rend: An R package for Ecological Network Dynamics

Jonathan J. Borrelli

25 March, 2016

Contents

Abstract:

The structure of ecological communities determines how they respond to environmental impacts such as global climate change and anthropogenic disturbance. Despite continually changing environments and numerous different habitat types, the structure of observed ecological communities exhibit some remarkable similarities. Food web structure is generated by the introduction of new species through invasion and speciation, species loss from extinction, and the altering of interactions through foraging decisions and population dynamics. Many packages currently exist to describe the structure (topology) of networks using the R programming languages, but there are none that implement any of the variety of models developed to understand the dynamics of these systems. In this paper I introduce a new (in-development) R-package *rend that allows users to apply a bioenergetic model of multispecies predator-prey dynamics to trophic networks. The* rend package is being developed with the goal of providing stronger links between the structure and dynamics of complex ecological systems.

1 Introduction

A great deal of research in ecology has focused on the dynamics of interacting populations from simple Lotka-Volterra models of predator and prey to much more complex bioenergetic models generalized to any number of predators and their prey. The most basic forms of predator-prey equations take the general form of the following differential equations:

$$\frac{dN}{dt} = f(N) - g(N)P \tag{1}$$

$$\frac{dP}{dt} = eg(N)P - qP \tag{2}$$

where the change in prey (N) abundance is due to prey growth (*f(N)*) and consumption by the predator (*f(N)P*) and predator (P) abundance is increased through consumption (*eg(N)P*) and decreased by death (*qP*). Lotka's (CITATION) and Volterra's (CITATION) equations assumed that the prey grew exponentially, $g(N) = rN$ (where r is the Malthusian growth parameter). Later, and more modern, forms of the equations modified the growth function to incorporate density-dependent growth, $f(N) = rN(1 - N/K)$ (where K is the carrying capacity).

The most widely discussed aspect of these equations is the functional response, *g(N)*, which for Lotka and Volterra took the form αN (where α is the attack rate). The Lotka-Volterra form was later modified by Holling (CITATION) into three functional forms; Type I, Type II, and Type III. Type I functional responses are simply the same as the original Lotka-volterra form. Type II functional responses incorporate an additional parameter, the handling time, which allows for predator satiation (Eq. 3). The Type III form of the equation (Eq. 4) allows for prey refugia, meaning that predators cannot feed at low prey densities.

$$g(N)_{TypeII} = \frac{\alpha N}{1 + \alpha h N} \tag{3}$$

$$g(N)_{TypeIII} = \frac{\alpha N^q}{1 + \alpha h N^q} \tag{4}$$

The Lotka-Volterra and Holling functional responses are known as prey-dependent. The other side of this is the ratio-dependent functional response, which assumes $g(\frac{N}{P})$ rather than $g(N)$ (AG CITATION). Ratio-dependent functional responses have taken several forms, many of which are similar to Holling's Type II. Arditi and Ginzburg suggested a form where the N in the Type II response is simply replaced with $\frac{N}{P}$. Another suggested (Getz CITATION) ratio-dependent functional response takes the form:

$$g(N)_{Getz} = \frac{\alpha N}{cP + N} \tag{5}$$

where c is a constant representing **something from Getz 1984 here**. Arditi and Akcakaya (CITATION) proposed an equation (Eq. 6) that includes a parameter (m) for mutual interference, which is the putative mechanism explaining ratio-dependence.

$$g(N)_{AA} = \frac{\alpha N}{P^m + \alpha h N} \tag{6}$$

After proposing the ratio-dependent functional response, Arditi and Ginzburg (CITATION) suggest that functional responses lie along a continuum. Prey-dependence and ratio-dependence represent opposite sides of the continuum and predator-dependence is somewhere in between. It has also been argued (Abrams and

Ginzburg CITATION) that while predator-dependence is the functional form that most closely represents reality, ratio-dependence is a better approximation than prey-dependence.

The typical predator- prey equation is expressed in terms of population abundances. Yodzis and Innes (1992 CITATION) altered this tradition by developing a bioenergetic model of consumer-resource interactions. Their two-species consumer-resource model was later generalized to N species (McCann CITATION). Bioenergetic models have their basis in energetics and allometries. Many of the parameters of these models can thus be derived from knowledge of the body sizes of the participating organisms. There is a great deal of similarity between bioenergetic models and models based on population abundances. Many of the functional forms of the abundance-based models (described above), such as the Holling functional responses, can be altered to be utilized in the bioenergetic framework.

Most of these simple two-species models can be generalized to apply to multiple species, with many predators and their prey. The simplest, is the generalization of the Lotka-Volterra model,

$$\frac{dN_i(t)}{dt} = N_i \left(b_i + \sum_j a_{ij} N_j \right) \tag{7}$$

where N_i now represents the population abundance of the *ith* species in the system, b_i is the vector of growth rates (if species i is basal) and death rates (for all other species), and a_{ij} is the interaction coefficient (effect of species j on species i). This type of generalization has been made not only for Lotka-Volterra equations, but may incorporate other functional response forms, such as those described above, as well as intra-specific density dependence (i.e., when $a_{ii} < 0$ in the Lotka-Volterra model).

Multi-species predator prey models have been used to assess a variety of hypotheses related to the stability of food webs. They are often used in conjunction with assembly or evolution based models of community development. In these studies researchers build food webs by the introduction of random species from some predefined species pool, or by "speciation" whereby new species are introduced as modified versions of already present species. Multi-species predator-prey models are then employed to determine whether these newly introduced species are able to persist in the web as is, or whether they cause extinctions or go extinct themselves. Assembly- and evolution-based models are typically run until the system reaches some kind of steady-state or equilibrium and the resultant community can then be analyzed and compared to existing empirically described communities. In one study, Drossel et al. (CITATION) found that the form of the functional response in the predator-prey equations has an effect on the final structure of the community, and that non-linear functional responses (especially a ratio-dependent one) led to more realistic food webs. Loeuille and Loreau (2005 CITATION), using another evolution-based model, showed that the size-structure often observed in natural communities is an emergent property of community development.

