

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. The model can be implemented with different numbers of species per functional group. The package is constructed in such a way that the results contain the input parameter used, so that a saved results can be loaded again and the simulation be repeated. Furthermore, the package framework and code should serve as a useful starting point for making simulation models of other types of ecosystems.

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

*Corresponding author

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

¹Corresponding Author

²Equal contribution

1. Required Metadata

1.1. Current code version

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

2. Motivation and significance

Mathematical models play a key role in the development of ecosystem models and in increasing the understanding of how and why ecosystems change when the environment changes [1, 3, 17]. They are also important for developing hypotheses

15 to test in empirical studies. One area of ecology in which models have been
16 influential is the understanding of ecosystem responses to gradual change in an
17 environmental driver [12]. An environmental driver is an environmental condition
18 that affects an ecosystem, but which is not itself affected by the ecosystem. An
19 example driver would be the amount of nutrients entering a lake from run-off
20 from surrounding agricultural land.

21 It is conceivable that an ecosystem state, such as the total biomass of a
22 particular type of bacteria, may remain unchanged when an environmental driver
23 changes. It is also possible that the ecosystem state changes gradually. It is
24 also possible that the ecosystem state changes abruptly to a new state that is
25 difficult to recover from [12]. This possibility for abrupt, perhaps catastrophic
26 changes, which are difficult to reverse, causes considerable concern [4, 9, 16].

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

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

44 We decided to fill this research gap by making a simulation study of how
45 within functional group biodiversity affects ecosystem responses to environmental
46 change Limberger et al. [8], and to base our work on the work and model of
47 Bush et al. [2]. With this goal in mind, we developed the `microxanox` package
48 [7]. The first stage of development was to write code from scratch (as there was
49 no available code to start from) and to confirm that this new implementation
50 would reproduce the previously published results. The resulting reproduction is
51 available as one of the package vignettes: [vignette Partial reproduction of Bush](#)
52 [et al.](#)

53 The second stage was to add functionality that would be necessary to answer
54 our research question. Most importantly, we made it possible to have multiple
55 species of bacteria within each of the three functional groups, for the multiple
56 species to differ in their characteristics, and to vary the number of species and
57 amount of variability among them. We also added functionality that allowed:
58 temporally varying environmental conditions, the addition of random noise to
59 state variables, and immigration. In addition to the model itself, the package
60 includes some functions to analyse the results and visualize these to provide a

61 starting point for customized visualizations based on own requirements. The
62 basic and additional functionality is described in the package [User Guide](#).

63 3. Software description

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

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

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

86 3.1. Software architecture

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

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



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.

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 [14].
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. Examining ecosystem responses to environmental change*

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

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

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

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

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

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

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

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

163 3.4. Analysing and visualising results

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

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

172 4. Impact

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

178 The design of the package code and functionality is with reproducibility in
179 mind (for other example of reproducibility directed software, see e.g. Correndo
180 et al. [5], Spillner et al. [15] or Fox et al. [6]): the combination of all parameters
181 being in a single parameter object and the return of the simulation as a result
182 object which inherits from the parameter object, provides a relatively easy-to-use
183 framework to implement reproducible experiments.

