we developed the *microxanox* package.
56 The first stage of development was to write code from scratch (as there was
57 no available code to start from) and to confirm that this new implementation
58 would reproduce the previously published results. The resulting reproduction is
59 available as one of the package vignettes: [vignette Partial reproduction of Bush](#)
60 [et al.](#)

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

71 4. Software description

72 The *microxanox* package is for simulating a three functional group system
73 (*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 Bush et al. [2] (See Bush et al. [2] 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 for running a set of simulations across, for example, a range of environmental conditions.

To make the simulation run with multiple species, 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 to use 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 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).

4.1. Software architecture

The framework used when writing this package aimed to maximise simplicity for the user, and to make it easy to reproduce results (see the supplement to Limberger [5] for an example of how this is used). As such, all the parameters needed to run a simulation or find a stable state (i.e. the final state of the ecosystem) are contained in a single object (which can easily be created using included functions). This parameter object is given to a function that runs the simulations and returns the results. The returned results object is identical to the parameter object but with an additional slot named `results`, which contains the results of the run. Thus the returned results object contains the simulation conditions (parameters) as well as the results, and can be used to run the simulation again. This promotes reproducibility and makes incremental changes of individual parameters and re-running the simulations straightforward.

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

4.2. Running one simulation

A typical simulation would look as shown in 1.

A simulation is run using the `run_simulation()` function. In this function, the ODEs are solved using the function `ode()` in the package *deSolve* package [11]. The `run_simulation()` function needs only one argument - an object as created by the function `new_runsim_parameter()`. The parameter object returned by `new_runsim_parameter()` contains among other things the `strain_parameter` object, which can be created by the function `new_strain_parameter()`. For a detailed description of the parameter objects, their meaning and how they



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.

117 are created and have values set and changed please see the *User Guide* which
118 accompanies the package or is available at [User Guide](#)

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

125 4.3. Finding Stable States

126 The general approach used to find the stable state of the system with a
127 specific parameter set is to run the simulation for a long time and record the
128 final state. When one does this across a range of environmental conditions, one
129 discovers how the steady state of the system responds to the environmental
130 conditions. The package contains functionality for finding steady states that
131 correspond to values of one environmental driver, namely the value of oxygen
132 diffusivity.

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

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` [7], the multi-
147 threading only works on Linux and Mac. It is planned to move to `parLapply()`
148 [7] in a future release.

149 This temporal method 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 4.4. Analysing and visualising results

155 From the results returned, summary measures about how the ecosystem
156 stable states respond to 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 2.

163 5. 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 respond to environmental change, and how
168 biodiversity may modulate this response.

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 REF NEEDED [8].

179 5.1. Use case 1: Regime shifts during temporal environmental change

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

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

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

```
201 #>  
202 #> Attaching package: 'dplyr'
```

