# Selection on stability in food webs

A Dissertation Presented

by

**Jonathan Joseph Borrelli**

to

The Graduate School

in Partial Fulfillment of the

Requirements

for the Degree of

**Doctor of Philosophy**

in

**Ecology and Evolution**

Stony Brook University

**August 2016**

**Stony Brook University**

The Graduate School

**Jonathan Joseph Borrelli**

We, the dissertation committee for the above candidate for the

Doctor of Philosophy degree, hereby recommend

acceptance of this dissertation.

**Lev Ginzburg – Dissertation Advisor**
**Professor, Emeritus, Department of Ecology and Evolution**

**Dianna Padilla - Chairperson of Defense**
**Professor, Department of Ecology and Evolution**

**Stephen Baines**
**Associate Professor, Department of Ecology and Evolution**

**Stefano Allesina**
**Professor, Department of Ecology and Evolution, University of Chicago**

This dissertation is accepted by the Graduate School

Nancy Goroff
Dean of the Graduate School

Abstract of the Dissertation

# Selection on the stability of food web building blocks

by

**Jonathan Joseph Borrelli**

**Doctor of Philosophy**

in

**Ecology and Evolution**

Stony Brook University

**2016**

Whereas much of ecological research has been focused on enumerating the differences among ecological communities, it is equally important to understand their similarities. Food webs exhibit remarkably conserved patterns regardless of the type of environment in which they are found. Evolutionary and ecological processes are able to generate a variety of community configurations, but some of these will have a higher chance of being stable. Selection on these configurations based on their stability properties should lead towards an increased representation of stable structures. Moreover this selective process will operate the same across different environmental contexts. The goal of this dissertation is to examine whether commonly observed food web properties may be the result of selection on stability. I have focused on two properties in particular: (1) that food chains tend to be short, and (2) three-species food web motifs. For both of these properties I detected the signature of a selection on stability process. Food webs with short food chains are more likely to be stable. Likewise, the three-species modules that are most likely to be stable (tritriphic chains, apparent and exploitative competition) are also those that occur more frequently than expected by chance in empirical food webs. To better understand how selection on stability may interact with pattern generating mechanisms I used simulated species removals and introductions coupled with a bioenergetic model of species dynamics. Species removals and introductions that alter food web properties towards unstable structures should result in decreased persistence and invasion success respectively. I found some support for this hypothesis, although the manner by which dynamics were modeled had a substantial impact on the results. Nonetheless the framework of selection on stability may be a simple yet powerful tool to aid ecologists in generating an expectation of what should be observed in natural communities.

## Dedication Page

To my parents Betta Vice and Joseph Borrelli, and to my Grandparents John and Elizabeth Vice.

# Table of Contents

## List of Figures

**Chapter 2:**

**Chapter 3:**

**Chapter 4:**

**Chapter 5:**

**Appendix 1:**

**Appendix 2:**

# List of Tables

Chapter 4:

Chapter 5

## Acknowledgments

I would like to first acknowledge my advisor, Lev Ginzburg, whose enthusiasm and encouragement greatly shaped my thoughts and work while writing this dissertation. I would also like to thank my committee; Dianna Padilla and Stephen Baines constantly pushed me to avoid falling into the theoretical rabbit-hole and to always bring my theory back towards the empirical side of ecology and Stefano Allesina for hosting me in his lab and helping me to develop my ideas further.

I must also thank my fellow students; Mica McCarty-Glenn, Ben Weinstein, Shea Lambert, Charlie DiRienzo, Zach Chejanovski, Mike McCann, and Fab Spagnolo. They always provided interesting discussion and distraction, often when sorely needed.

I am most thankful to Mary Alldred who has been constantly supportive, pushed me to do better, and helped me enjoy not only doing science but life as well.

Figures 2.1, 2.2, 2.3, 2.4, and 2.5 have been reproduced from their original publication in Food Webs with permission from Elsevier.

Figures 3.1, 3.2, and 3.3 have been reproducd from their original publication in Oikos with permission from John Wiley and Sons.

# Chapter 1: Introduction

“In nature we deal not with arbitrary complex systems, but rather with ones selected by a long and intricate process” (May 1973a p. 3-4). An early hypothesis among ecologists was that ecological communities must be complex, with many pathways for energy transfer, in order for them to be stable. Observations supported this hypothesis, as most natural communities appear to be both complex and stable. That complexity begets stability dominating ecological thinking for over a decade until May (May 1972, 1973a) demonstrated that, mathematically, there is no reason to expect increased complexity in the form of more species, more links among them, or stronger interactions, should lead to increased stability. Rather, the opposite appeared to be true, and in random communities (composed of multiple interaction types) increased complexity decreased stability. Although, later studies showed that when these random systems were constrained such that all species had positive equilibrium densities, stability increased with complexity (Roberts 1974, Christianou and Kokkoris 2008), May’s (1972) paper sparked over four decades of research on ecological networks.

If complex communities are observed in nature despite reduced stability, then logic dictates that there must be some property of these systems that allows them to persist with all species coexisting. To quote May (1973a, p174), “[t]he task, therefore, is to elucidate the devious strategies which make for stability in enduring natural systems.” While more recently, research interest has expanded to networks of plants and their pollinators (Olesen et al. 2008, Bascompte 2009) or frugivores (Plein et al. 2013), parasites and their hosts (Lafferty et al. 2006, Poulin 2010, Dunne et al. 2013), and other types of ecological interactions (Ings et al. 2009, Kéfi et al. 2012), by far the bulk of research has focused on food webs (Dunne et al. 2002, Vermaat et al. 2009).

**What is a food web?**

Food webs are network representations of the trophic relationships among organisms in a community. The nodes of the network are the species and the links indicate that species *i* is preyed upon by species *j*. There are three main types of food webs; connectedness, energetic, and functional (Paine 1980). Connectedness webs show all trophic interactions among species. Energetic and functional webs include measures of the strength of the interactions, and functional webs only show the strongest interactions. The links in these webs may be either binary (connectedness) or weighted (energetic and functional). Most food web datasets include only binary descriptions of links, and those that have weights associated with them may indicate any number of potential measures of interaction strengths (Berlow et al. 2004). Many are familiar with the graphical representation of networks, but these can also be represented in the form of a matrix, where in binary data a one indicates that the row species is consumed by the column species. Both of these representations allow us to apply mathematical methods to assess the structure of the web.

Early food web datasets were aggregated from a variety of literature sources and compiled into a collection of 113 webs (Cohen et al. 1986). They vary in size from as few as 5 species to as many as 50. Further development of food web theory, and application to these data demonstrated that there were serious methodological issues associated with their collection. Foremost was the unequal levels of aggregation such that the upper trophic levels were relatively well resolved, including single species, yet the lower levels were lumped, with groups of many species considered together. Later food web data were collected to specifically address the methodological issues associated with the earlier data. The first well-resolved food webs were collected from Little Rock Lake, Wisconsin (Martinez 1991) and Coachella Valley Desert, California (Polis 1991). Many other data sets have since been collected and a core set of 18 webs

(Dunne et al. 2002) has been the focus of a great deal of theoretical development. There are now just over 300 food web datasets currently available in the GlobalWeb Database (globalwebdb.com).

**Food web patterns**

Along with better and more realistic models, new and improved data allowed for better pattern detection in food webs. For example, Neutel et al. (2002) found that there is a pattern of weaker interaction strengths in long trophic loops. Food webs composed of mostly weak and few strong interactions tend to be more stable (May 1972, McCann et al. 1998). A number of food web structural patterns have also been investigated in order to determine the "deviant strategies" by which real communities can be both complex and stable. Two important structural patterns are modularity and nestedness (Allesina and Pascual 2008, Fortuna et al. 2010, Thébault and Fontaine 2010).

Modularity, or compartmentalization, is the tendency for groups of species to be more connected within the group than between groups (Krause et al. 2003, Kondoh 2008). In a recent paper Rezende et al. (2009) showed that in a marine food web, compartments are arranged around different shark species and based on phylogeny, body size, and the structure of the habitat. Nestedness, on the other hand, is a pattern whereby the more specialized consumers utilize a subset of those resources consumed by more generalist species (Bascompte et al. 2003, Bastolla et al. 2009, Thébault and Fontaine 2010).

There is some debate as to whether food webs exhibit patterns of nestedness, with authors arguing both for (Bascompte et al. 2003, Bascompte 2010, Fortuna et al. 2010) and against (Allesina and Pascual 2008, Thébault and Fontaine 2010) this notion, while others purport that food webs consist of nested modules (Kondoh et al. 2010). It is generally thought that these two structural patterns allow food webs and other networks to be more stable (but see Allesina and Pascual 2008), and allow for greater diversity and complexity (Thébault and Fontaine 2010). Modularity reduces the impact of perturbations throughout the web by isolating them within a compartment (Krause et al. 2003, Stouffer and Bascompte 2011). The nested structure of networks reduces competition, allowing greater coexistence (Bascompte et al. 2003, Bastolla et al. 2009).

There are also apparent patterns in the way components of the food web are put together. Food webs are composed of a number of different food chains, and energy flows from basal species to top predators. Such food chains vary in their length, the number of energy transfers, and determine the trophic position of species, defined as one plus the average trophic position of their prey (basal species are trophic position one). While food chain length is variable there are fairly clear limits to the number of levels in a community. Consumer species typically have a trophic position between three and five, and very rarely more than six.

Another important structural pattern is the degree of representation of various subgraphs, or motifs (Milo et al. 2002, Stouffer et al. 2005, Camacho et al. 2007). A motif in this case is a subgraph that is overrepresented in a network (Milo et al. 2002, Bascompte and Melián 2005, Stouffer et al. 2007). Food webs in particular tend to have more tritrophic chains and biparallel chains than expected at random (Milo et al. 2002). Thus these structural motifs represent the building blocks of food webs (Bascompte and Melián 2005), and, as such, are likely to be dynamically important with respect to the stability of the network (Prill et al. 2005, Rip et al. 2010). For computational reasons, only 3- and 4-node subgraphs have been investigated (Milo et al. 2002, Stouffer et al. 2007). There are 13 possible 3-node subgraphs and 199 for 4-nodes, for example (Stouffer et al. 2007). For 3-nodes it has been shown that the same subgraphs tend to be overrepresented in real food webs regardless of habitat type, and that model food webs exhibit similar patterns (Stouffer et al. 2007, Stouffer 2010). Stouffer et al. (2007) used these results to validate the prey selection mechanism of the niche model (Williams and Martinez 2000), where species are assigned a value along an arbitrary axis and consume prey within a defined range whose center is below their own value.

## Food web stability

Stability has been a central theme in the study of ecological networks since May (1972) sparked an interest in understanding what makes for stability in enduring natural systems. Not only have food webs been examined for particular common patterns, but those patterns are investigated for how they relate to the stability of the system.

### *What is stability?*

Unfortunately, as with many aspects of ecology, there is a lack of a clear single definition of the term *stability*. Moreover, many definitions do not readily translate from the theoretical realm into the empirical and vice versa. Stability may be measured as variability in time and space, species turnover, number of extinctions, number of invasions, resistance, resilience, and local stability (Pimm 1984, Donohue et al. 2013). Variability refers to the flux in species biomasses, either in time or space, and a stable community would be less variable. A more stable community should also be relatively unaffected by species extinctions (i.e., if one species goes extinct others do not as well; Dunne and Williams 2009, Lafferty and Kuris 2009), they should also be resistant to invasions (Fridley et al. 2007, Ives and Carpenter 2007). From both of these definitions we can conclude that stable communities should have low species turnover through time. Resistance, resilience, and local stability are all measures of stability that rely on equilibrium. A resistant community will tend to remain at equilibrium, a resilient one will return from perturbation faster. Local stability simply indicates that given a small perturbation, the community will return to equilibrium.

By far the most commonly used measure of stability, in theoretical studies at least, is that of local stability (May 1972, Pimm and Lawton 1977, Allesina and Pascual 2008, Allesina and Tang 2012). Local stability is measured as the sign of the eigenvalue with the largest real part of the Jacobian matrix, the matrix of effects each species has on the others' growth. The downside to this measure of stability is that it requires both the assumption of some form of the equations governing species growth and the ability to properly parameterize those equations. Assuming a model structure is not straightforward, as there is some debate as to which of the many proposed model structures most closely resembles reality (Abrams and Ginzburg 2000). Regardless of the chosen form, for any reasonably realistic number of species there are exponentially many parameters that would require estimation, a feat that is not simple given the data currently available. Some have gotten around this last issue by using the allometry between body size and metabolism to estimate interaction strengths (Brose et al. 2006).

An alternative to estimating local stability is the use of a qualitative measure, one that does not require estimation of parameter values. Qualitative, or sign-, stability allows one to determine whether or not a system will be stable based solely on the sign pattern of the Jacobian matrix (e.g., +/- for predator-prey systems) rather than their particular magnitude (May 1973b). Given a sign-stable structure, the system will return to equilibrium following a small perturbation regardless of the magnitude of the species' effects on one another. Despite this relatively appealing measure of stability, in order for a structure to be sign-stable it must fulfill specific conditions that most empirical food webs do not. While food webs in nature are not sign-stable, it is possible to measure the degree to which they are sign-stable. By estimating the distribution of relative impacts of species on the others' growth random Jacobian matrices can be simulated, and the proportion of randomly sampled matrices that are locally stable is the degree to which it is sign-stable. This measure is called quasi sign-stability (Allesina and Pascual 2008) and is a very useful measure of how a system will respond to environmental variation.