It is abundantly apparent that the use of models of multi-species dynamics has yielded great insight into why we observe certain patterns in natural communities. These models are a useful tool to find what Robert May called the "devious strategies which make for stability in enduring natural systems" (May 1973 p174 CITATION). To gain further insight we must integrate these dynamic-model-based approaches with the many tools that have been developed to assess the static structure (topology) of ecological communities.

While there is an abundance of packages available in the R programming environment to analyze the structure of networks, such as `igraph` (CITATION), `foodweb` (CITATION), `sna` (CITATION), `enar` (CITATION), `cheddar` (CITATION), there is a dearth of packages that can be used to apply dynamic models to the interacting populations. To my knowledge there is one other package that has been explicitly developed to apply a bioenergetic model to food webs, `gruyere` (CITATION), which has since become outdated and is no longer available for current R versions. With the numerous models for multispecies predator-prey dynamics it would be possible to simply utilize the numerical integration package `deSolve` (CITATION) to implement them, but it is rarely obvious how to go from the mathematics of the model to the required code.

In this paper I present an in-development R package for the simulation and analysis of ecological network dynamics. The package is still in-development because the current version limits the user to the simulation of food web (trophic) dynamics. Future versions will incorporate additional functionality (discussed below). My goal is to make it easier for researchers to apply models of multi-species dynamics as well as analyze the resulting changes in the community structure. Below I describe the model and variations currently implemented in this package, and describe how the simulation of multi-species predator-prey dynamics works. I also describe examples of the package usage, beginning with a simple two-species system to demonstrate how altering parameters affects the dynamics and ending with an example of a simulation that uses a niche model (CITATION) food web as a starting point. The niche model example additionally includes example code for the analysis of the simulation output to describe changes in the food web structure. I conclude by discussing the future development of the R package to extend simulation and analysis beyond trophic community dynamics.

2 Overview

In this section I present an overview of the core components of the package. There are three main aspects of the `rend` package: simulation, visualization, and analysis. In the simulation section I will describe the model used to simulate trophic dynamics as well as the modifications to the model structure and parameter changes the user can make. The visualization aspect of the package is focused on creating dynamic network visuals by generating an html movie of the simulation. This package also includes functions to aid in the analysis of community dynamics. Users are able to assess changes in nine commonly used food web metrics, change in the sub-structure of the food web (motif analysis), and changes in the trophic structure of the food web.

2.1 Food Web Dynamics Simulation

The current implementation of the `rend` package allows the user to take a food web, or any binary adjacency matrix, and simulate the trophic dynamics of the constituent species. The model is a bioenergetic model that was adapted from a version presented by Williams and Martinez (CITATION). The user may choose between two model structures: one that allows for a Holling functional response, or one that allows for consumer interference. The Holling functional response can be parameterized such that it may be either Type II or Type III to varying degrees (see below). There are also a number of other parameters the user may choose to alter, although they are currently set to default values suggested by Williams and Martinez (CITATION).

2.1.1 Consumer-Resource Model

A bioenergetic consumer-resource model is at the core of the food web dynamics simulator function. The model was originally developed for two species by Yodzis and Innes (CITATION), generalized to multiple species by McCann et al. (CITATION). The version included in this version of `rend` was first presented in Williams and Martinez (CITATION) and used again in Romanuk et al. (CITATION).

$$\frac{dB_i(t)}{dt} = G_i(B) - x_iB_i(t) + \sum_{j}^{n}(x_iy_{ij}F_{ij}(B)B_i(t) - x_jy_{ji}F_{ji}(B)B_j(t)/e_{ji}) \tag{8}$$

There are four basic parts to the model: growth, death, consumption of prey, and being consumed by predators. The first term, $G_i(B)$, is the function describing primary production of producer species in the absence of predation. Producers grow exponentially with density dependence.

$$G_i(B) = r_iB_i(t)(1 - \frac{B_i(t)}{K_i}) \tag{9}$$

In the equation, r_i is the intrinsic rate of increase, B_i is the biomass of population i, and K_i is the carrying capacity of population i. The consumption of resource j by consumer i is modeled by:

$$F_{Hij}(B) = \frac{B_j^{1+q}}{\sum_k B_k^{1+q} + B_0^{1+q}} \tag{10}$$

Here, B_j is the biomass of the consumed resource, in the denominator the summed biomass across all k resources, and B_0 is the half saturation density. The parameter q is a tuning parameter that lets the modeller shift the functional response between a type II ($q = 0$) and a type III ($q = 1$). The functional response above defines the fraction of a predator's maximal ingestion that is realized at a given time step (Figure 1).

In this model, the impact of the predator on the prey is equivalent to the impact of the prey on the predator. Thus the numerical response of the predator is the same magnitude as the functional response of the prey but of the opposite sign. This is accomplished as $F_j i = t(F_i j)$.

An alternative functional response is based on consumer interference (Beddington DeAngelis CITATION).

$$F_{BDij}(B) = \frac{B_j}{\sum_{k=1}^{n} \alpha_{ik} B_k(t) + (1 + c_{ij} B_i(t)) B_{0ji}} \tag{11}$$

Here the parameter c_{ij} gives the strength of the interference. Figure 1 shows how c_{ij} changes the shape of the functional response.

2.1.2 Implementation

The model is implemented in the code as a numerical integration using the deSolve R package (CITATION). Currently, the primary function is `CRsimulator`, which allows the user to specify all parameter values and the desired functions for the growth of basal species and the functional response. Inputs to `CRsimulator` are described in Table 1.