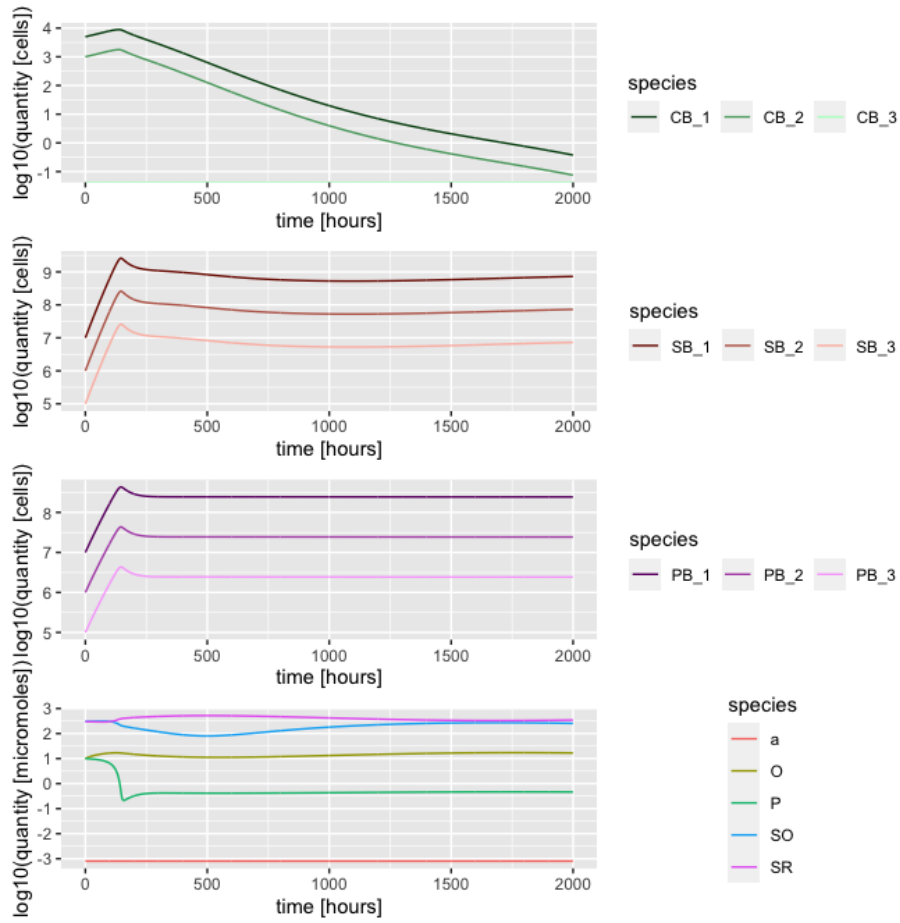


Figure 2: Results of a simulation run shown using the function `plot_dynamics()`. In this case, there were three strains per functional group, though strains within functional groups had identical properties in this example. CB extunderscore1 = cyanobacteria strain 1; SB extunderscore1 = sulfur reducing bacteria stain 1; PB = phototrophic sulfur bacteria strain 1. Here we show the figure produce by `plot_dynamics()`, which produces figures not with all features ideal for publication.


```

203 #> The following objects are masked from 'package:stats':
204 #>
205 #>     filter, lag
206 #> The following objects are masked from 'package:base':
207 #>
208 #>     intersect, setdiff, setequal, union
209 #> Warning in knitr::include_graphics(fig): It is highly recommended
210 #> to use relative paths for images. You had absolute paths: "/"
211 #> Users/rainer/git/0.RPackages/microxanox/inst/manuscript/figures/
212 #> gen_uc1_partrep_temporal_state_switching.pdf"

```

213 5.2. 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 [9]. The amount of hysteresis is measured as the extent
219 of the environmental condition (here oxygen diffusivity) for which there were
220 two stable 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 (a definition of hysteresis).

223 Using the package to calculate the extent of hysteresis involves setting
224 ecosystem and simulation parameters, including parameters for the finding of
225 stable states across an environmental gradient, running the stable 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* vignette.

228 The results show that the amount of hysteresis depends greatly on the
229 combinations of organisms present (Figure ??figL:uc2)). For example, with
230 only the CB (cyanobacteria) present, there was no hysteresis. In contrast, the
231 presence of both CB and SB (sulfate reducing bacteria) led to a large amount of
232 hysteresis. (These results are also given in the *Partial Reproduction* vignette.)

```

233 #> Warning in knitr::include_graphics(fig): It is highly recommended
234 #> to use relative paths for images. You had absolute paths: "/"
235 #> Users/rainer/git/0.RPackages/microxanox/inst/manuscript/figures/
236 #> gen_uc2_user_guide_hysteresis.pdf"

```

237 5.3. Use case 3: Effects of functional diversity on regime shifts

238 As discussed in the Introduction section, the package was motivated by the
239 question of how biodiversity influences ecosystem responses to environmental
240 change. Extensive results concerning this question are given in a separate
241 publication Limberger [5]. Here we describe one of the results, which is that
242 having biodiversity in a functional group can allow state changes to occur that



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 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.

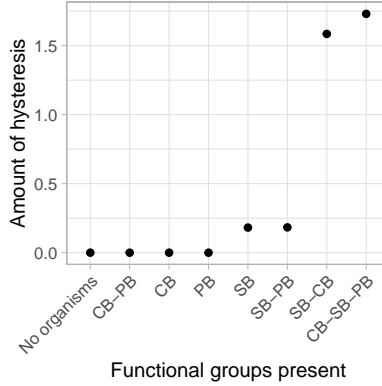


Figure 4: The amount of hysteresis depends on the combination of types of organisms present. The model is entirely deterministic, hence there are no error bars.