### Selection on stability

Typically ecologists are interested in the describing the differences among systems, yet, it is equally important to understand how similarities develop. The quote of May (1973a) at the start of this introduction illustrates a wider view that is held by a number of ecologists, that there is some selective process that leads to the preferential observation of stable configurations of species. The idea that selection on stability can generate observed patterns in nature requires three parts. First, there must be a mechanism that leads to the generation of patterns. Of these patterns some will be more likely to be stable than others, and there will thus be some selective process (i.e., the preferential loss of unstable configurations). Finally, the selective process leads to the observed patterns.

This formalized framework for selection on stability can readily be applied to the observations made on food web structure. Patterns in food web structure are generated via the introduction of species through invasion and speciation, their loss via extinction and migration, and/or the rewiring of links as result of foraging decisions and population dynamics. Of the many potential food web configurations, some will allow for the stable coexistence of all species, others will not. If not, then species loss should result in a configuration that has a higher probability of being stable. Those configurations that are most likely to be stable could then be considered to have been selected from the pool of potential configurations, and should then be those that are most often observed in natural systems.

**What follows…**

Chapters 2 and 3 of this dissertation are focused on two particular food web patterns, food chain length and three-species subgraphs. In both chapters I investigated the role that selection on stability may play in generating the observed patterns in nature. I determined the signature that would be expected from the process of selection on stability by comparing the stability of these food web components to their observed frequency. While the expected signature was found, this does not constitute substantial evidence that the patterns of short food chains and three-species motifs develop as a result of selection on stability. Evidence of this process could, however, come from models of multi-species dynamics. Chapter 4 describes an R package that I developed to simulate the dynamics of multi-species predator prey systems. This package offers an easy to use method of simulating food web dynamics as well as functions to analyze the output of the simulations. Finally, in Chapter 5 I used this package to simulate the dynamics of disassembling and assembling food webs.

# Chapter 2: Why there are so few trophic levels: selection against instability explains the pattern

## Abstract

Food chains are short, rarely more than five trophic levels long. But, the cause of this pattern remains unresolved, and no current hypothesis fully explains this phenomenon. I offer an explanation based on the stability of food chains that have been shifted away from linearity to be more web-like. I start with a simple example of food webs of two to six species arranged so that species consume all those with a trophic level less than their own. The probability of stability, for such "universal omnivory" chains declined strongly with chain length, and was as low as 1% with six level chains but highest for two and three level chains. I further explored the influence of chain length on food web stability by testing food webs with varying levels of connectance that were constructed either randomly or with the niche model. By additionally altering the relative impacts of predators on prey, and vice-versa, I test the role of my assumptions on the relationship between chain length and stability. Food webs characterized by low to moderate degrees of connectance, asymmetrical interactions, and relatively weak density dependence showed a pattern of reduced stability with longer trophic chains. The simple view that food webs characterized by long trophic chains are less stable seems to resolve the long-standing question of why there are so few trophic levels in nature.

## Introduction

Food chains are typically short, often having as few as two steps (Elton 1927, Pimm and Lawton 1977) and rarely more than four (Yodzis 1981, Pimm 1982). Food chain length is directly related to the number of trophic levels in a food web. A food chain with two steps has three trophic levels; a producer, intermediate consumer, and a top predator. By defining trophic level this way, however, it becomes difficult to determine the trophic levels of species embedded in complex food webs. To better define the trophic level of a given species in a food web, I will use two related, but distinct, definitions: trophic position and longest chain.

The distribution of trophic position, measured as one plus the average trophic position of a species' prey, for 50 published food webs shows that 98.8% of consumer nodes have a trophic position less than or equal to four (Figure 2.1a). Very few species have a trophic position higher than five in these food webs (see Appendix 1 for details on the webs used). Alternatively, the maximum number of steps between a given consumer and a basal species, the longest chain, in 39 of the 50 published webs (78%) is less than or equal to five levels (Figure 2.1b).

A recent study by Ulanowicz et al. (2013) demonstrated that by accounting for the amount of biomass flowing along the links (links with more biomass flow are weighted more heavily) the number of effective trophic levels is approximately three for a set of 16 networks. Ulanowicz *et al.* (2013) speculated that this pattern may result from the elimination of configurations of interacting species that are less likely to persist than others.

The most commonly tested hypotheses for variation in food chain length are associated with relatively conflicting support. The earliest explanation for food chain shortness is that the efficiency of energy transfer between trophic levels is low. Available energy at a trophic level should therefore decrease rapidly going up the chain (Elton 1927, Lindeman 1942, Hutchinson

1959). Areas with higher energy availability (often measured as productivity) should then support longer chains, a prediction not supported by empirical observations. Up to five trophic levels are observed in both the highly productive tropics and the low productivity polar regions (Pimm 1982). An alternative explanation is that larger ecosystems allow for longer food chains (Post et al. 2000). Ecosystem size has been found to be related to food chain length in lakes and some islands (Post 2002, Takimoto et al. 2008) but not on other islands (Young et al. 2013). The combination of productivity and ecosystem size, the productive-space hypothesis, has also been proposed to explain variation in food chain length (Schoener 1989, Spencer and Warren 1996, Vander Zanden et al. 1999, Post et al. 2000). Evidence for the productive-space hypothesis, however, is contradictory with an equal number of studies finding support as those failing to find support (Post 2002, 2007, Young et al. 2013).

Longer food chains are also expected to be dynamically fragile. Pimm and Lawton (1977) explored the role of dynamic constraints in limiting the length of food chains. Dynamic constraints would occur if the number of trophic levels in a community was limited by population dynamics of the constituent species. Using analyses of community matrices of four species food webs, they found that as the number of trophic levels increased, so too did the time it took to return to equilibrium following a small perturbation. A decrease in stability with increasing chain length is also supported by evidence from protist microcosm studies (Morin and Lawler 1996, Holyoak and Sachdev 1998).

Sterner et al. (1997), however, found that the theoretical result of Pimm and Lawton (1977) was primarily a methodological artifact resulting from differences in the number of negative (density-dependence) terms along the diagonal of the community matrix. Following the results of Sterner et al. (1997) studies exploring limits to food chain length have generally not explored the role of dynamic constraints. Most argue that dynamic constraints are less important to regulating the number of trophic levels compared to other mechanisms such as ecosystem size (Post 2002). Dynamic constraints due to colonization and extinction in a spatial context, however, have also been suggested as an alternative (Holt 2002) that seems to fit in with ecosystem size based hypotheses.

Nonetheless through systemic selection against dynamically unstable structures as suggested by Ulanowicz et al. (2013), dynamic constraints can provide a foundation for determining why the number of trophic levels is typically low. Systemic selection occurs when unstable food web structures (here referring to patterns of interactions) lead to the loss of some or all of the species in a web, thus altering web topology by eliminating nodes (species) and links (interactions). Food webs that are unstable are less likely to persist over time and more likely to undergo a change in species composition (e.g., through extinction) or interactions (such as by prey-switching). If systemic selection against unstable food web configurations leads to shorter food chains, food webs made of longer chains (meaning more trophic levels) should be less stable.

Webs of interactions that have a higher degree of stability, measured as quasi sign-stability (QSS; Allesina and Pascual 2008), should provide a buffer against changes in the magnitudes of interaction strengths resulting from stochastic environments, demography, and evolutionary change. Food webs that have greater QSS should be more persistent over time because the region of potentially stable parameter space will be larger, leading to a higher probability that the true values may remain within it. I hypothesize that webs with more trophic levels have lower QSS compared to webs with fewer trophic levels.

Omnivory, feeding on a range of trophic positions, was found by Pimm and Lawton (1978, 1977) to reduce the probability that a food web would be stable. This result, however, was given less weight compared to return time to equilibrium. They suggested that omnivory should be uncommon because chains that included omnivory were frequently unstable. Thompson et al. (2007), however, found that omnivory is common among species that occupy a trophic position higher than that of herbivores, with relatively few species occupying an integer trophic position (but see Thompson and Hemberg 2009). Likewise, other studies have found that anywhere between 46% (Williams and Martinez 2004) and 87% (Arim and Marquet 2004) of taxa in a given community feed on more than one trophic level. Furthermore, omnivory does not become less common when only looking at the strongest interactions in the web (Figure A1.1).

Below I examine how increasing food chain length (more trophic levels) impacts the degree to which the web is stable. I use food webs constructed at three levels of ecological realism; oversimplified chains with omnivory, random webs, and niche model food webs. The simplest example of chains with omnivory is used to demonstrate the expected relationship, while the random and niche model constructed webs allow us to further explore the impact of my assumptions, and determine under what conditions there is a relationship between food chain length and stability.

## Methods

The stability of food chains and food webs is typically determined by calculating the eigenvalues of the Jacobian matrix, whose elements $a_{ij}$ represent the impact of the population of species $j$ on the $i^{th}$ species' population (May 1972, Pimm and Lawton 1977, Sterner et al. 1997). In order to determine eigenvalues the matrix must first be evaluated based on data that most often are not available (e.g., interaction strengths, population sizes), a significant drawback.

Stability may also be determined based on the particular pattern of signs of the elements of the Jacobian matrix regardless of their magnitude. This can be useful for studies of food webs, because, while the interaction strengths and population sizes of the constituent species are rarely known, it is relatively simple to convert an adjacency matrix (a species by species matrix where a one denotes an interaction and a zero is no interaction) to a sign matrix. Food webs that consist of predator-prey interactions would assign each pair ($a_{ij}/a_{ji}$) a plus/minus. Purely linear food chains (where A only eats B, B only eats C, etc.), for example, are stable based solely on the signs of the elements of the matrix rather than their magnitude. Such chains or webs are termed qualitatively stable or sign-stable (May 1973b). Thus, regardless of the number of levels it contains a purely linear food chain will be stable. Hypothetically one could have a stable ecosystem with one hundred trophic levels, with species that consumed each other sequentially down the food chain.

A sign-stable food web must fulfill a set of specific conditions (May 1973b) that are not met by most webs in nature. For example, a sign-stable web cannot contain trophic loops (A eats B, B eats C, C eats A), a feature that is often found in food webs (Neutel et al. 2002). Nonetheless, following the very productive idea of Allesina and Pascual (2008), I am able to determine the degree to which a given food chain is sign-stable, termed quasi sign-stability (QSS). QSS is a measure of how often a matrix with a given sign structure is stable given elements ($a_{ij}$) whose magnitudes are randomized. While a sign-stable web will be stable regardless of the magnitudes of the elements of the Jacobian matrix (how large an impact each species has on those it interacts with), quasi sign-stable webs will only be stable for some range of magnitudes.

## General Model

The Jacobian matrix is found by taking the partial derivative of species growth equations. One of the benefits of using quasi sign-stability to determine the stability of food webs is that I do not need to explicitly define a model structure for the predator-prey equations. Rather, the elements of the Jacobian matrix can be sampled from an underlying distribution that can be based on the predictions from a general model.

I assumed a general predator dependent functional response (Abrams and Ginzburg 2000) where there is mutual interference among consumers. This means that the functional response is dependent on both the prey and predator densities, rather than only on prey density (as in prey-dependent). On the spectrum of interference between prey-dependence (no interference) and ratio-dependence (complete interference where the functional response is dependent on the number of prey per predator) assumed predator-dependent functional response is not as extreme as, but is closer to, ratio-dependence (Arditi and Ginzburg 2012).

Given “hungry” predators a ratio-dependent functional response is approximately linear and the dynamics of the system are donor-controlled (Arditi and Ginzburg 2012). In a donor-controlled system the mortality imposed by predation is independent of predator abundance (when abundance is high) in this idealized example. Predator abundance will therefore have no impact on prey abundance. The linear approximation of the ratio-dependent system away from saturation as a donor-controlled model can be considered similar to the way in which Lotka-Volterra equations represent a linear approximation of Hollings’ nonlinear model.

The sign matrix of a donor-controlled system is triangular, with positive values on one side of the diagonal and zeros on the other side. As May (1973) noted, a triangular matrix would imply qualitative stability, thus in this ideal scenario any number of trophic levels will create a stable structure, the same as in a purely linear food chain. With the assumption of a predator-dependent functional response that lies close to, but not at the ratio-dependent end of the spectrum in the numerical simulations I relaxed the conditions of a perfect donor-control model. Rather than assuming that predators have no impact on their prey I substituted small values relative to the impact of prey on predators. Thus the elements of the Jacobian matrix may be drawn from distributions that are asymmetrical, meaning that the distribution of the impact of the prey on the predator will have a mean that is larger (in absolute magnitude) than the distribution of the impact of the predator on the prey. For the random and niche model webs, I altered this assumption to test for a wider range of potential distributions varying in asymmetry.

## Web Construction

### *Simple webs*

To illuminate the hypothesis I start with a simple example of webs with two to six species arranged in a chain. These webs are characterized by universal omnivory, where each species consumes from all levels below its own, making the chain more web-like (Figure 2.2). For example, the fourth species eats species one, two, and three. Each web was then converted into a signed matrix, with (+/-) assigned to each predator prey link, a (+) indicating the positive effect of prey consumption on the predator’s population growth and a (-) indicating the negative effect of the predator’s consumption on the prey’s population growth. A (-) assigned to the diagonal, indicating negative density dependence, to avoid the problems illustrated by Sterner et al. (1997) Due to the use of universal omnivory, the sign matrices consist of positives on the upper triangle of the matrix, and negatives on the lower triangle.