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

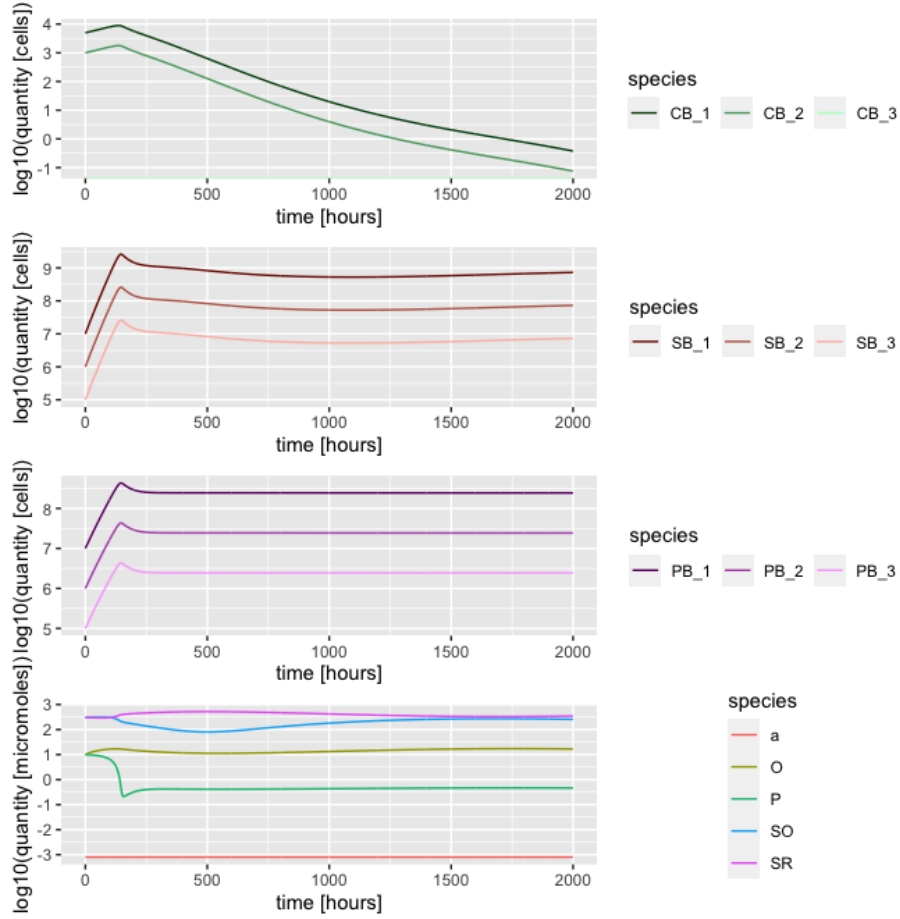


Figure 2: Results of a simulation run shown using the function `plot_dynamics()`. In this case, there were three strains per functional group, though strains within functional groups had identical properties in this example. CB_1 = cyanobacteria strain 1; SB_1 = sulfur reducing bacteria stain 1; PB_1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

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

190 4.1. Use case 1: Regime shifts during temporal environmental change

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

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

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

212 4.2. Use case 2: The extent of hysteresis depends on community composition

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

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

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



Figure 3: The temporal dynamics of the ecosystem model when an environmental condition (here parameter a , the oxygen diffusivity) changes. Plot of the final states of the simulation runs under different oxygen diffusivity. In this simulation there is only one strain in each functional group. CB extunderscore1 = cyanobacteria strain 1; SB extunderscore1 = sulfur reducing bacteria strain 1; PB = phototrophic sulfur bacteria strain 1. Here we show a figure adapted from the output of the `plot_dynamics()` function.

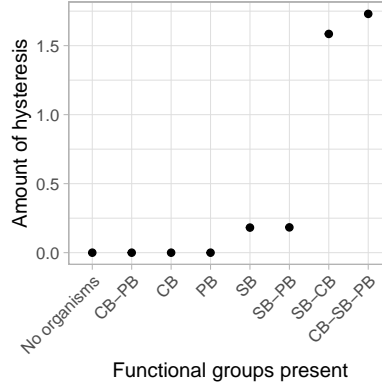


Figure 4: The amount of hysteresis depends on the combination of types of organisms present. The model is entirely deterministic, hence there are no error bars.

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

As discussed in the Introduction section, the package was motivated by the question of how biodiversity influences ecosystem responses to environmental change. Extensive results concerning this question are given in a separate publication Limberger et al. [8]. Here we describe one of the results, which is that having biodiversity in a functional group can allow state changes to occur that otherwise would not have, i.e. biodiversity can qualitatively change the state of the ecosystem.

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

Figure 5 shows a simulation with two species (strains) in each of the three functional groups. The ecosystem starts in the oxic state, though with relatively high abundance of each functional group. The strain of SB that is more tolerant to oxygen (SB_1) initially decreases in abundance, but then increases, and the other (SB_2) strain then becomes abundant and SB_1 declines. Furthermore, the cyanobacteria crash in abundance, and the system switches to the anoxic state. In contrast, if there are two identical strains with tolerance half way between those in Figure 5 the ecosystem remains in the oxic state.