Table 1: Parameter definitions for the dynamic model

Parameter	Definition
Adj	Adjacency matrix
t	Sequence of time steps
G	Function for basal resource growth
method	Function to input into the ode solver
FuncRes	Functional response
K	Carrying capacity
x.i	Mass specific metabolic rate
yij	Maximum rate at which species i assimilates species j per unit metabolic rate of species i
eij	Conversion efficiency
xpar	Tuning parameter either q or c depending on the functional response
B.o	Half saturation density of species j when consumed by species i
ext	Function describing extinction events during the simulation
plot	Whether or not to generate a plot of biomass against time

The `CRsimulator` function can be broken down into four main parts: growth rate, parameter collecting, simulation, and visualization.

The first part, `grow <- getR(Adj)` creates a vector of whether or not species are basal. It uses the `getR`

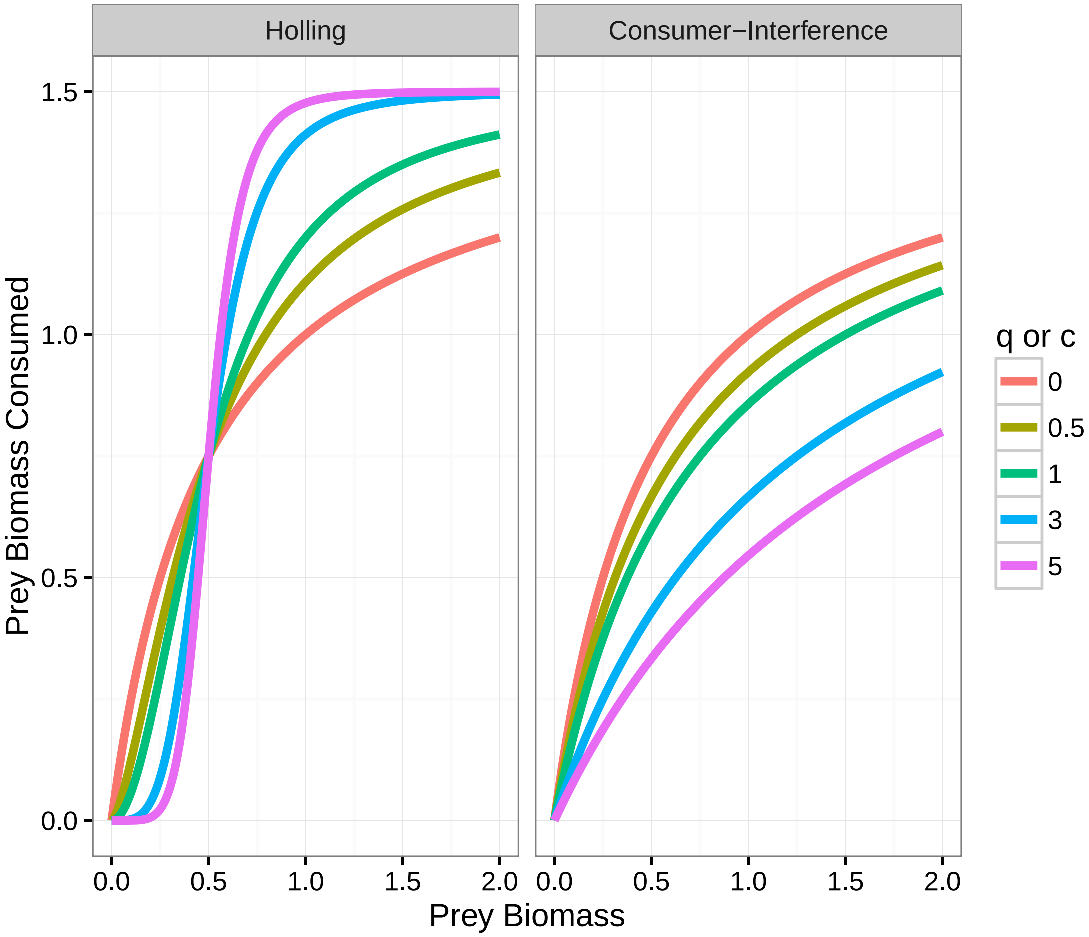

Figure 1: How the functional response of the prey changes with altered values of the tuning parameter

function to assess the column sums of the adjacency matrix to determine which species are basal. If there are no basal species, the function will return a warning (*"No basal species in simulation"*). The second part gathers all required parameters of the consumer resource model into a single list. The default values for these parameters are in Table 2. By default the function for growth is `Gi`, functional response is `Fij`, extinction events is `goExtinct`, and the default method is `CRmod`. All species initially start with a random biomass drawn from a uniform distribution between 0.5 and 1.

Table 2: Fixed parameter values for the dynamic model

Parameter	Value
K	1.0
x.i	0.5
yij	6.0
eij	1.0
xpar	0.2
B.o	0.5

The third part of the `CRsimulator` function is the numerical integration using `deSolve::ode`. This integration requires the method function `CRmod`, which codes the bioenergetic model. This function interacts with the event function, which by default is `goExtinct`, and determines whether a species has gone extinct at each time step by checking whether the species' abundance has passed below the threshold level for extinction ($10^{1}0$).

Currently `rend` only has two types of functional responses: one based on Holling's Type II and Type III (`Fij`), and a second based on consumer interference (`Fbd`).

Fij

Fbd

Both functions take the same input parameters: `B` is the vector of biomasses, `A` is the adjacency matrix, `B.0` is the half saturation constant, and `xpar` is either the q parameter of the Holling functional response or the consumer interference parameter c. `Fij` and `Fbd` also both return a matrix reflecting the impact of species i on species j.

The last part of `CRsimulator` will plot the output of the integration (when `plot = TRUE`).

2.2 Food Web Dynamics Visualization

In order to create a visualization of the dynamics of the food web through time, the rend package relies on the animation package. With the current version of rend, the user is able to take the output of the `CRsimulator` function and generate a video representation of the dynamics of the time series. Both biomass and interaction dynamics are visualized. Visualization is done by the `netHTML` function, which serves as a wrapper for the `animation` package's `saveHTML`. The function output is an HTML video of the food web at each time step.