### *Random webs*

Random ten-species webs were generated with five different levels of connectance (0.12, 0.16, 0.20, 0.24, and 0.28). Connectance is the proportion of all possible links between species that are realized (the number of links divided by the number of species squared). To ensure that I was sampling webs with all possible chain lengths the construction of each random web was started with a chain of two to ten species (one to nine links). Then, depending on the connectance, any remaining links were randomly assigned among species. For example, a ten species food web with a connectance (links per species squared) of 0.12 has 12 links. Thus a random web initiated with a ten species chain would have three more links randomly distributed, while a random web initiated with a five species chain would have eight more links distributed among the species. For these random webs, omnivory is included and increases with increasing connectance but is not universal as was assumed in the simple webs.

Each link was given a (+/-) to generate the sign matrix, with zeros indicating no interaction. Negatives were assigned to the diagonal. All random webs were constrained so that each species either consumed another species directly, or was consumed itself (i.e., no unconnected nodes). For each level of connectance 225 webs were generated.

I chose use a random model to construct these food webs for a thorough exploration of the parameter space of potential food web configurations. Food webs built using the cascade or niche model have similar structural properties to those that are observed in nature. Both the cascade and niche models arrange species along some hypothetical niche axis and use simple rules that determine what each species consumes to construct food webs. Using the random model I can include webs that one would not expect to find in nature, which is why it is useful to test my hypotheses. Additionally, many of the commonly found food web motifs are also found in these webs, such as tritrophic chains, apparent competition, direct competition, and intraguild predation. However, due to the way I built these webs there are no trophic loops.

### *Niche model webs*

The niche model (Williams and Martinez 2000) uses two parameters, the number of species and connectance, to construct food webs. Species are assigned a randomly drawn niche value and feeding range. The center of the feeding range is randomly placed at a point lower than the species' own niche value. Each species then feeds on all species whose niche values lie within the feeding range. Trophic loops are allowed in this model, because only the center of the feeding range must be below the given species' niche value up to half the feeding range can be on species with a greater niche value. It has been shown that the niche model accurately reproduces many structural properties of food webs (Stouffer et al. 2005, 2007, Camacho et al. 2007). As with the random model omnivory is included in these webs, but is not universal.

For each of five levels of connectance (0.12, 0.16, 0.20, 0.24, and 0.28) 105 ten-species niche model webs were generated. Each web was then converted into the corresponding sign matrix, and negatives were assigned along the diagonal for all species.

## **Simulations**

All simulations were completed in R version 3.1.1 (R Core Team 2014). Once webs were constructed, the signs of the interactions were replaced by random draws from predefined random uniform distributions. For each randomly drawn matrix the eigenvalue with the largest real part was calculated. This process was repeated multiple times for each web (see below for details). Quasi sign-stability could then be computed as the proportion of randomly drawn matrices with eigenvalues (largest real part) that were all negative.

### *Simple webs*

For the simplified webs I sampled the impact of the predator on the prey from a random uniform distribution between -1 and 0. The impact of the prey on the predator was drawn from a uniform distribution between 0 and 10. These distributions fulfill expectations based on the predictions of the general predator-dependent function response outlined above (asymmetrical impacts) but are otherwise not based on empirical data. For the diagonal of the Jacobian matrix, negative ones were assigned to all species. Thus density dependence was as large as the maximum possible impact of a predator on its prey for all species.

For each of the five sign matrices constructed I sampled 10000 Jacobian matrices and computed the eigenvalue with the largest real part for each. The proportion of these eigenvalue's real parts that were negative was then recorded as the web's quasi sign-stability.

*Random webs*

For each of the 225 webs per level of connectance the sign matrix was filled in with randomly drawn values from one of nine different pairs of distributions. All distributions were uniform, but were varied in their maximum/minimum values. The impact of the prey on the predator was set to a minimum of zero and a maximum of either one, five, or ten. The impact of the predator on the prey was set to a maximum of zero and a minimum of negative one, five or ten. This leads to nine possible combinations of the two distributions, along with the five levels of connectance used yields 45 distinct combinations of distributions and connectance. The values for the Jacobian matrices in the simple model were drawn from the respective distributions and used to compute the eigenvalue with the largest real part. This process was repeated 1000 times for each web and distribution combination. Quasi sign-stability was again recorded as the number of randomly sampled matrices whose largest real eigenvalue was negative.

*Niche model webs*

The same simulation procedure was completed for the niche model webs as for the random webs. I used the same uniform distributions for the relative impacts as well, from zero to one, five, or ten for the impact of the prey on their predator (and the same, but negative for the predator on the prey). For each pair of distributions I sampled values to fill in the sign matrix of each web 1000 times. Quasi sign-stability was determined as the proportion of those 1000 iterations that were stable, whose eigenvalue with the largest real part was negative.

## Results

### *Simple webs*

As the number of trophic levels in the chain increased, the probability of the chain being stable (QSS) decreased (Figure 2.3). The two-species case is a qualitatively stable food chain given my assumptions; it is a pure chain that is always stable. The largest decrease in QSS occurred between three (85.7%) and four levels (41.4%), dropping by 44%. Adding an additional level, to five species reduced the probability of a chain being stable by 31%, to nearly 10%. The chance that a chain of six species would be stable was only approximately 1%.

### *Random webs*

Asymmetry in interaction strength, and number of interactions impact the relationship of QSS and food chain length in random webs (Figure 2.4). As long as there is some degree of asymmetry in the interactions (10/-1, 5/-1, 1/-5, 1/-10), whether it leans toward the impact of the prey on the predator or vice versa, webs with longer chains on average are less likely to be stable at equilibrium (lower QSS).  When the relative impacts are drawn from symmetrical distributions the negative relationship of QSS and trophic level disappeared. Drawing from a symmetrical distribution (1/-1) that is equal in magnitude to the strength of density dependence, however, an increase in the longest trophic chain slightly increases quasi sign stability. Increasing the total number of interactions (connectance) always decreased stability with respect to any given maximum trophic position.

*Niche model webs*

There is a negative relationship between the longest food chain length and quasi sign-stability in for most of the simulated niche model webs (Figure 2.5). In webs with asymmetrical distributions there tends to be a negative trend that appears more consistently across different connectance levels when average interaction strength is lower. Asymmetrical webs with comparatively high interaction strength (e.g., 10/-5, 5/-10) the negative trend is not consistent across connectance levels. Drawing from a symmetrical distributions (10/-10, 5/-5) there was no clear trend between longest chain length and quasi sign-stability. Again, when density dependence was strong relative to the impacts of predation (1/-1) webs were consistently stable.

## Discussion

My results support the hypothesis that webs with more trophic levels are less likely to be stable. I argue that quasi sign-stability is a good predictor of the observed pattern of food chain lengths in nature. Food webs with fewer trophic levels have a larger range of parameter space (magnitudes of the elements of the Jacobian matrix) within which they exhibit stability; they have a greater buffer against perturbation due to environmental and demographic stochasticity. The ability to buffer against change has long-term consequences for the dynamics and structure of food webs observed in nature. Over time species participating in webs that are unstable will have a higher likelihood of going extinct, changing the structure of the web. In contrast webs that have a higher probability of being stable will likely persist, increasing the chances that they are observed in nature.

Here I have demonstrated that quasi sign-stability declines with increasing number of trophic levels in food chains made more reticulate by omnivory and in webs, reinforcing the results of Pimm and Lawton (1977). I found that two- and three-level webs should be most prevalent, with four- and five-level webs less likely, while webs with chains greater than six levels should be rare. Thus, my predictions closely mirror reality, where most non-basal species occupy trophic positions between two and three, and very few occupy a position higher than five (Vander Zanden et al. 1999, Beaudoin et al. 2001). Furthermore, in most scenarios webs whose longest chain length was high were typically much less likely to be stable than webs with shorter chains.

The primary focus of prior work on the impact of omnivory on the stability of food chains (Pimm and Lawton 1978, Morin and Lawler 1996, Holyoak and Sachdev 1998, Long et al. 2011) has been on whether or not omnivory should be observed in nature due its destabilizing impact on food chain dynamics. Alternatively, I have chosen to assume omnivory, in the sense of feeding on prey occupying a range of trophic positions, is prevalent, and investigate how that will impact the probability of observing food chain length patterns in food webs. I would not expect that predictions pertaining to purely linear food chains should match the observation of short food chains, and for the most part it has not (Sterner et al. 1997). While omnivory may not be as ubiquitous as I have assumed in for the simple webs (Thompson and Hemberg 2009), I propose that chains with universal omnivory are a more useful abstraction than linear chains because in nature food chains are embedded in reticulate webs. The predictions from the simple webs are reflected in the results of the analysis of random and niche model food webs.

My choice for the distributions from which values for the impact of species on each other were drawn was not based on empirical evidence. By using multiple different distributions for the relative impacts of the predator and prey, however, I showed that the results are relatively robust to the choice of the magnitudes of the impact of the predator on the prey and vice versa. When there is asymmetry in interaction strengths, and moderate levels of connectance the pattern is maintained. It is not surprising that simulations with weak asymmetry but relatively strong interaction strength (10/-5, 5/-10) showed a weak trend towards decreasing QSS with more trophic levels, following May's (1972) result.

May (1972) showed that highly connected random webs were less stable than those that were less connected. Similarly when the total number of interactions was high (24 and 28 interactions, connectance of 0.24, 0.28) all webs had a low probability of stability, with the exception of webs with high density dependence. As I have suggested for webs with more trophic levels, webs with high connectance values should be selected against. It is unlikely however, that systemic selection against highly connected communities is resulting in the observed pattern of food chain length. Highly connected webs are typically characterized by shorter food chains on average, and thus fewer trophic levels. One may then consider that the observed limitation of food chain length to three or four is the result of a balancing of dynamic constraints operating on both the number of trophic levels and the overall connectance of the web. Such a balance would need to be struck by shorter food chains, but lower connectance.

Dynamic constraints seem to offer a reasonable explanation for the observed pattern of food chain length in nature. I anticipate that the results of these simulations merely offer qualitative descriptions of the relationship between food chain length and stability rather than precise quantitative estimates. Nonetheless previous studies (Post et al. 2000, Post 2002, Young et al. 2013) have demonstrated that other possible mechanisms may be at work by finding relationships between chain length and productivity or ecosystem size. However, in these studies food chain length does not exceed 6 as predicted by the dynamic constraints hypothesis and their results often conflict.

Post et al. (2000) found that maximum trophic position increased with ecosystem size (measured as lake volume), but not productivity in lake ecosystems. In contrast, Young *et al.* (2013) found that food chain length increased with productivity but not ecosystem size (measured as island area) for tropical islands. The variation in trophic chain length in both studies, however, ranged between two and six. In a thorough analysis of food chain length variation in freshwater springs, Glazier (2012) showed that, for a biologically tolerable range of temperatures, food chain length does not change significantly and has a mean around 3.2. The systemic selection hypothesis eliminates many of these apparent contradictions by proposing a mechanism that predicts the distribution of food chain lengths observed most often in nature (Figure 2.1).

Ecosystem size and productivity influence population dynamics. It is somewhat intuitive that larger areas should support larger populations, on average. Similarly, theory based on predator interference predicts that with increasing basal productivity I would expect proportional increases in equilibrium population sizes for all trophic levels (Ginzburg and Akçakaya 1992). Larger populations are less susceptible to extirpation resulting from stochastically varying demography or environments. Thus, gains in the stability of populations through larger numbers of individuals may buffer against losses in QSS resulting from increased length of chains. Different types of environment (e.g., lakes or islands) may also differ in the relative importance of variables such as ecosystem size and productivity for population size. Alternatively ecosystems of greater size may offer a greater opportunity for patch dynamics where longer chains may exist in a spatial context (e.g., by top predators being present in some patches but not others; Holt 1996, 2002, Polis et al. 1997). The contradictory patterns observed by Post et al. (2000) and Young et al. (2013) may therefore be explained by slight modification to the general hypothesis of dynamic constraints and systemic selection I have outlined above.

**Figure 2.1:** The trophic position of each species in 50 published food webs (a) and the length of the single longest chain in each web (b). The data sources and code for generating this figure can be found in Appendix 1.