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.

5.4. 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 making software design decisions, and rather focus on appropriately representing their ecosystem, and finding the results that interest them. Nevertheless, adaptation of the code

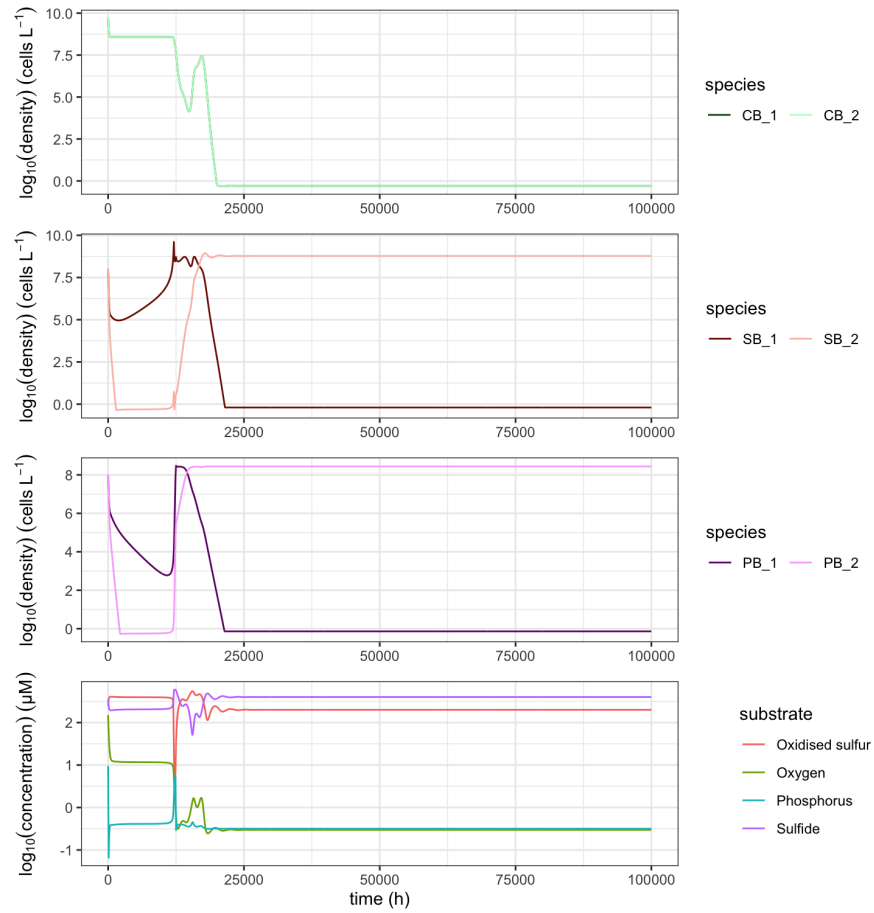


Figure 5: The dynamics of the ecosystem when there are two species in each functional group, and some variation (diversity) in species parameters.

271 in the package will require a person / persons that can take a conceptual model of
272 an ecosystem and then represent that in terms of parameters and rate equations,
273 and that is relatively proficient in R programming.

274 6. Conclusions

275 The *microxanoz* R package allows the simulation, visualisation, and analysis
276 of a model of a microbial ecosystem while allowing variation in the amount of
277 diversity containing in each of the functional groups of organisms present. It has
278 been used for the research described in another paper that describes one of the
279 first investigations of the effects of diversity on ecosystem resilience Limberger
280 [5]. In that paper, we show that diversity can have large and important effects
281 of ecosystem responses, highlighting the need for models such as ours, with
282 which one can easily manipulate the amount of biodiversity. The *microxanoz*
283 package has also been used to reproduce the results of the paper that inspired
284 the package development Bush et al. [2].

285 The package greatly lowers the amount of work required in further investi-
286 gations of the specific ecosystem modelled. There has, for example, been quite
287 limited investigation of how biodiversity influences the short-term responses of
288 the modelled ecosystem to environmental change. Likewise, the package could be
289 used to power an investigation of the effects of biodiversity on the usefulness of
290 early warning signals of abrupt ecosystem change [10]. In addition this package
291 could be used as a template for the implementation for developing models of
292 other types of ecosystems and organism. By doing so, other models can profit
293 from the overall framework used, and the reproducibility aspects as well as the
294 flexibility implemented.