4.4. Adapting for other ecosystems and organisms

We anticipate that the package can be a useful starting point for investigating models of other types of ecosystem and how biodiversity in them affects responses to environmental change. The overall framework of the package, the purpose of each function, and the objects used for storing parameters and results could be retained. E.g., all such models would have parameters that differ among

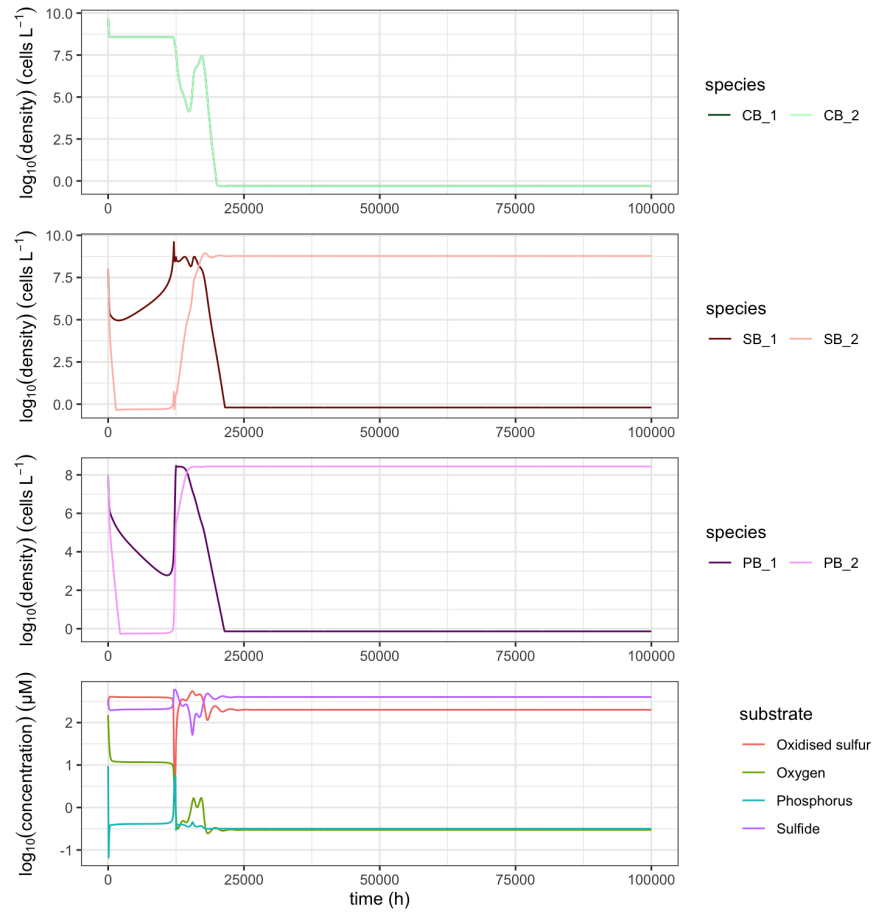


Figure 5: The dynamics of the ecosystem when there are two species in each functional group, and some variation (diversity) in species parameters.

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 a person or persons relatively proficient in R programming.