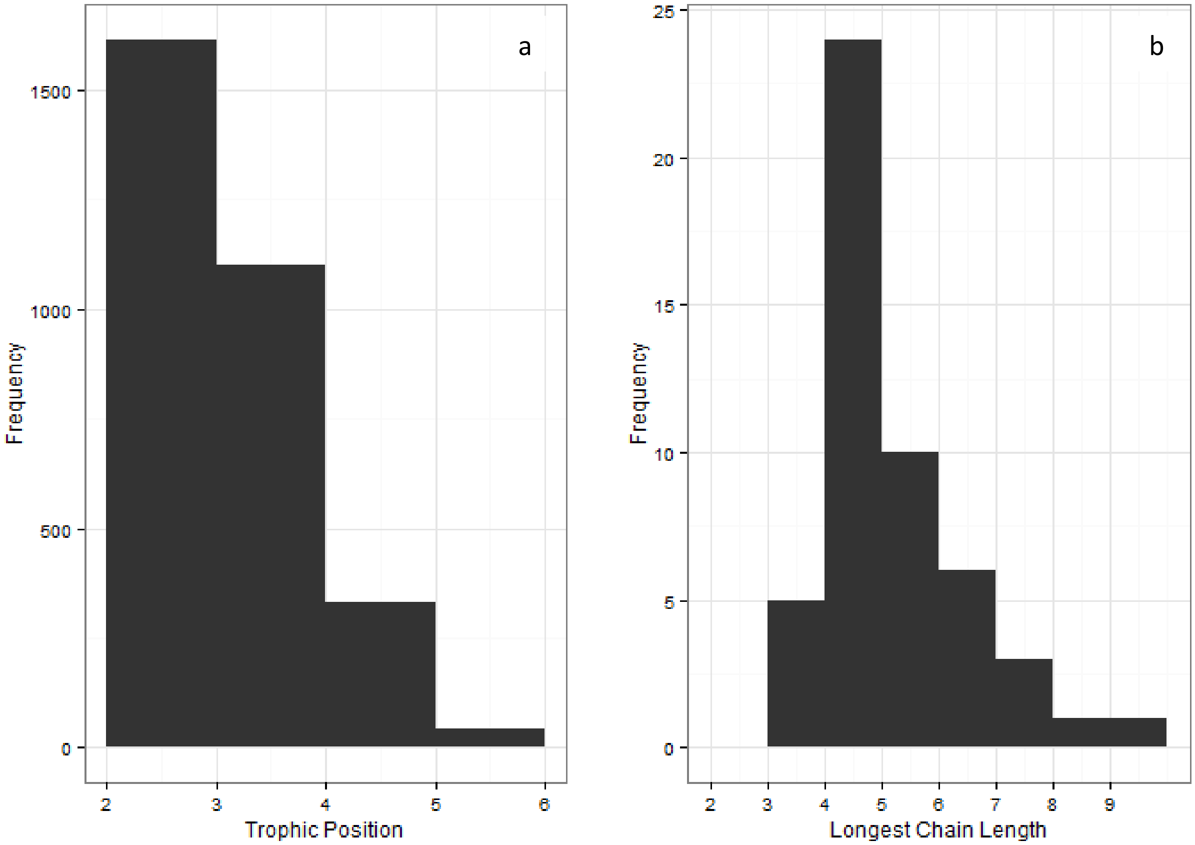


**Figure 2.2:** The five simple food webs with two to six trophic levels. The measured longest chain is highlighted in bold. Code for making these webs and generating this figure can be found in Appendix 1.

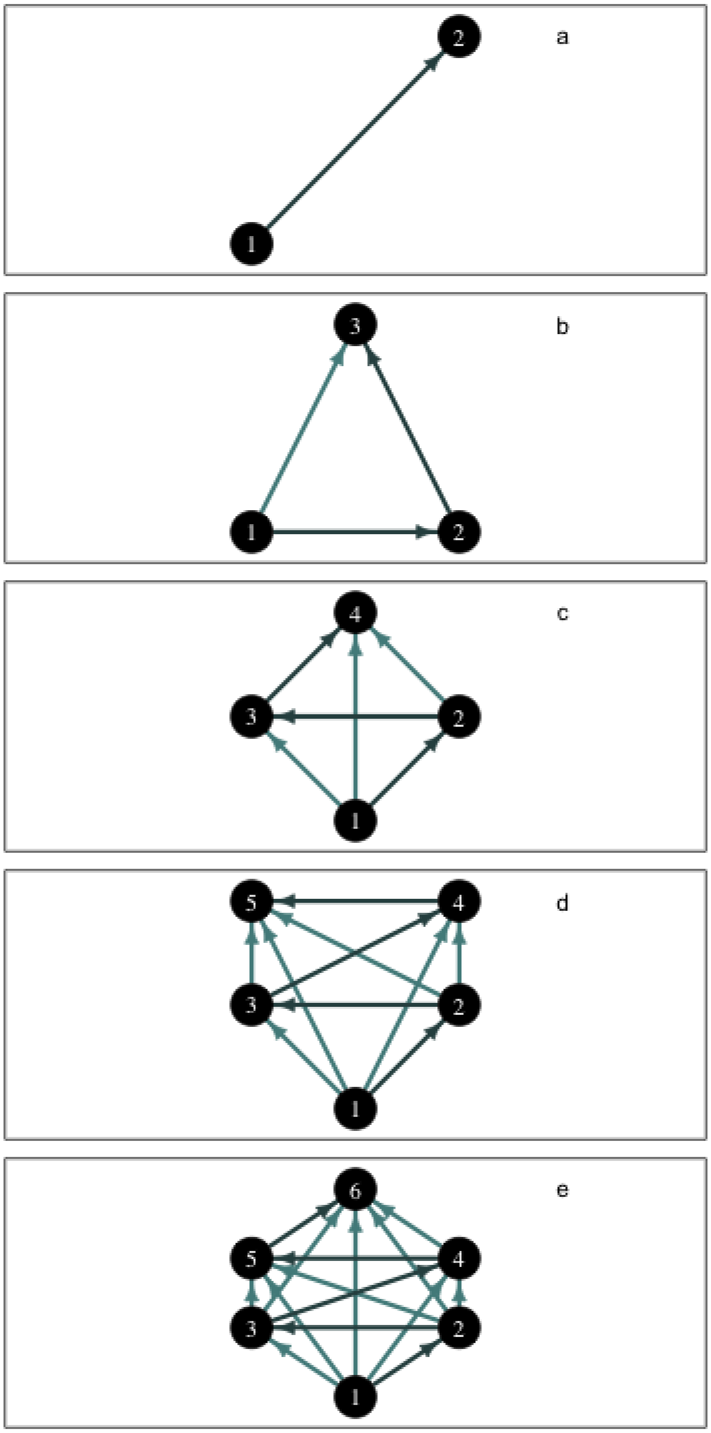


**Figure 2.3:** The quasi sign-stability of the simple webs with two to six trophic levels, pictured in Figure 2.2. Quasi sign-stability (QSS) is the proportion of matrices with randomly sampled elements ($a_{ij}$) that have an eigenvalue whose largest real part is negative. Code for the simulation and producing this figure can be found in Appendix 1.

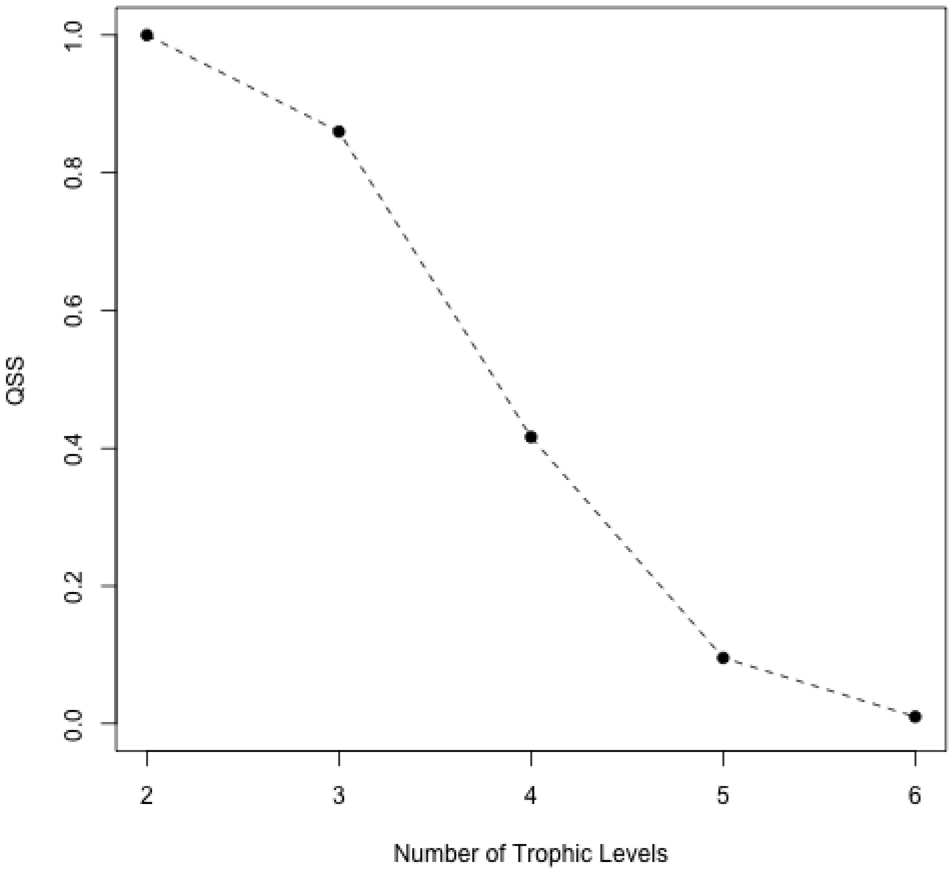


**Figure 2.4:** Plot of quasi sign-stability against longest chain length in random food webs. Gray dots are individual data points (data have been jittered to avoid overlap) while black dots are the mean. Error bars are 95% confidence intervals calculated as the mean plus/minus 1.96 times the standard error. Columns represent different levels of connectance (labelled across the top) while rows represent the different parameters used to create the distributions of relative impacts. The rows are labelled according to *impact of the prey on the predator*/*impact of the predator on their prey* with each value being the maximum/minimum of the uniform distribution used. The code to create this figure can be found in Appendix 1.

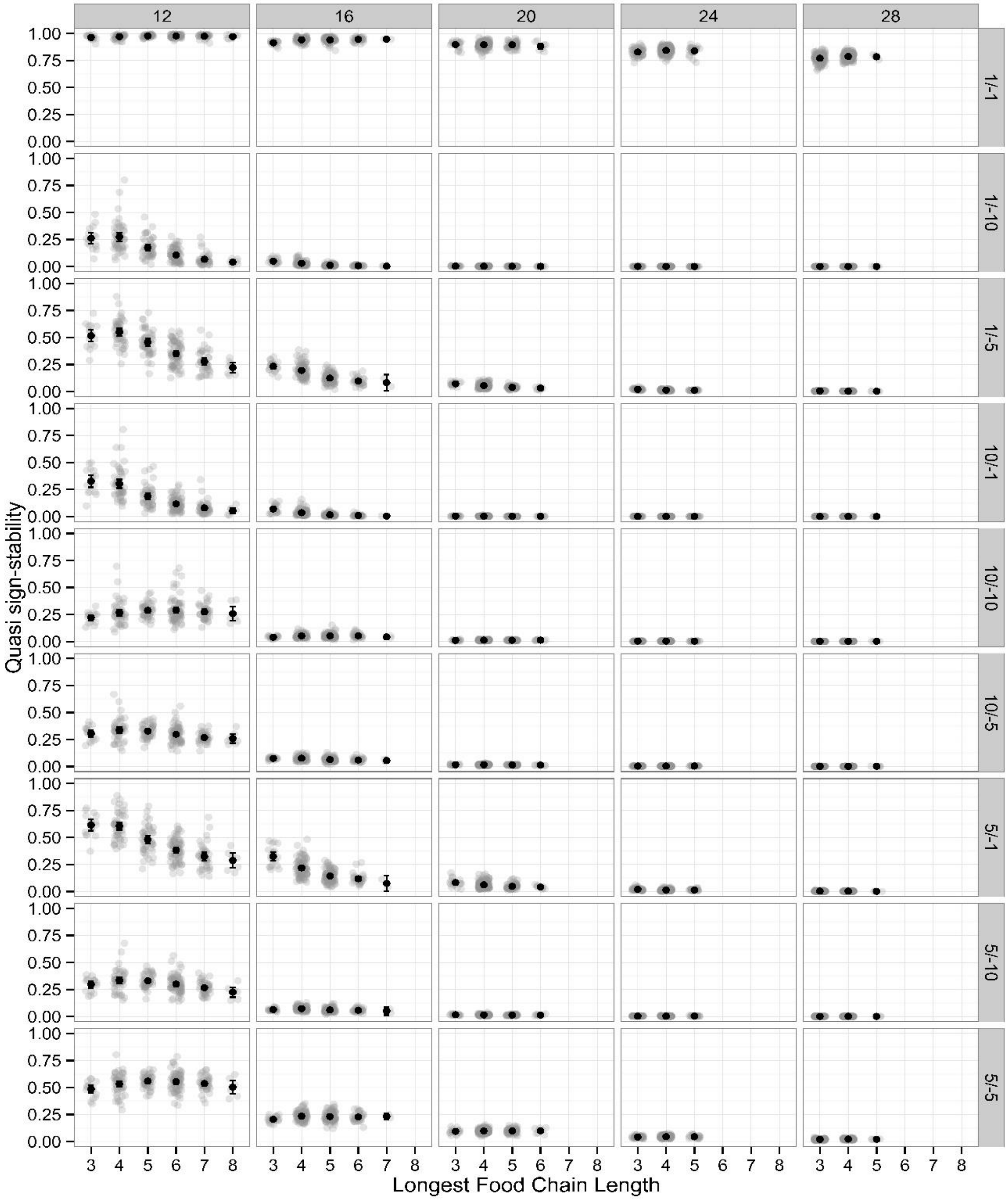
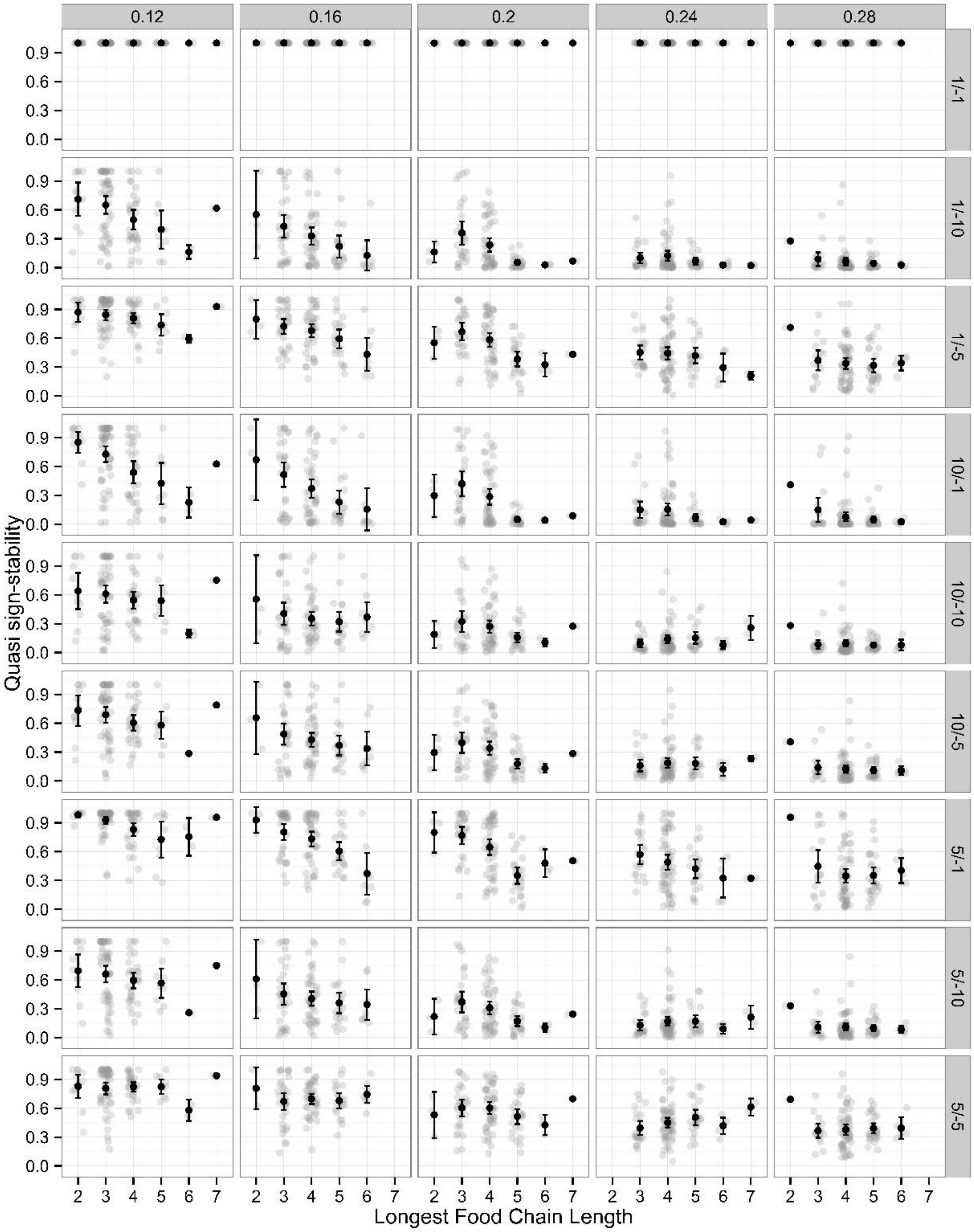