295 7. Conflict of Interest

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

299 8. Acknowledgements

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

303 References

- 304 [1] Amrei Binzer, Christian Guill, Björn C. Rall, and Ulrich Brose. Interactive
305 effects of warming, eutrophication and size structure: Impacts on biodiversity
306 and food-web structure. *Global Change Biology*, 22(1):220–227, January
307 2016. ISSN 13541013. doi: 10.1111/gcb.13086.

- 308 [2] Timothy Bush, Muhe Diao, Rosalind J. Allen, Ruben Sinnige, Gerard
309 Muyzer, and Jef Huisman. Oxic-anoxic regime shifts mediated by feed-
310 backs between biogeochemical processes and microbial community dynamics.
311 *Nature Communications*, 8(1):789, October 2017. ISSN 2041-1723. doi:
312 10.1038/s41467-017-00912-x.
- 313 [3] P. Catalina Chaparro-Pedraza. Fast environmental change and eco-
314 evolutionary feedbacks can drive regime shifts in ecosystems before tip-
315 ping points are crossed. *Proceedings of the Royal Society B: Biological*
316 *Sciences*, 288(1955):20211192, July 2021. ISSN 0962-8452, 1471-2954. doi:
317 10.1098/rspb.2021.1192.
- 318 [4] Scott L. Collins, Jesse B. Nippert, John M. Blair, John M. Briggs, Pamela
319 Blackmore, and Zak Ratajczak. Fire frequency, state change and hysteresis
320 in tallgrass prairie. *Ecology Letters*, 24(4):636–647, April 2021. ISSN
321 1461-023X, 1461-0248. doi: 10.1111/ele.13676.
- 322 [5] Romana Limberger. ###TO BE UPDATED### - The role of functional
323 diversity in ecosystem regime shifts. 2022.
- 324 [6] Amanda C. Northrop, Vanessa Avalone, Aaron M. Ellison, Bryan A. Ballif,
325 and Nicholas J. Gotelli. Clockwise and counterclockwise hysteresis charac-
326 terize state changes in the same aquatic ecosystem. *Ecology Letters*, 24(1):
327 94–101, January 2021. ISSN 1461-023X, 1461-0248. doi: 10.1111/ele.13625.
- 328 [7] R Core Team. *R: A Language and Environment for Statistical Computing*.
329 R Foundation for Statistical Computing, Vienna, Austria, 2022.
- 330 [8] Author REF NEEDED. REFERNECE NEEDED. *Journal of missing*
331 *references*, 2222.
- 332 [9] Marten Scheffer, Steve Carpenter, Jonathan A. Foley, Carl Folke, and Brian
333 Walker. Catastrophic shifts in ecosystems. *Nature*, 413(6856):591–596,
334 October 2001. ISSN 0028-0836, 1476-4687. doi: 10.1038/35098000.
- 335 [10] Marten Scheffer, Jordi Bascompte, William A. Brock, Victor Brovkin,
336 Stephen R. Carpenter, Vasilis Dakos, Hermann Held, Egbert H. van Nes,
337 Max Rietkerk, and George Sugihara. Early-warning signals for critical
338 transitions. *Nature*, 461(7260):53–59, September 2009. ISSN 0028-0836,
339 1476-4687. doi: 10.1038/nature08227.
- 340 [11] Karline Soetaert, Thomas Petzoldt, and R. Woodrow Setzer. Solving
341 differential equations in R: Package deSolve. *Journal of Statistical Software*,
342 33(9):1–25, 2010. doi: 10.18637/jss.v033.i09.
- 343 [12] John Vandermeer and Ivette Perfecto. Hysteresis and critical transitions in
344 a coffee agroecosystem. *Proceedings of the National Academy of Sciences*,
345 116(30):15074–15079, July 2019. ISSN 0027-8424, 1091-6490. doi: 10.1073/
346 pnas.1902773116.

- 347 [13] David A. Vasseur and Kevin S. McCann. A Mechanistic Approach for
348 Modeling Temperature-Dependent Consumer-Resource Dynamics. *The*
349 *American Naturalist*, 166(2):184–198, August 2005. ISSN 0003-0147, 1537-
350 5323. doi: 10.1086/431285.