2.3 Analysis

In addition to the functions for simulation, the `rend` package also has several functions to analyze the output of the simulations. The three main analyses that can be run are (1) changes in food web indices through time, (2) change in motif structure through time, and (3) change in trophic position for each species through time. The function `WEBind` takes the output of `CRsimulator` and the initial adjacency matrix that was fed into the simulation as input. From this, new adjacency matrices are created for each time step and a eight indices are computed for each matrix. The output is the number of species, number of links, link density, connectance, diameter, average path length, clustering coefficient, and modularity (from the `rnetcarto` package, CITATION). These terms are defined in Table 3. Change in three-species motif structure through time can be found using the function motifCounter3. Just like with WEBind the inputs for the motif counting function are the output of the simulation and the initial adjacency matrix. The function then converts the adjacency matrices for each time step into igraph (CITATION) graph objects, which are then fed into the igraph function triad.census to get the frequency of three node configurations for each time step. The third analysis function, trophicChange, takes the same inputs: the output of the simulation and initial web. Then trophicChange outputs a species x time matrix where each row is the trophic position for each species at a given time step (and 0 represents extinction). Trophic position is found using the TrophInd function from the NetIndices R package (CITATION).

Table 3: Definitions of the common food web indices given from WEBind

Index	Definition
N	The number of species wih positive biomass
Ltot	The total number of links among species with positive biomass
LD	The average number of links per species
C	The number of realized links as a fraction of the number of possible links
D	The longest shortest path between two species
APL	The average number of links between two species
CC	The probability of intraguild predation
M	The degree to which species are more connected to other species in their compartment than in others
nMod	The number of modules in the food web

3 Examples

In this section I outline two examples of the rend package in action. The first example is a two-species predator-prey system. I use this simple example to highlight the differences in the dynamics given by the two model structures, Holling and consumer interference. The second example is a more realistic use of the package, where the dynamics of a model food web are simulated and analyzed. I chose to use a well-studied model of food web topology, the niche model (CITATION). This example is primarily used to highlight the analytical functions used to assess changes to topology, motifs, and trophic positions.

3.1 Two-Species Dynamics (Multiple Functional Response Types)

A simple example of this package is with two species, a basal resource and a consumer. The adjacency matrix is two rows and two columns.

```
     [,1] [,2]
[1,]    0    1
[2,]    0    0
```

All that is required is to feed this adjacency matrix into the CRsimulator function, and choose whether to alter any of the default parameter settings. The first six rows of the output are shown in Table 3. Column one is the time step, column two is the biomass species 1 (the basal species), and column three is the biomass of species 2 (the consumer). The resulting biomass dynamics are shown in Figure 2 below. The two species reach equilibrium rather quickly, within 100 time steps, following damped oscillations.

Table 4: First six rows of the output of the CRsimulator function

time	1	2
1	0.8872	0.8984
2	0.0186	1.1540
3	0.0030	0.7135
4	0.0016	0.4346
5	0.0014	0.2643
6	0.0015	0.1607

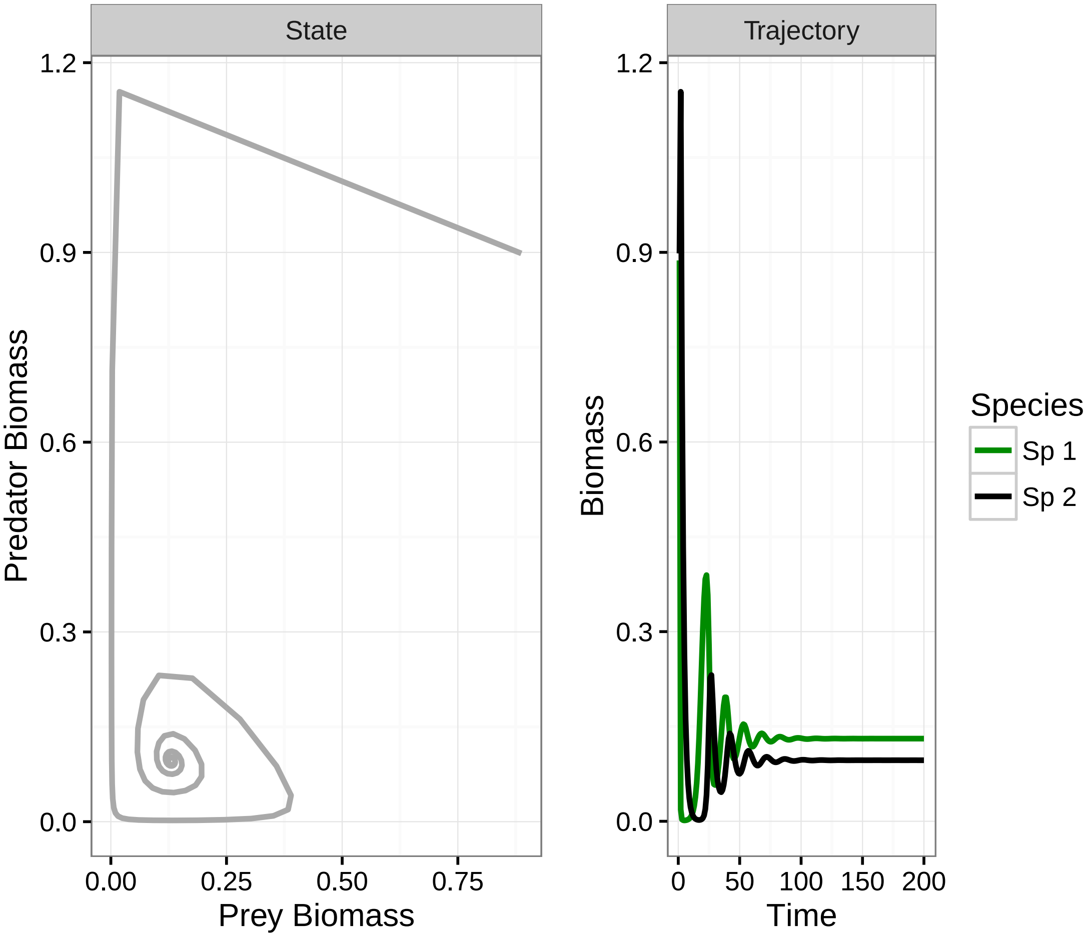

Figure 2: The dynamics of consumer and resource given default settings in CRsimulator