**Figure 2.5:** Plot of quasi sign-stability against longest chain length in niche model food webs. Gray dots are individual data points (data have been jittered to avoid overlap) while black dots are the mean. Error bars are 95% confidence intervals calculated as the mean plus/minus 1.96 times the standard error. Columns represent different levels of connectance (labelled across the top) while rows represent the different parameters used to create the distributions of relative impacts. The rows are labelled according to *impact of the prey on the predator*/*impact of the predator on their prey* with each value being the maximum/minimum of the uniform distribution used. The code to create this figure can be found in Appendix 1.



# Chapter 3: Selection against instability: stable subgraphs are most frequent in empirical food webs

## Abstract

Food web structure can be characterized by the particular frequencies of subgraphs found within them. Although there are thirteen possible configurations of three species subgraphs, some are consistently over-represented in empirical food webs. This is a robust pattern that is found across marine, freshwater, or terrestrial environments. The preferential elimination of unstable subgraphs during the assembly of the food web can explain the observed pattern. It follows from this hypothesis that there should be differences in the stability of different subgraphs, and that stability should be positively correlated to their frequency in food webs. Using 50 food webs collected from a variety of databases I determined the frequency of each of the thirteen possible subgraphs with respect to randomized webs. Then by numerical simulation I determined the quasi sign stability (QSS) of each subgraph. My results clearly show a positive correlation between QSS and over-representation of the different subgraphs in empirical food webs.

## Introduction

Much of the past 40 years of food web ecology has focused on finding the "devious strategies which make for stability in enduring natural systems," to quote Robert May (1973a). There is a clear link between those structures observed in ecological networks (e.g., modularity in food webs and nestedness in mutualistic webs) and the population dynamics of the species that make them up (Drossel et al. 2004, Bastolla et al. 2009, Thébault and Fontaine 2010). While large scale emergent properties of networks (modularity, nestedness, connectance, etc.) have been the focus of much research in the past, there are many other ways to characterize the structure of an ecological network. Here I assess the substructural composition of food webs in light of the stability of their basic building blocks (three-node subgraphs).

Larger networks are made up of many smaller subnetworks (hereafter subgraphs) that are assembled together. Any large network with *N* nodes can be decomposed into smaller networks of size 1 to *N*-1. There are thirteen possible (connected, directed) configurations of three nodes, five of which require only single direction links and 8 which combine single and bi-directional links (Figure 3.1). Milo et al. (2002)  showed that the observed frequency of these three node configurations varies in different types of networks (e.g., food webs, neural networks, electronic networks, etc.). For example, in food webs the tritrophic chain tends to be over-represented compared to random. Subgraphs that tend to be over-represented are commonly termed *motifs*.

In a study of 16 food webs Stouffer et al. (2007) showed that there were two distinct patterns of the frequency of subgraphs in food webs. Most webs (10 of 16) were characterized by over-representation of intra-guild predation and under-representation of apparent and direct competition. The other 6 webs were characterized by the opposite pattern. Overall most webs tended to have similar relative frequencies of the different subgraphs.

The two prevailing hypothesis for the pattern of subgraph frequencies are: (1) that there are constraints in the assembly of a network, and (2) that there is some advantage to motifs that make them more likely to become over-represented (Prill et al. 2005, Camacho et al. 2007). While these two hypotheses are not mutually exclusive, Camacho et al. (2007) and Prill et al. (2005) offer different perspectives on which is more important for generating the observed pattern. Camacho et al. (2007) found that because food web models are able to reproduce observed subgraph patterns, constraints on food web generation are most important. Alternatively, because structure is related to function, Prill et al. (2005) hypothesize that it is the properties of the subgraph that lead to its abundance in a network. I suggest that a combination of the two hypotheses in the form of selection against unstable structures offers the simplest explanation for the occurrence of particular motifs. Such a process would eliminate those subgraphs that are unstable with greater frequency than expected by chance, while increasing the relative abundance of those that are more likely to be stable.

Many definitions of stability have been used in the ecological literature (Pimm 1984, Donohue et al. 2013). Here I focus on the unity of two forms of mathematical stability, local (or eigenvalue) stability and qualitative stability. A system is locally stable if all of the eigenvalues of the Jacobian matrix (evaluated at equilibrium) have negative real parts. Functionally this means that following a small perturbation from equilibrium, the system will return to the original equilibrium state. Local stability has been frequently used in the theoretical ecology literature to understand the relationship between structure and stability (May 1972, Pimm and Lawton 1977, Sterner et al. 1997, Allesina and Tang 2012). Determining local stability, however, requires knowledge of the magnitudes of the elements of the Jacobian matrix (where each element represents the impact that species *i* has on the $j^{th}$ species). A qualitatively stable system is one that is stable based only on the signs of the elements of the Jacobian matrix, rather than their magnitudes (May 1973b). In order for a food web to be qualitatively stable the Jacobian matrix must fulfill certain conditions, such as having at least one negative term along the diagonal. Most ecological communities do not fulfill the conditions for qualitative stability (May 1973b), for example many food webs contain trophic loops of three or more species.

Allesina and Pascual (2008) linked these two concepts with the development of quasi sign-stability (QSS). QSS is a measure of how robust a system is to changes in the magnitude of the elements of its corresponding Jacobian matrix. Thus, the more quasi sign-stable a system is, the closer it is to being qualitatively stable. If a number of matrices are created by sampling the values of the Jacobian from a distribution (preserving the sign structure) the proportion that are locally stable is the system's QSS. Using this measure I can determine the tendency of a given sign structure, determined by the configuration of the subgraph, to be stable.

In this study I asked whether the commonly observed pattern of subgraph frequency fits with what is expected based on the hypothesis of selection against unstable configurations. To answer this question I examined the variation in both frequency and the QSS of the thirteen different three-node subgraphs. Where other studies comparing subgraph frequency to their dynamic properties have used absolute frequencies (Prill et al. 2005), I computed frequency relative to a null model. If there is a systemic selection against unstable subgraphs in nature, then there should be a positive correlation between the frequency of a given subgraph and the probability that that subgraph will be stable.

## Methods

### Data

I used 50 food webs collected from a variety of sources. Three food webs were downloaded from the Dryad Digital Repository (Roopnarine and Hertog 2012a, b). Another seven were available from Ecological Archives (Hechinger et al. 2011, Thieltges et al. 2011, Zander et al. 2011, Mouritsen et al. 2011, Preston et al. 2012). These seven webs all included parasitic links, so to compare these webs to food webs without parasites only predator-prey links were considered. Fourteen webs were also provided by Jennifer Dunne of the PEaCE Lab (Baird and Ulanowicz 1989, Warren 1989, Polis 1991, Hall and Raffaelli 1991, Martinez 1991, Christensen and Pauly 1992, Havens 1992, Goldwasser and Roughgarden 1993, Opitz 1996, Waide and Reagan 1996, Yodzis 1998, 2000, Martinez et al. 1999, Christian and Luczkovich 1999, Memmott et al. 2000, Link 2002). The remaining 26 food webs were downloaded from the Interaction Web Database (Townsend et al. 1998, Jaarsma et al. 1998, Thompson and Townsend 1999, 2000, 2003, 2005, Thompson and Edwards 2001).

### Subgraph Frequency

The frequency of each subgraph was found using the *triad.census* function from the *igraph* package (Csárdi and Nepusz 2006) in R version 3.0 (R Core Team 2014). Counts of each of the thirteen subgraphs were determined for each of the 50 food webs described above. The frequency of each subgraph was then compared against a null distribution.

To create the null distribution each of the fifty adjacency matrices (each food web) was permuted using the recently described Curveball algorithm (Strona et al. 2014). This algorithm maintains both the number of predators of a species has and the number of prey a species has (maintaining row and column sums) and has been shown to be unbiased in the creation of null distributions (Strona et al. 2014). As a second null model additional constraints were added to the Curveball algorithm to maintain not only the row and column sums, but also the number of single, double, and self (cannibalistic) links in each web. This is similar to the null model used by Stouffer et al. (2007), but, where they preserve the number of single, double, and cannibal links per species, the modified Curveball algorithm I use preserves the number of each type of link per web.

The frequency of each subgraph was determined in each of thirty thousand permuted matrices that were generated for each food web using both null models. Z-scores were computed using the formula:

$$z_i = \frac{X_i - \overline{X_\imath}}{\sigma_l} \qquad (eq. 3.1),$$

where $X_i$ is the frequency of the *i*$^{th}$ subgraph in each empirical food web, $\overline{X_\imath}$ is the mean frequency of the *i*$^{th}$ subgraph in the permuted matrices, and $\sigma_l$ is the standard deviation. The z-scores were then normalized by dividing by the square root of the sum of the squared z-scores for that food web following previous studies of motifs by Milo et al. (2002) and Stouffer et al. (2007),

$$n_i = \frac{z_i}{\sqrt{\sum z_j^2}} \qquad (eq. 3.2).$$

The code for determining subgraph frequency can be found in Appendix 2.

**Subgraph Stability**

Using numerical simulations in R *version 3.0* (R Core Team 2014) I determined the probability that a given subgraph would be stable (quasi sign-stability, QSS). The adjacency matrix of each subgraph was translated into its corresponding sign-structured matrix. Only predator-prey interactions were considered, so if $a_{ij} = 1$ then $a_{ij}$ will be a positive while $a_{ji}$ is negative. When interactions were bidirectional, both $a_{ij}$ and $a_{ji}$ were positive (if negative then quasi sign-stability is increased, see Appendix 2). Positive elements of the matrix were given a random value drawn independently from a uniform distribution between 0 and 10, while negative elements were given a value drawn independently from a uniform between -1 and 0.

These distributions were chosen to reflect asymmetry in the relative impact of the prey on the predator (positive) and the predator on the prey (negative). Asymmetry can be derived from a general predator-dependent functional response (a functional response that is dependent on both prey and predator densities) where the predator is not saturated ("hungry predators"). Note that the particular structure of the equations governing population dynamics was not defined. Instead, assumptions about the values of the Jacobian matrix were made based on what may be predicted from the general form of the function.

Diagonal elements have a large impact on the stability of the matrix (Sterner et al. 1997), so each diagonal element was assigned a random value drawn from a uniform distribution between -1 and 0. Thus the average intraspecific effect on population growth rate was approximately the same as the effect of predation. By including density dependence for each species I allow the greatest chance of stability.

This process was repeated 10,000 times for each subgraph. Quasi sign stability was then calculated as the proportion of randomly sampled matrices whose eigenvalue with the largest real part (*Max(Re(λ))*) was negative (Allesina and Pascual 2008). See Appendix 2 for details on the code used for simulations.

## Results

Subgraphs differed in how frequently they were observed relative to permuted webs (Figure 3.2). Tritrophic chains (s1), apparent competition (s4), and direct competition (s5) all tended to be over-represented in food webs based on both null models. Intra-guild predation (s2) was typically under-represented, more so when compared to webs generated by the modified Curveball algorithm, but was over-represented in some webs. The three-species trophic loop (s3) was under-represented in all webs, as were most subgraphs that include bidirectional links (A eats B, B eats A). When the numbers of each link type were preserved the d1, d2, and d6 subgraphs were over-represented in the twelve webs that had double links.