5. Conclusions

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

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

6. Conflict of Interest

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

7. Acknowledgements

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

297 References

- 298 [1] Amrei Binzer, Christian Guill, Björn C. Rall, and Ulrich Brose. Interactive
299 effects of warming, eutrophication and size structure: Impacts on biodiversity
300 and food-web structure. *Global Change Biology*, 22(1):220–227, January
301 2016. ISSN 13541013. doi: 10.1111/gcb.13086.
- 302 [2] Timothy Bush, Muhe Diao, Rosalind J. Allen, Ruben Sinnige, Gerard
303 Muyzer, and Jef Huisman. Oxidic-anoxic regime shifts mediated by feed-
304 backs between biogeochemical processes and microbial community dynamics.
305 *Nature Communications*, 8(1):789, October 2017. ISSN 2041-1723. doi:
306 10.1038/s41467-017-00912-x.
- 307 [3] P. Catalina Chaparro-Pedraza. Fast environmental change and eco-
308 evolutionary feedbacks can drive regime shifts in ecosystems before tip-
309 ping points are crossed. *Proceedings of the Royal Society B: Biological*
310 *Sciences*, 288(1955):20211192, July 2021. ISSN 0962-8452, 1471-2954. doi:
311 10.1098/rspb.2021.1192.
- 312 [4] Scott L. Collins, Jesse B. Nippert, John M. Blair, John M. Briggs, Pamela
313 Blackmore, and Zak Ratajczak. Fire frequency, state change and hysteresis
314 in tallgrass prairie. *Ecology Letters*, 24(4):636–647, April 2021. ISSN
315 1461-023X, 1461-0248. doi: 10.1111/ele.13676.
- 316 [5] Adrian A. Correndo, Austin Pearce, Carl H. Bolster, John T. Spargo,
317 Deanna Osmond, and Ignacio A. Ciampitti. The soiltestcorr R package:
318 An accessible framework for reproducible correlation analysis of crop yield
319 and soil test data. *SoftwareX*, 21:101275, February 2023. ISSN 23527110.
320 doi: 10.1016/j.softx.2022.101275.
- 321 [6] Nathan Fox, Tom August, Francesca Mancini, Katherine E. Parks, Fel-
322 ix Eigenbrod, James M. Bullock, Louis Sutter, and Laura J. Graham.
323 “photosearcher” package in R: An accessible and reproducible method for
324 harvesting large datasets from Flickr. *SoftwareX*, 12:100624, July 2020.
325 ISSN 23527110. doi: 10.1016/j.softx.2020.100624.
- 326 [7] Rainer M Krug and Owen L Petchey. Microxanox: Model to simulate three
327 functional group system. June 2022. doi: 10.5281/ZENODO.6624125.
- 328 [8] Romana Limberger, Uriah Daugaard, Anubhav Gupta, Rainer M. Krug,
329 Kimberley D. Lemmen, Sofia J. Van Moorsel, Marcel Suleiman, Debra
330 Zuppingier-Dingley, and Owen L. Petchey. Functional diversity can facilitate
331 the collapse of an undesirable ecosystem state. *Ecology Letters*, 26(6):
332 883–895, June 2023. ISSN 1461-023X, 1461-0248. doi: 10.1111/ele.14217.
- 333 [9] Amanda C. Northrop, Vanessa Avalone, Aaron M. Ellison, Bryan A. Ballif,
334 and Nicholas J. Gotelli. Clockwise and counterclockwise hysteresis charac-
335 terize state changes in the same aquatic ecosystem. *Ecology Letters*, 24(1):
336 94–101, January 2021. ISSN 1461-023X, 1461-0248. doi: 10.1111/ele.13625.

- 337 [10] Owen L Petchey, Uriah Daugaard, Anubhav Gupta, Rainer M Krug, Kimber-
338 ley D. Lemmen, van Moorsel, Sofia J., Marcel Suleiman, Zuppinger-Dingley,
339 Debra, and Romana Limberger. Code package for: "Functional diversity
340 can facilitate the collapse of an undesirable ecosystem state". Zenodo, June
341 2022.
- 342 [11] R Core Team. *R: A Language and Environment for Statistical Computing*.
343 R Foundation for Statistical Computing, Vienna, Austria, 2022.
- 344 [12] Marten Scheffer, Steve Carpenter, Jonathan A. Foley, Carl Folke, and Brian
345 Walker. Catastrophic shifts in ecosystems. *Nature*, 413(6856):591–596,
346 October 2001. ISSN 0028-0836, 1476-4687. doi: 10.1038/35098000.
- 347 [13] Marten Scheffer, Jordi Bascompte, William A. Brock, Victor Brovkin,
348 Stephen R. Carpenter, Vasilis Dakos, Hermann Held, Egbert H. van Nes,
349 Max Rietkerk, and George Sugihara. Early-warning signals for critical
350 transitions. *Nature*, 461(7260):53–59, September 2009. ISSN 0028-0836,
351 1476-4687. doi: 10.1038/nature08227.
- 352 [14] Karline Soetaert, Thomas Petzoldt, and R. Woodrow Setzer. Solving
353 differential equations in R: Package deSolve. *Journal of Statistical Software*,
354 33(9):1–25, 2010. doi: 10.18637/jss.v033.i09.
- 355 [15] Josef Spillner, Panagiotis Gkikopoulos, Pamela Delgado, and Christine
356 Choirat. Towards reproducible software studies with MAO and Renku.
357 *SoftwareX*, 17:100947, January 2022. ISSN 23527110. doi: 10.1016/j.softx.
358 2021.100947.
- 359 [16] John Vandermeer and Ivette Perfecto. Hysteresis and critical transitions in
360 a coffee agroecosystem. *Proceedings of the National Academy of Sciences*,
361 116(30):15074–15079, July 2019. ISSN 0027-8424, 1091-6490. doi: 10.1073/
362 pnas.1902773116.
- 363 [17] David A. Vasseur and Kevin S. McCann. A Mechanistic Approach for
364 Modeling Temperature-Dependent Consumer-Resource Dynamics. *The*
365 *American Naturalist*, 166(2):184–198, August 2005. ISSN 0003-0147, 1537-
366 5323. doi: 10.1086/431285.