The previous example in Figure 2 simply used the default settings for the `CRsimulator` function. An important setting for the user to select is the form of the functional response. Depending on whether the functional response is Lotka-Volterra or based on consumer intereference, you would expect to observe differences in the dynamics of the two species. The result of altering the functional response, and preserving the other default settings is shown in Figure 3. Looking at the difference between Lotka-Volterra (`Fij`) and consumer interference (`Fbd`) it appears as if the dynamics of the two species with Lotka-Volterra dynamics equilibrate faster. The pair of species with consumer interference are still exhibiting damped oscillations at time step 200, while the other pair are no longer oscillating.

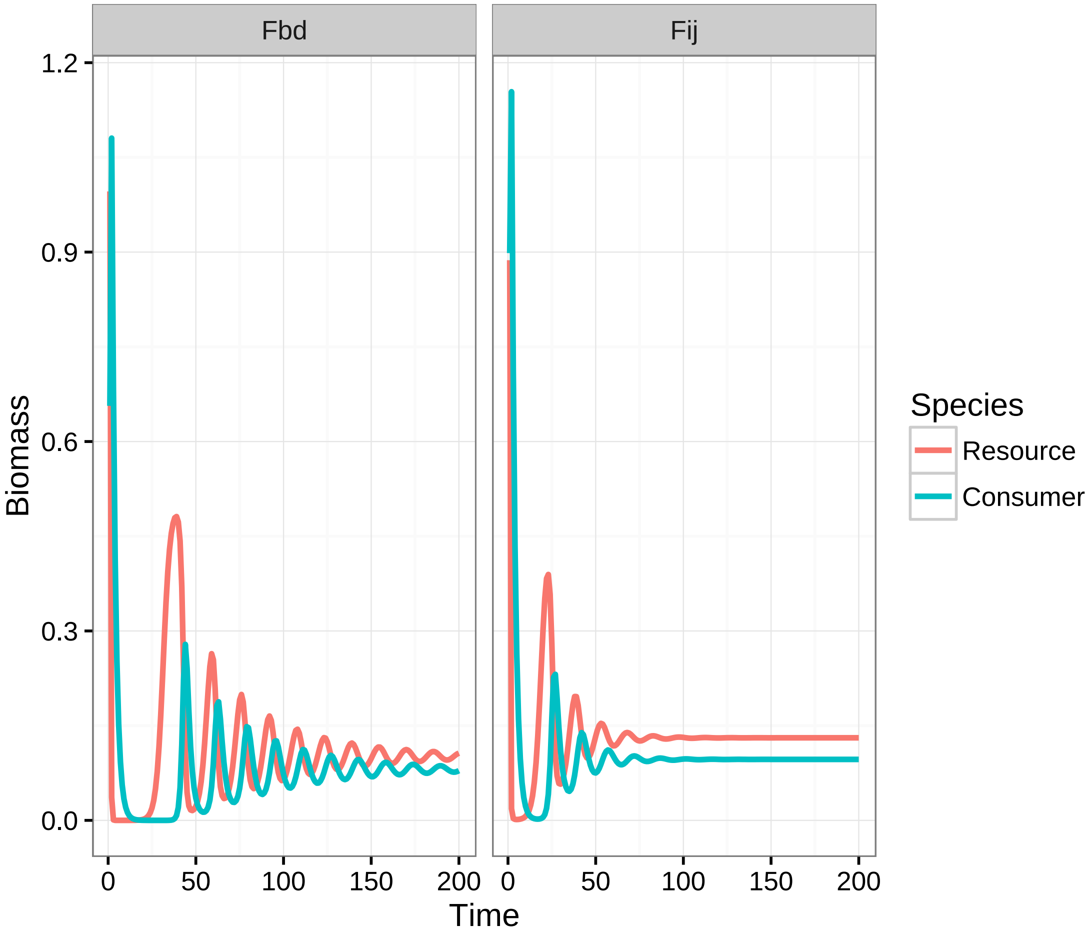

Figure 3: The dynamics of consumer and resource with differing functional responses given default settings in CRsimulator

One would also expect that differences in the tuning parameter (q in the Holling functional response or c for consumer interference) will also alter the dynamics of the interacting species. By systematically altering the tuning parameter (`xpar` in `CRsimulator`) I can show how the dynamics associated with each functional response type change. The results of the two species `CRsimulator` with default settings except for `xpar` equal to 0, 0.2, 1, and 5 for each functional response type are shown in Figure 4.

Williams and Martinez (CITATION) note that for the Holling functional response q of 0 corresponds to a Type II functional response and q of 1 corresponds to a Type III functional response. When q is greater than one, the size of the prey "refugia" is larger as demonstrated by the larger lag in the functional response shown in Figure 1. For a consumer interference functional response (`Fbd`) the parameter c reflects the strength of the interference among consumers.

As the parameter q increases the two species reach equilibrium faster and the equilibrium biomass of the prey (Species A) is higher. With increasing consumer interference the two species also reach equilibrium faster, but there is a smaller difference in equilibrium biomasses. At the highest level of consumer interference (c = 5), however, the equilibrium abundances of both species are higher than at lower levels of interference.