There was also variation in the quasi sign-stability of each subgraph (Figure 3.3). The three subgraphs that tended to be over-represented were all very likely to be stable, a result that is robust to the choice of sampling distributions for the randomized matrices (see Figure A2.1). Intra-guild predation was also moderately likely to be stable (approximately 0.5). The remaining subgraphs were all unlikely to be stable, with quasi sign-stability less than 0.2. Subgraphs with a higher quasi sign-stability occurred more frequently than expected by chance, while those with lower quasi sign-stability tend to occur less frequently than expected by chance.

## Discussion

If there is a selection against unstable food web configurations, then those configurations that are most likely to be stable in a varying environment should be those that are most commonly observed in nature. In order for such a process to work, different food web configurations must be variable in their stability. A signature of this selection against instability should then be if variability in the stability of different food web configurations is correlated with their observed frequency.

The signature of selection against unstable configurations is clearly seen by comparing the observed frequencies of three-node subgraphs and their quasi sign-stability. Those subgraphs that are most quasi sign-stable are also those that are observed more frequently than expected by random chance. Subgraphs that are less quasi sign-stable are typically either under-represented (observed less frequently than expected by chance) or observed about as frequently as expected by random chance, although this depends on the null model to some extent. A similar finding of a relationship between subgraph frequency and contraction loss (a measure of stability) mirrors my results (Angulo et al. 2014). While this does not provide concrete proof that there is a selection against unstable subgraphs in nature, it does offer convincing evidence that such a process could affect the way food webs are structured.

The null model used to generate the subgraph frequencies does have an effect. In part this explains why the pattern of frequencies is different from those determined by Stouffer et al. (2007). Additional differences may also be the result of the larger sample of food webs, 50 compared to 16. The frequency of subgraphs that include double links is especially of interest when comparing the outcome from the two different null models used (Figure 3.2). In particular, the d1, d2, and d6 subgraphs occur more frequently than expected at random when the number of single, double, and self links are constrained. However, while this is true for those webs that have double links, the occurrence of double links is rare with only 12 of 50 webs having them. In both null models the tritrophic chain, apparent competition, and direct competition are consistently over-represented.

This study represents the third case in which quasi sign-stability has shown to be a useful measure to help explain observed food web patterns (Allesina and Pascual 2008, Borrelli and Ginzburg 2014). Quasi sign-stability is a simple yet powerful measure when considering stability in a stochastic environment. As a result of environmental and/or demographic stochasticity any given parameterization of the community matrix (to determine local stability) is not likely to be informative. Yet, quasi sign-stability can reveal a considerable amount about the system, given just the sign-structure of the community.

The simplicity of the argument underlies its usefulness to ecologists. Given knowledge of the quasi sign-stability of community configurations an expectation of what should be observed in nature can be built. In their study introducing the concept of quasi sign-stability Allesina and Pascual (2008) found that when predator-prey links dominate the community, the community should be more stable. My results show that tritrophic chains, direct competition, and apparent competition should be expected more frequently; while Borrelli and Ginzburg (2014) demonstrated that short food chains should be more frequent than longer ones. The results of this study show that these expectations match what is actually observed in nature, as was shown for food chains by Borrelli and Ginzburg (2014).

If observed pattern does not match my expectation based on quasi sign-stability, it may be an indication of some underlying biologically interesting phenomenon. For example, while the intra-guild predation subgraph (A eats B, A eats C, B eats C) is moderately quasi sign-stable, the expectation is that it should be somewhat over-represented in the data. Yet I found that intra-guild predation is mostly under-represented, albeit with a large variance. Likewise, in a few food webs apparent competition is under-represented yet is very likely to be stable.

One potential explanation for the lack of intra-guild predation subgraphs is one of feasibility. An equilibrium point would be feasible if all species had positive densities. In the present study I have not considered feasibility constraints, as I chose not to impose any particular equation structure, rather relying on the general case of a predator dependent functional response. It is possible that for species to maintain positive densities in an intra-guild predation framework there are constraints on the magnitude of the elements of the Jacobian, and that region of “parameter space” could be smaller than the region that is considered stable.

Alternatively, constraints could be imposed at higher levels of organization. As different subgraphs are assembled into larger networks, there could be interactions among them, which alter their probability of being stable. The next step in understanding the relationship between motifs and stability will be to learn how the stability properties of the three-node configurations scale up to larger networks. I expect that, in general, networks with subgraph patterns that show over-representation of subgraphs that are more quasi sign-stable should themselves be more stable.

Recent work has highlighted the role of trophic coherence in stabilizing large food webs (Johnson et al. 2014). Trophic coherence is a measure of the homogeneity of the distribution of the trophic distances between predators and their prey. A linear food chain, where species A eats B, B eats C, and C eats D, is a perfectly coherent web. Johnson et al. (2014) showed that stability increases with size and complexity in food webs that are trophically coherent. Therefore it is worth noting that those subgraphs that are most stable, the tritrophic chain, apparent competition, and direct competition are those that are trophically coherent. If webs that are more coherent are more stable, then webs built with coherent components should be more stable.

Like the pattern of subgraphs in food webs, trophic coherence may be explained by the two hypotheses of constraints on the assembly of the web, or intrinsic benefits of the structure. Johnson et al. (2014) suggest that adaptations for feeding on a given species may be more helpful for preying upon other species at a similar trophic level, evoking the idea of constraints on assembly of the web. The observed frequency of subgraphs may then be the result of such feeding preferences and the apparent relationship between stability and frequency only coincidental. One could also imagine that species vary in how they choose prey and when species feeding preferences lead to the increase of s1, s4, and s5 subgraphs the coherence of the web and therefore its stability increases. The observed pattern would then result from a combination of constraints on assembly and the intrinsic properties of the subgraphs (stability).

An ideal test of the hypothesis of selection against unstable configurations would involve food web assembly data. During food web assembly, new species enter the food web and establish feeding links with the species that are present. This establishment alters the structure of the food web, and can be measured as changes in the relative frequency of the different three-node subgraphs. Following the introduction of the new species one of two events could occur; (1) the new species becomes integrated into the network and the food web grows (in terms of the number of species), or (2) the introduction of the new species leads to extinction events and the food web shrinks. The selection against instability hypothesis would predict that when the addition of new species increases the relative frequency of more stable subgraphs compared to less stable ones, the web should grow. If the frequency of less stable subgraphs is increased more than the stable ones, then extinctions would be more likely to occur.

**Figure 3.1:** Graphical representations of the thirteen possible configurations of 3 species ordered by decreasing quasi sign-stability. The five with single links only have the “s” designation while those including double links have a “d”.

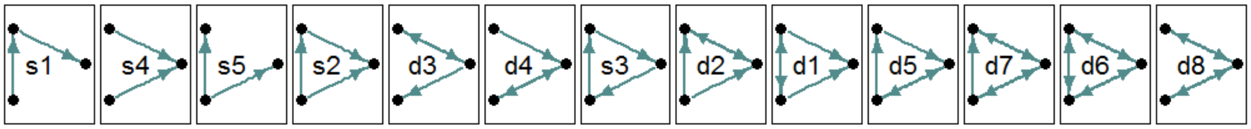


**Figure 3.2:** The normalized profile of the 13 possible 3-node subgraphs in a set of 50 food webs arranged in order of decreasing quasi sign-stability. Boxplots represent the median and interquartile range of the normalized z-scores generated using the Curveball algorithm (red) and the modified Curveball algorithm (blue). Whiskers extend to the most extreme point within 1.5 times the interquartile range.

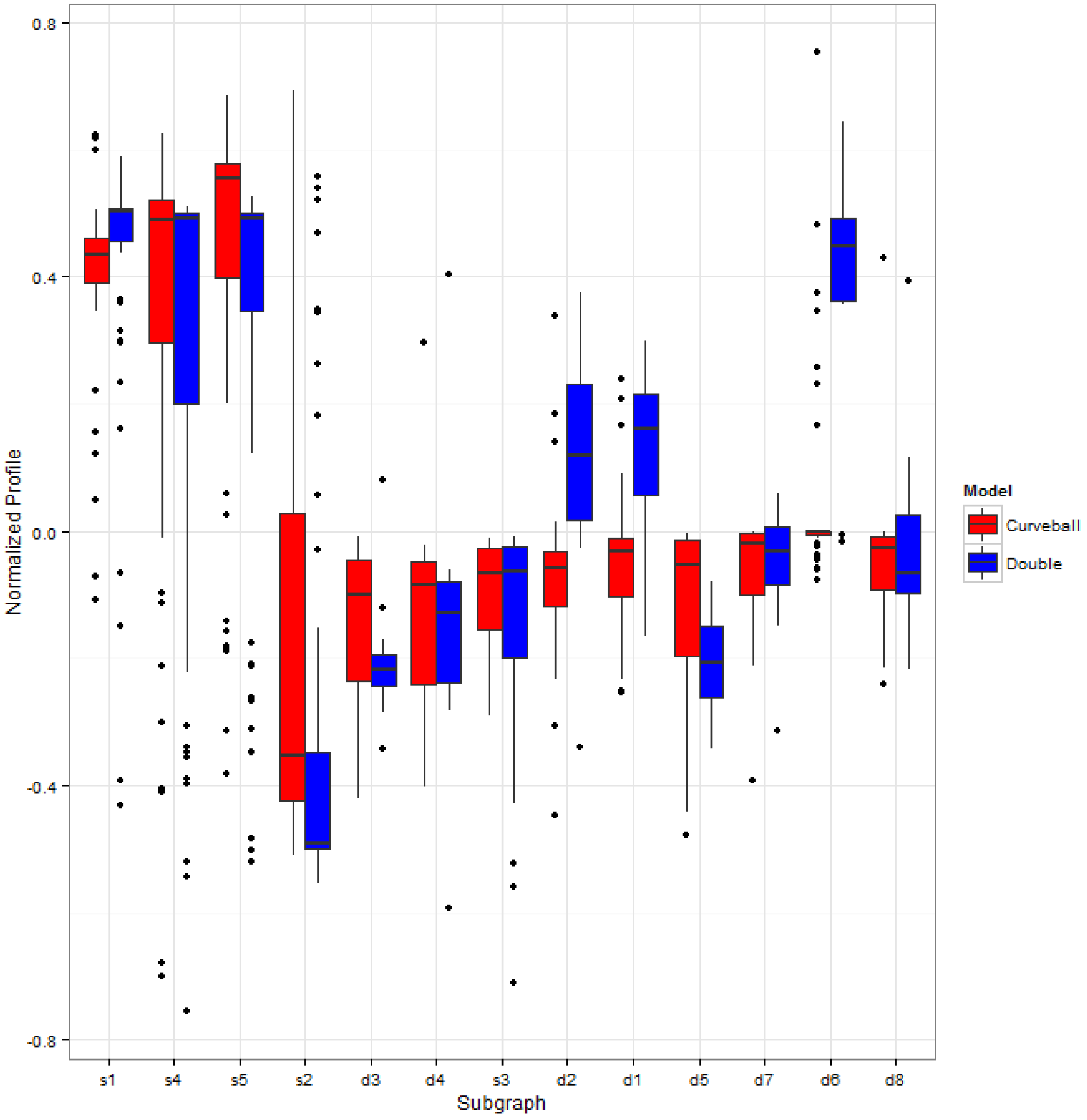
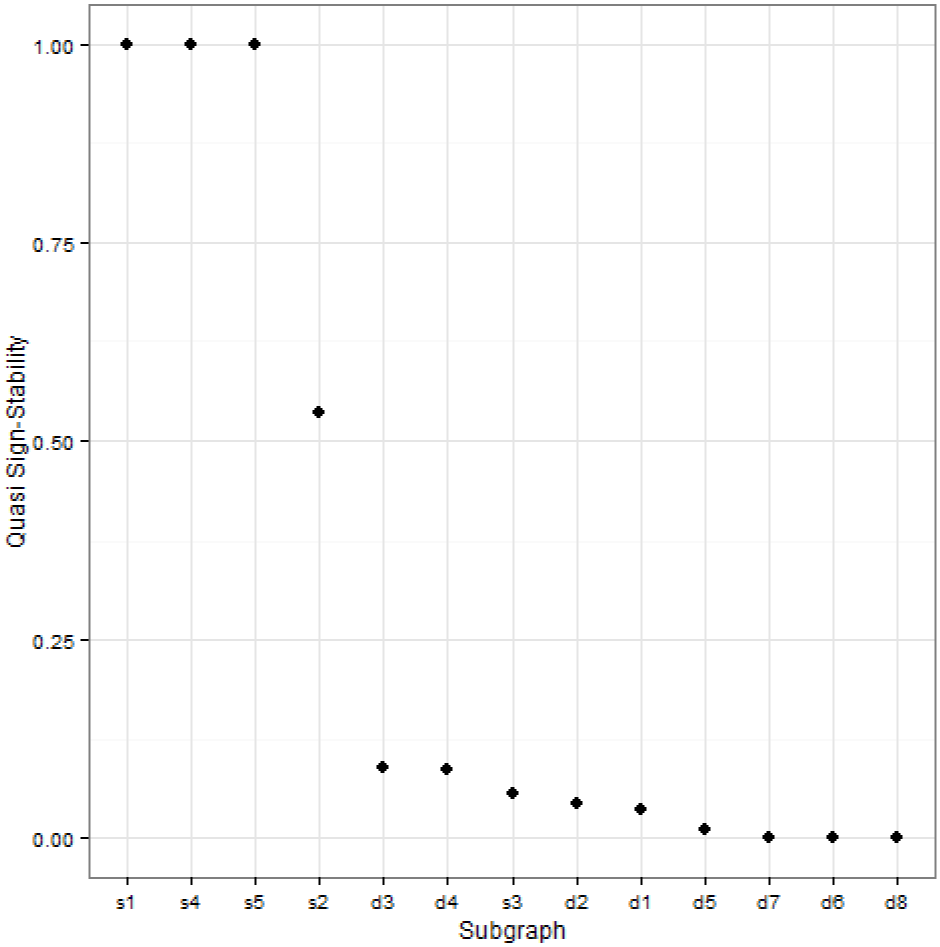


**Figure 3.3:** Quasi sign-stability for each subgraph, determined as the proportion of randomly parameterized sign matrices that were locally stable. Subgraphs have been ordered along the x-axis from greatest to least quasi sign-stability.



# Chapter 4: *rend*: An R package for Ecological Network Dynamics

## 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 resources 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 Chapter 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.

## Introduction

The structure of an ecological network is both the product and driver of the dynamics of the interacting species. How many species there are, the manner in which they are connected, as well as the strength of those connections make up the structure of the network. This also then defines how different species impact each other's population growth and decline. As a direct result of these individual population dynamics, the structure of the network can change. Species can be lost via local extinction, new species can be gained via introduction/invasion, and connections among species can be altered as a result of prey-switching and adaptive foraging. An important question in ecology is how to go about enumerating this intimate link between structure and dynamics of ecological networks.

Perhaps the most obvious answer to this question is to go out and collect data. This requires time-series data of ecological networks. The problem is that these data are very difficult to acquire and require a great deal of effort, time, and money. There have been some studies, however, that have focused on the change in network structure through time, primarily in plant-pollinator systems ( Alarcón et al. 2008, Olesen et al. 2008, Dupont et al. 2009, Albrecht et al. 2010). There are fewer data available on changes in the structure of food webs over time, although Fahimpour and Hein (2014) used a mesocosm approach to quantify the food web at different time points during food web assembly. In order to circumvent the problem of difficult data collection, many ecologists have turned to models of multi-species communities to generate more easily testable predictions.

Multi-species interaction models have been used to assess a variety of hypotheses related to the structure and stability of ecological communities. Fundamental research has focused on the equilibrium dynamics of communities by assessing the Jacobian matrix, which is a matrix of the partial derivatives of species population growth equations reflecting their relative impact on one another. Following May's (1972, 1973) pioneering approach, these studies focus either on stability alone (Allesina and Tang 2012, Tang and Allesina 2014, Borrelli 2015) using random matrix theory (James et al. 2015), or on the stability of communities constrained to be feasible (Roberts 1974, Christianou and Kokkoris 2008, Saavedra et al. 2016).

In food web studies, non-equilibrium approaches are utilized through assembly- (Lockwood et al. 1997, Côté and Parrott 2006) or evolution- (Caldarelli et al. 1998, Loeuille and Loreau 2010, Stegen et al. 2012) 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 (Romanuk et al. 2009, Baiser et al. 2010, Lurgi et al. 2014).

Assembly-based 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. (2001) 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 response) result in more realistic food webs (Drossel et al. 2004). Loeuille and Loreau (2005), using another evolution-based model, showed that the size-structure often observed in natural communities is an emergent property of community development.

A problem with many current modeling approaches is that the theoreticians that are developing them offer a mathematical black box. Models are run in a variety of languages, such as Java (Romanuk et al. 2009), C/C++ (Thébault and Fontaine 2010), R (Lurgi et al. 2014), and Mathematica (Baiser et al. 2010). The source code, however, rarely gets included with these publications. The standard information given is the mathematical equation describing the model. For many biologists this is not useful (e.g., Fawcett and Higginson 2012), especially as it is not always straightforward how to implement these equations in a computer language.

In ecology, the use of the R programming language is rapidly outpacing other languages. This growth is likely tied to the increasing number of packages available in R for the analysis of ecological data. Many of the R packages associated with ecology are related to new statistical methods. For networks, there is a range of packages available to assess the topological properties of networks. One of the most widely used packages to assess network structure is *igraph* (Csárdi and Nepusz 2006), which is not ecologically focused but offers many of the mathematical tools developed to detect structural features of networks. The *bipartite* package (Dormann et al. 2008) is focused on two-level networks, like plant-pollinator or host-parasite networks (Dormann et al. 2009). The *betalink* package was designed to assess the beta-diversity of species interaction networks (Poisot 2015). Ecosystem networks, which are networks of nutrient and energy flows, can be analyzed using *enaR* (Borrett and Lau 2014). Packages like *foodweb* (Perdomo et al. in press) and *cheddar* (Hudson et al. 2012) were developed to explicitly compute the structure of food webs, and include metrics specific to food web theory (e.g., trophic level, and predator-prey ratios). Other packages related to the analysis of ecological networks were developed for fairly specific functions, including *NetIndices* (Kones et al. 2009), which provides functions to determine whole network properties and trophic indices for each species, and *rnetcarto* (Doulcier and Stouffer 2015) for determining modularity and module participation.

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 useful tools to find what Robert May called the "devious strategies which make for stability in enduring natural systems" (May 1973, p.174). 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 are many packages designed to determine the structure of networks using R, there is a dearth of R packages available to apply dynamic models to network structures. To my knowledge, only one package is available to model multispecies predator-prey dynamics on food webs, the now outdated *gruyere* package, which was developed to work in conjunction with the cheddar package, but is no longer available in current versions of R.

In this Chapter I present an in-development R package, *rend*, 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, with both a simple two-species system, to demonstrate how altering parameters affects the dynamics, and an example of a simulation that uses a niche model (Williams and Martinez 2000) 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.

## Package Overview

There are three core components of the *rend* package: simulation, visualization, and analysis. Below I describe the model used to simulate trophic dynamics and its implementation in code, 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 change in the trophic structure of the food web. All of the required code for this package can be found in Appendix 3.

## 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 (2005). 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 (2005).

### *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 (1992), and generalized to multiple species by McCann et al. (1998). The model structure included in this version of *rend* was first presented in Williams and Martinez (2005) and used again in Romanuk et al. (2009) to predict success of invaders in model communities. The change in species' total biomass over time is defined as

$$\frac{dB_i(t)}{dt} = G_i(B) - \ x_iB_i(t) + \sum_{j=1}^{n}(x_iy_{ij}F_{ij}(B)B_i(t) - \ x_jy_{ji}F_{ji}(B)B_j(t)/e_{ij}) \qquad (eq. 4.1).$$

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 according to

$$G_i(B) = r_iB_i(t)\left(1 - \frac{B_i(t)}{K_i}\right) \qquad (eq. 4.2),$$

where $r_i$ is the intrinsic rate of increase, $B_i$(t)is the biomass of population *i* at time *t*, and $K_i$ is the carrying capacity of population *i*. Natural death in the population is modeled as $x_iB_i(t)$, where $x_i$ is the mass specific metabolic rate of species *i*.  The consumption of species *i* by species *j* is modeled as

$$\sum_{j=1}^{n}\left(x_iy_{ij}F_{ij}(B)B_i(t)\right) \qquad (eq. 4.3),$$

with $y_{ij}$ as the maximum rate at which species *i* assimilates species *j* per unit metabolic rate of species *i*, and $F_{ij}$ is the functional response, which gives the fraction of the maximum rate of ingestion of species *j* by species *i*. Likewise,

$$\sum_{j=1}^{n}(x_j y_{ji} F_{ji}(B) B_j(t)/e_{ij}) \qquad (eq.4.4),$$

represents the consumption by species *j* of species *i*. The parameter $e_{ij}$ is the energetic efficiency of biomass transfer from species *j* to *i*. Dividing by this parameter converts the biomass assimilated by the consumer into biomass lost by the resource.

There are a number of possible functional response types ranging from the traditional Lotka-Volterra through predator-dependence to ratio-dependence. In the model presented by Williams and Martinez (2005), they suggest two types of functional responses; one based on Holling's Type II and III (Holling 1959a, b), and another based on consumer interference (Beddington 1975, DeAngelis et al. 1975). The Holling functional response ($F_{Hij}$) takes the form,

$$F_{Hij}(B) = \frac{B_j^{1+q}}{\sum_k^n B_k^{1+q} + B_0^{1+q}} \qquad (eq.4.5).$$

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). Values of *q* greater than one alter the lag time in the Type III response (Figure 1), effectively increasing the size of the prey refuge.

Alternatively the functional response with consumer interference ($F_{BDij}$) follows

$$F_{BDij}(B) = \frac{B_j}{\sum_{k=1}^{n} B_k(t) + (1 + c_{ij} B_i(t) B_{0ji})} \qquad (eq.4.6),$$

where $c_{ij}$ is the strength of predator interference (Skalski and Gilliam 2001). As with $F_{Hij}$ when *c* = 0 the functional response is a standard Holling Type II. The strength of interference has been estimated in nature to be approximately 1 (Skalski and Gilliam 2001, Williams and Martinez

2005). Increasing values of $c_{ij}$ leads to reductions in the fraction of the maximum ingestion rate of prey biomass (Figure 4.1).

*Implementation*

The model is implemented in the code as a numerical integration using the *deSolve* R package (Soetaert et al. 2010). 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. This function can be broken down into four main parts: growth rates, parameter collecting, simulation, and visualization.

The first part creates a vector of whether or not species are basal using the *getR* function to assess the column sums of the adjacency matrix. If there are no basal species, the function will return a warning ("No basal species in simulation"). This action only occurs when the user elects not to supply a vector of growth rates.

The second part gathers all required parameters of the consumer resource model into a single list. The default values for these parameters are given in Table 2. By default, the function for growth is *Gi* (an internal package function describing growth according to the bioenergetic model above), functional response is *Fij* (the Holling-based functional response), extinction events is *goExtinct* (checks biomasses to see if they are below an extinction threshold), and the default method is *CRmod* (function describing the full bioenergetic model). All species initially start with a random total biomass drawn from a uniform distribution between 0.5 and 1.

The third part of the *CRsimulator* function is 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. By default the extinction threshold is $10^{-10}$, although this is arbitrary and based on Williams and Martinez (2005). Alternative extinction thresholds may also be implemented based on species' characteristics.

There are two functions for the functional responses, one for each type. 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*.  The functions *Fij* (Holling) and *Fbd* (consumer interference) 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 the argument *plot = TRUE*. This is just a very simple way of visualizing the biomass dynamics through time. Each species is represented as a line on the plot, where the x-axis is time and the y-axis is biomass.

### Food Web Dynamics Visualization

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.

### 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 nine indices (defined in Table 3) are computed for each matrix. The output is the number of species, number of links, link density, connectance, diameter, average path length, clustering coefficient, modularity, and number of modules.

Change in three-species motif structure through time can be found by 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* (Csárdi and Nepusz 2006) 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 (Kones et al. 2009).

**Examples**

Here 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 produced by the two model structures, Holling and consumer interference. The second example is primarily used to highlight the analytical functions used to assess changes to topology, motifs, and trophic positions. Thus it offers 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 (Williams and Martinez 2000).

**Two-Species Dynamics**

Two species, a basal resource and a consumer, provide a simple example for this package. The adjacency matrix is two rows and two columns. 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 of species 1 (the basal species), and column three is the biomass of species 2 (the consumer). The resulting biomass dynamics are shown in Figure 4.2. The two species reach equilibrium rather quickly, within 100 time steps, following damped oscillations.

The example in Figure 4.2 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 Holling or based on consumer intereference, one 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 4.3. Looking at the difference between Holling (*Fij*) and consumer interference (*Fbd*) it appears that 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 with a Holling functional response are no longer oscillating.

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. The results of the two species simulation with *xpar* equal to 0, 0.2, 1, and 5 for each functional response type (all else set to default) are shown in Figure 4.4. By altering the tuning parameter (*xpar* in *CRsimulator*) the dynamics associated with each functional response type change.

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.

**Dynamics on a Niche Model Food Web**

Most models of food webs generate a binary adjacency matrix *A* (Figure 4.5) where $a_{ij}$ = 1 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* (proportion of realized links; Links/(S*(S-1))).

Figure 4.6 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 highly oscillatory behavior the system reaches equilibrium with 8 of the initial 20 species in the food web having gone extinct (Figure 4.7, *N*). All measured food web indices exhibit changes through time during the simulation (Figure 4.7), 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 step 50, suggesting that the extinctions in that time step 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 because the reduction in the number of species is large compared to the number of interactions lost. 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 4.8). The clustering coefficient (*CC*) and modularity (*M*) 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.

As would be expected with the decline in both *D* and *APL*, the individual trophic position of each species declines through time (Figure 4.8). Species starting at higher trophic positions (close to 6) declined more than mid-level consumers. Likewise, there were sharp declines in three-species configurations as a result of species extinctions (Figure 4.9). Of course, there is relatively little information that can be obtained from simply describing changes. To move from description to inference, the changes in trophic position and three-species configurations must be compared to a null model. Ideally the changes in trophic position and/or three-species configurations could be compared to sequences of random extinctions (i.e., irrespective of dynamics) although a suitable alternative would be to compare the network at each time step to a niche model web with the same *S* and *C*.