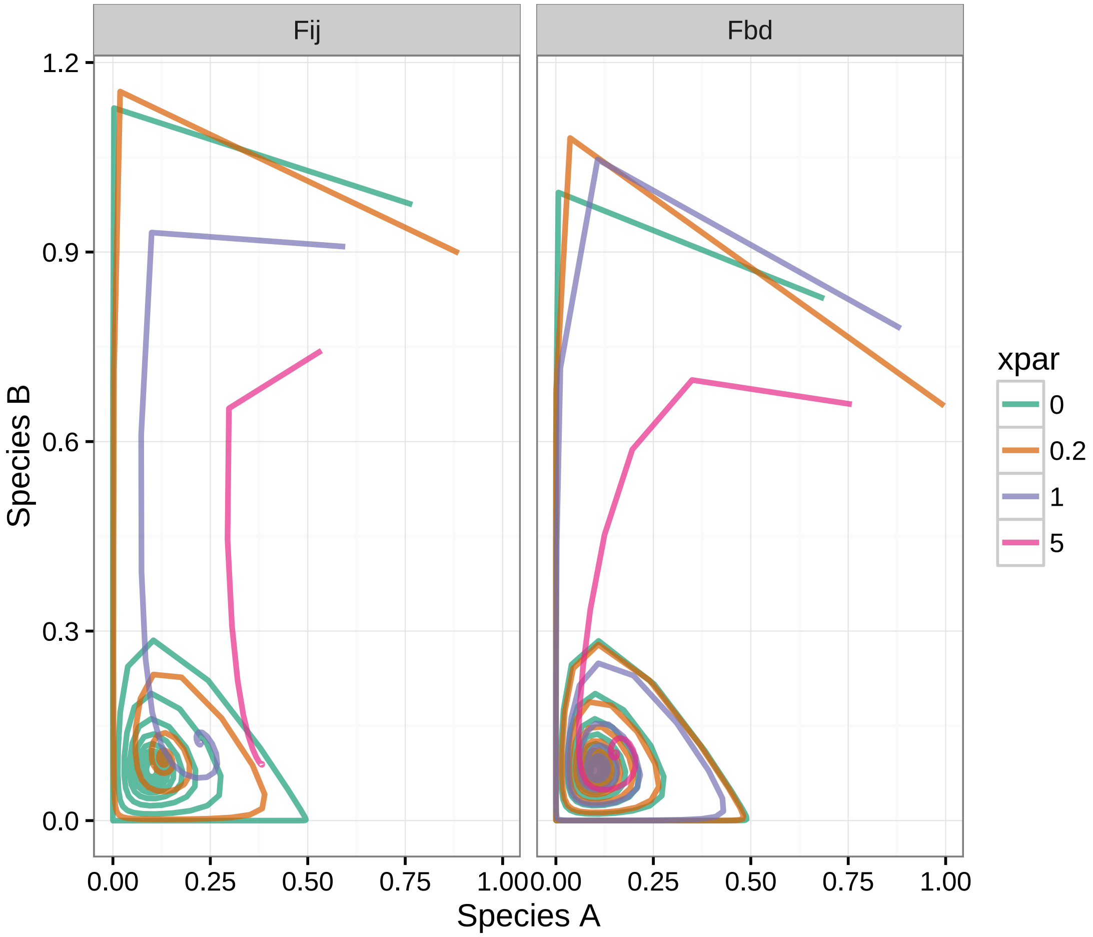

Figure 4: The dynamics of consumer and resource given varying xpar settings in CRsimulator

3.2 Sample Run of Dynamics on Model Food Webs with Analysis

Most models of food webs generate a binary adjacency matrix (Figure 5) where a one indicates that species i is consumed by species j. This allows for convenient pairing with the `rend` package. The `CRsimulator` function can take the output of these models as input and simulate the dynamics of the participating species. In this example I demonstrate how the package can be used to simulate the dynamics of a niche model food web. The niche model is a food web model that arranges S species along some hypothetical niche axis. Each species consumes other species based on a randomly sampled feeding range with mean c_i sampled from a uniform distribution from 0 to the species' niche value. The range is dependent on the user specified connectance C.

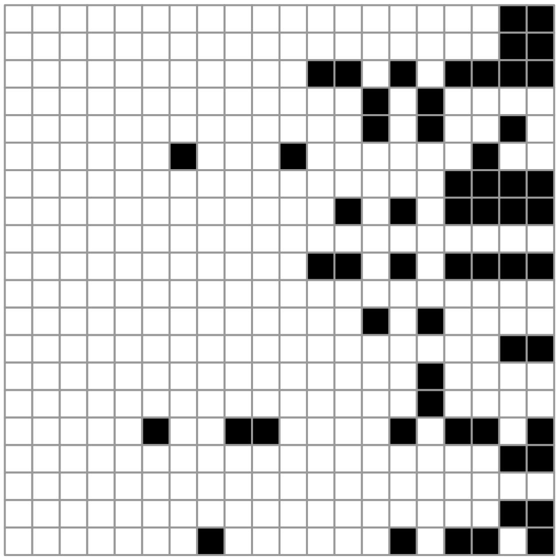

Figure 5: Adjacency matrix of a niche model food web with 20 species and connectance of 0.15

Figure 6 shows the output that would be given if `plot = TRUE` in the `CRsimulator` function, with an additional inset showing the last 25 time steps for clarity. It is clear that following a short period of high oscillatory behavior the system reaches equilibrium. Following the dynamics of predator prey interactions in this example, of the initial 20 species in the food web 8 have gone extinct (Figure 7, N). All measured food web indices exhibit changes through time during the simulation (Figure 7), which indicates that the extinctions caused by population dynamics in the model have an effect on the structure of the food web. The most obvious change in structure resulting from species extinction is a reduction in the number of species (N), and therefore the number of total links in the food web ($Ltot$). Concomitantly, the average number of links per species is reduced overall, however there is a bump around time steps 50, suggesting that the extinctions in that time were species with relatively few links. The initial sharp decline in LD was likely caused by the loss of species with many links. Connectance (C) increases through time which is likely the result of the greater influence of the number of species on the measure compared to the effect of losing interactions. Decreases in food web diameter (D) and average path length (APL) are most likely to be caused by the loss of higher trophic level species (see also Figure 8). The clustering coefficient (CC) and modularity (C) show opposite patterns, with CC increasing and M decreasing. There is no change in the number of modules ($nMod$) and with the decrease in modularity the three modules get more strongly connected to one another.

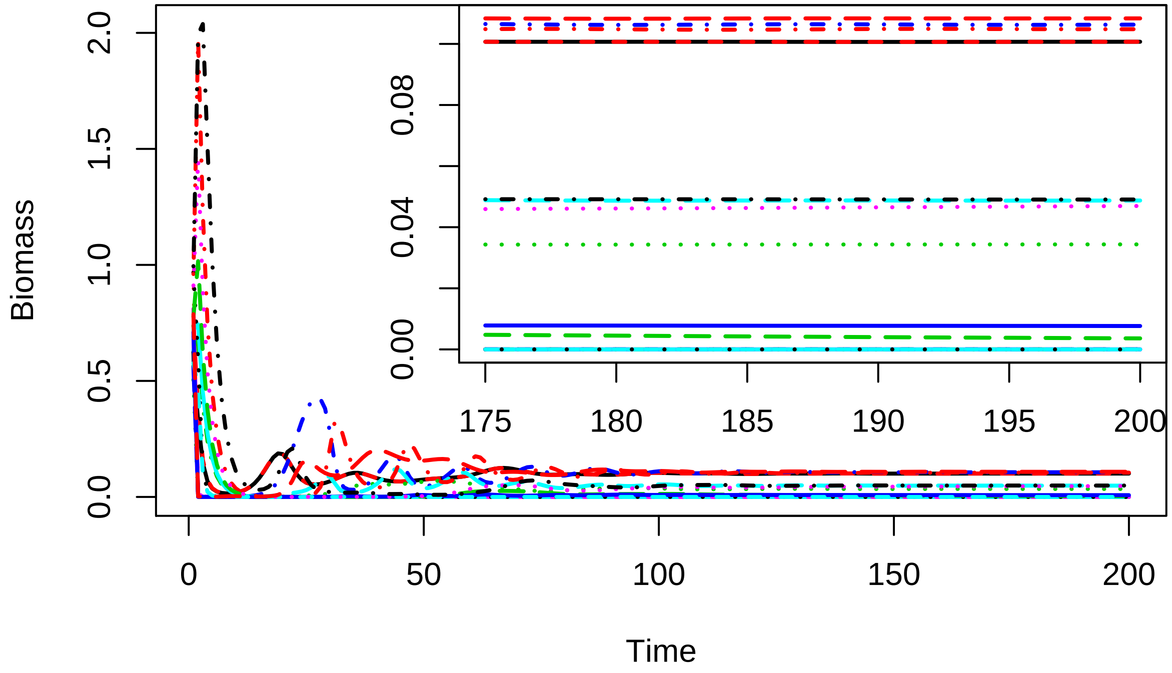

Figure 6: Biomass dynamics of species in a niche model food web, inset shows dynamics of last 25 time steps

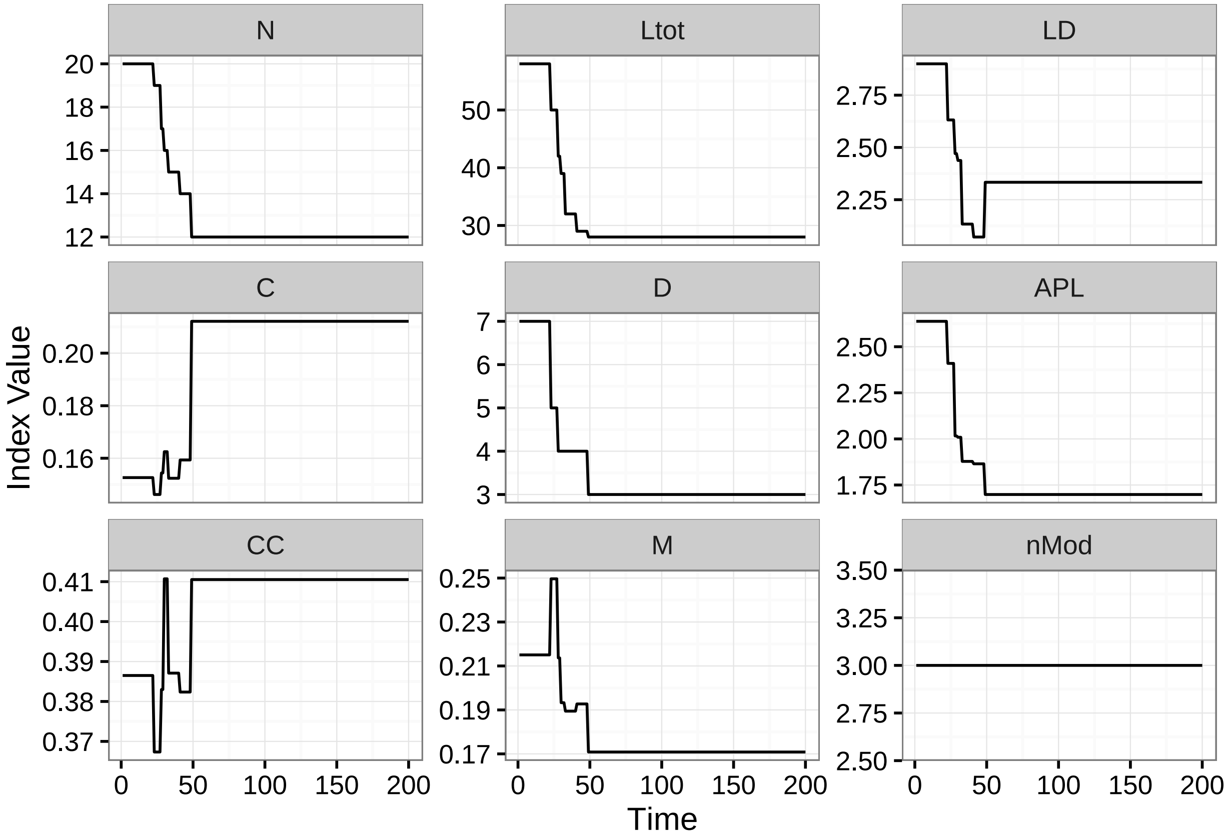

Figure 7: Change in food web indices over time

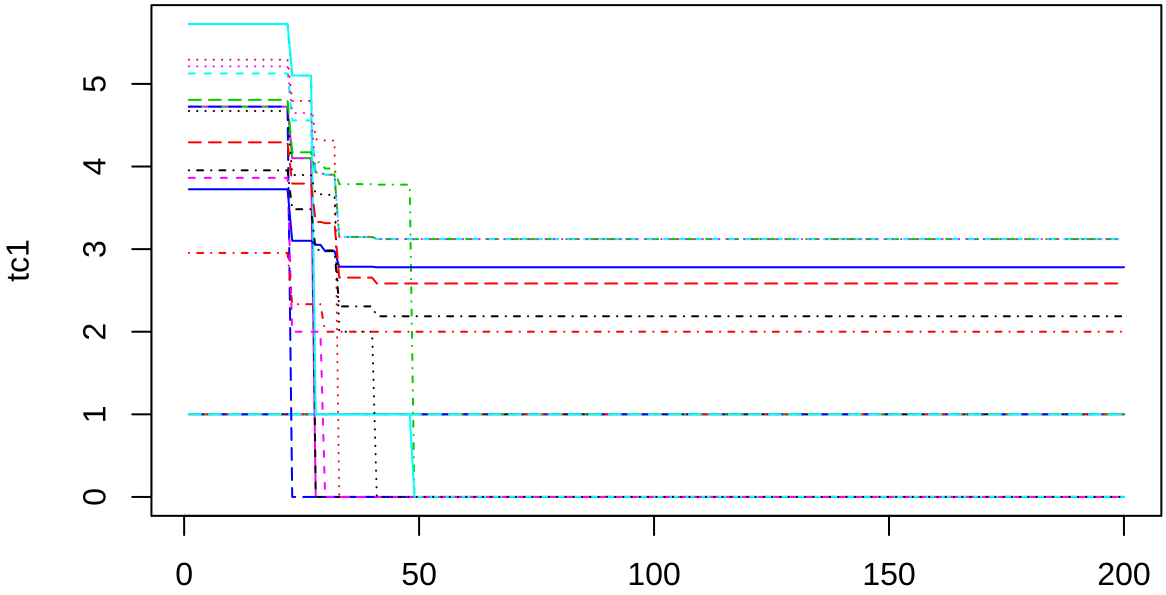

Figure 8: Change in trophic position over time

3.3 Food Web Dynamics Explorer via Shiny Web App

4 Conclusions

5 Future Directions

I can see a number of different directions this package could take in terms of increasing performance and functionality. Below I highlight several lines I plan to take in further developing the `rend` package.

5.1 Output assessment

The current functions for assessing the output of the simulation are descriptive in nature. They simply show the change in some index through simulated time. An important addition to this suite of functions will be ways to test hypotheses throught the implementation of null models. One null model would be to assume species extinctions were random. By comparing the network structure following random extinction orders to the food web following the simulation, one could determine if certain structures were more or less likely to be stable.

5.2 Increasing options for trophic dynamic models

Currently this package only supports a narrow range of model structures. Users can simulate dynamics with a Holling Type II, II.2 (CITATION) or III function response, or they can use the Beddington-DeAngelis functional response with consumer interferences. This is only a small subset of the proposed functional response types and ignores both simpler and more complex forms.

Drossel et al. (CITATION) demonstrated that ratio-dependent functional responses led to more realistic food web structures. Therefore, in order to more properly assess how population dynamics impacts the structure of food webs using this package, I should incorporate this type of functional response. The problem here is that there is not currently an implementation of a ratio-dependent functional response for bioenergetic models. There has, however, been work to that effect in population-abundance based models.

Additional options for simulation of trophic dynamics would be to incorporate abundance-based models in addition to the current bioenergetic model. This would allow for an expanded selection of functional forms and more flexibility for the user.

5.3 Adding models for alternative interaction types

The package `rend` was initially imagined as a package for ecological network dynamics, which extends beyond food webs and predator-prey interactions. The umbrella of ecological networks includes other interaction types, and mutualistic or competitive networks are becoming increasingly popular areas of research. In addition to providing support for simulations of alternative interaction type networks, I would like to allow for dynamic simulations of whole communities. Natural communities are never made up of species interacting only one way, and there should be a way to integrate models of different interactions into a single simulation.

5.4 Incorporate Stochasticity

Currently the models I have implemented in the `rend` package are deterministic. Multiple runs with the same food web adjacency matrix will result in the same equilibrium community, and the only source of randomness in the model is the initial biomasses of species, which are selected from a random uniform distribution. Any differences in initial biomass, however, are rarely large enough to alter the final community structure.

The reason that these models must be deterministic is because the `deSolve` package, which allows for the numerical integration, does not support stochastic differential equations. The alternative is to utilize the `sde` package (CITATION), which is a package for the numerical integration of stochastic differential equations.

Incorporating stochasticity in the simulation would allow for risk-based assessment of community configurations. Analysis of the dynamics resulting from stochastic models could proceed probabilistically rather than as binary outcomes (either a species goes extinct or it does not). Furthermore, as species in nature typically display stochastic dynamics, any predictons made from stochastic models would likely hold up better.