### Food Web Dynamics Explorer

Accompanying this package is a web-based application (https://jjborrelli.shinyapps.io/FW_Dynamics/) that allows users to simulate biomass dynamics on niche model and random networks using the *rend* package. The benefit of this application is that people can explore the predictions of the bioenergetic model with absolutely no coding experience. This would be especially useful as a teaching tool for students to use in order to explore and learn how the dynamics of many species can be different from those of just two species.

Users can manipulate the initial properties of the food web by first selecting either niche or random models. For each of those models food webs can have as few as 5 and as many as 30 species, and connectance values can range from 0.05 to 0.3, which is the range found for real food webs. The number of basal species can also be set between 1 and 5. The simulation parameters (inputs to the *CRsimulator* function under the hood) that can be changed are the number of time steps (50 to 300), the functional response type (Holling or consumer-interference), and the tuning parameter (*q* or *c* respectively).

The Food Web Dynamics Explorer application consists of five web pages. The first page (“The Model”) gives information about the underlying model. The same information can be found above in the *Consumer-Resource Model* section. On this page the model is outlined and the parameters and their default values are described, along with a description of what the user can manipulate. The second page (“Food Web”) displays the structure of the initial food web as both a network and an adjacency matrix. On the third page (“Simulation”) the output of the simulation is presented as a plot of biomass dynamics through time. The last two pages display output analysis. The “Web Structure” page displays a figure similar to Figure 4.7, with the changes in food web indices through time. There is also a table listing descriptions of what each index means. The “Subgraphs” page gives the same information as Figure 4.9, the change in frequency of three-species subgraphs over time.

## Conclusions

The *rend* package allows for the simulation of biomass dynamics on any potential food web structure. This ability will allow for further enumeration of the relationship between ecological network structure and dynamics. At this point the package grants the ability to measure the impact of predator-prey dynamics on food web structure, without allowing structure to change beyond species extinction. Nonetheless it would be relatively straightforward to use *rend* in conjunction with additional code for the generation of new and additional network structure. From there, the two-way relationship could be explored further.

Moreover, that *rend* is easily usable with a small amount of skill in R makes it a valuable tool. Anyone who is interested may download the package from GitHub, where it is currently being hosted. Simply type *devtools::install_github(“jjborrelli/rend”)* into the R console and it is ready to go.

## Future Directions

There are a number of different directions further development of this package could take in terms of increasing performance and functionality. Below I highlight several directions I plan to take in further developing the *rend* package.

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 through the implementation of null models. One null model would be to assume species extinctions are 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.

Currently this package only supports a narrow range of model structures. Users can simulate dynamics with a Holling Type II, II.2, or III function response (Holling 1959a, b, Williams and Martinez 2004, Romanuk et al. 2009), or they can use the consumer-interference functional response (Beddington 1975, DeAngelis et al. 1975, Skalski and Gilliam 2001). This is only a small subset of the proposed functional response types and does not include both simpler and more complex forms.

Drossel et al. (2004) demonstrated that ratio-dependent functional responses lead 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, this type of functional response should be included. However, currently there is no implementation of a ratio-dependent functional response for bioenergetic models. There has, however, been work in this direction for 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.

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. An alternative will be to use the *sde* package (Iacus 2015), 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 predictions made from stochastic models would generate more realistic outcomes.

The *rend* package was initially imagined for ecological network dynamics, which extends beyond food webs and predator-prey interactions (Ings et al. 2009, Kéfi et al. 2012). The umbrella of ecological networks includes other interaction types, and mutualistic or competitive networks are becoming increasingly popular areas of research (Medan et al. 2007, Bascompte 2009, Johnson and Amarasekare 2013, Rohr et al. 2014). 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 (Kéfi et al. 2012).

**Figure 4.1:** How resource biomass consumed changes with the tuning parameter (*q* or *c*) for both the Holling and consumer-interference functional responses (all other parameters set to default; *states = NULL, t = 1:200, G = Gi, method = CRmod, K = 1, x.i = 0.5, yij = 6, eij = 1, B.o = 0.5, r = NULL, ext = goExtinct).*

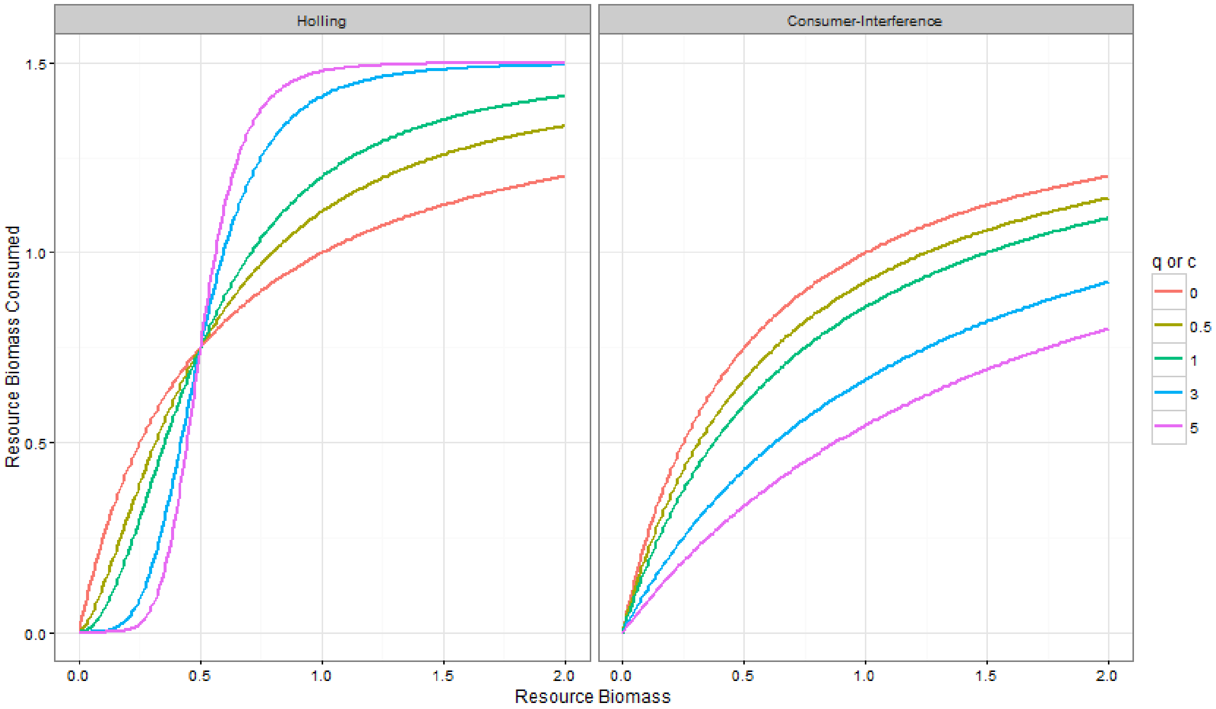


**Figure 4.2:** Biomass dynamics of a two-species predator prey system given default settings for the *CRsimulator* function (*states = NULL, t = 1:200, G = Gi, method = CRmod, FuncRes = Fij, K = 1, x.i = 0.5, yij = 6, eij = 1, xpar = 0.2, B.o = 0.5, r = NULL, ext = goExtinct*) displayed both in state space (Consumer vs. Resource Biomass) and as a trajectory through time.

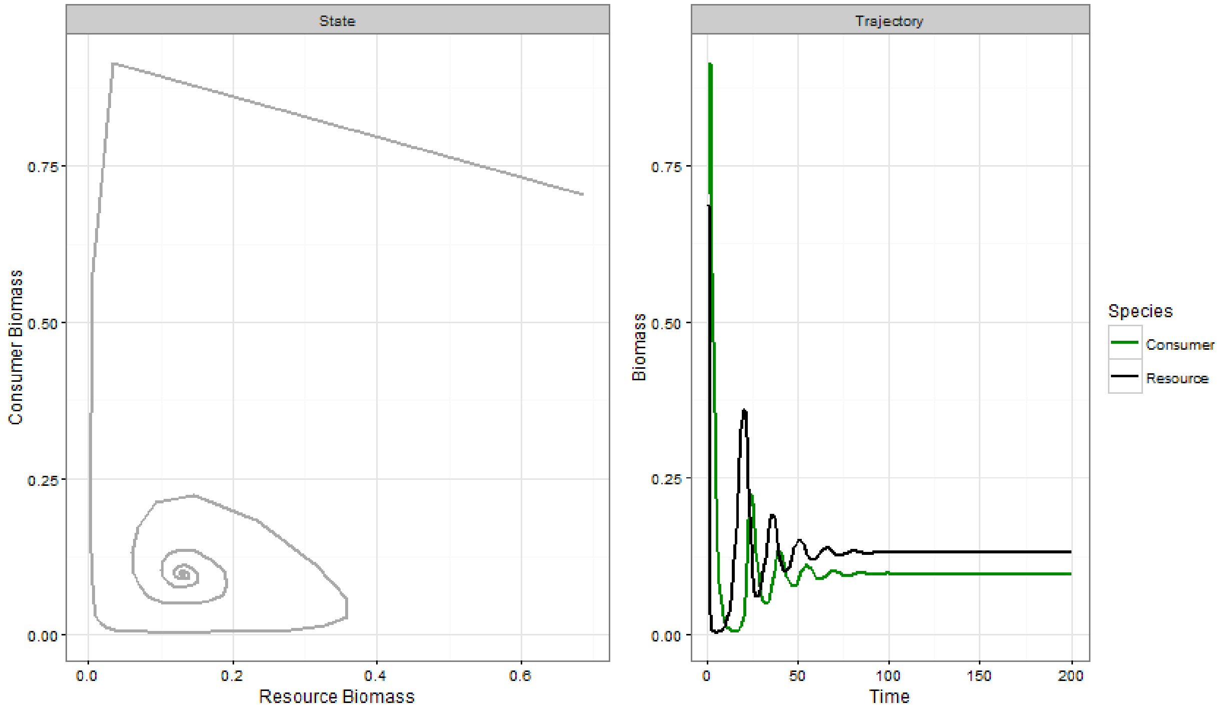


**Figure 4.3:** Differences in dynamics of a two-species predator-prey system as a result of alternative functional responses (Holling, *Fij*; Consumer Interference, *Fbd*) displayed as Resource and Consumer trajectories. Simulation uses default parameters except for functional response (*states = NULL, t = 1:200, G = Gi, method = CRmod, K = 1, x.i = 0.5, yij = 6, eij = 1, xpar = 0.2, B.o = 0.5, r = NULL, ext = goExtinct*).

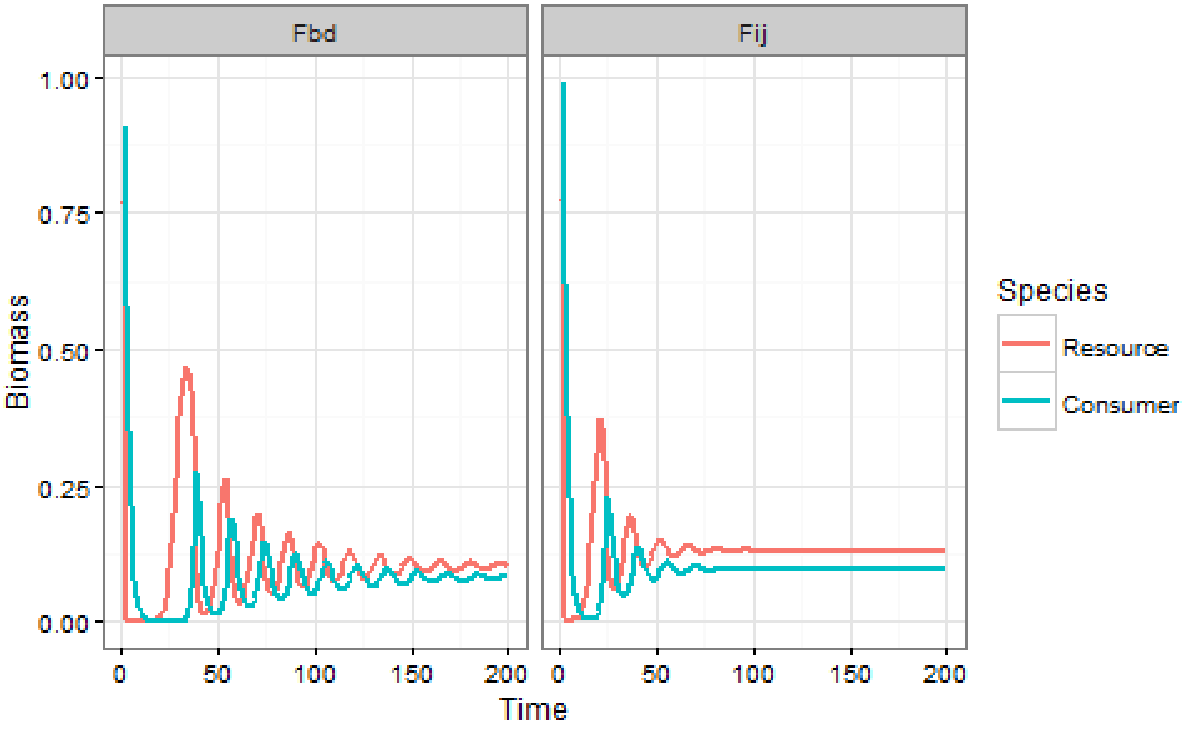


**Figure 4.4:** How varying the tuning parameter affects the dynamics of a two-species predator-prey system with alternative functional responses. Simulation uses default parameters except for functional response (*states = NULL, t = 1:200, G = Gi, method = CRmod, K = 1, x.i = 0.5, yij = 6, eij = 1, xpar = 0.2, B.o = 0.5, r = NULL, ext = goExtinct*).

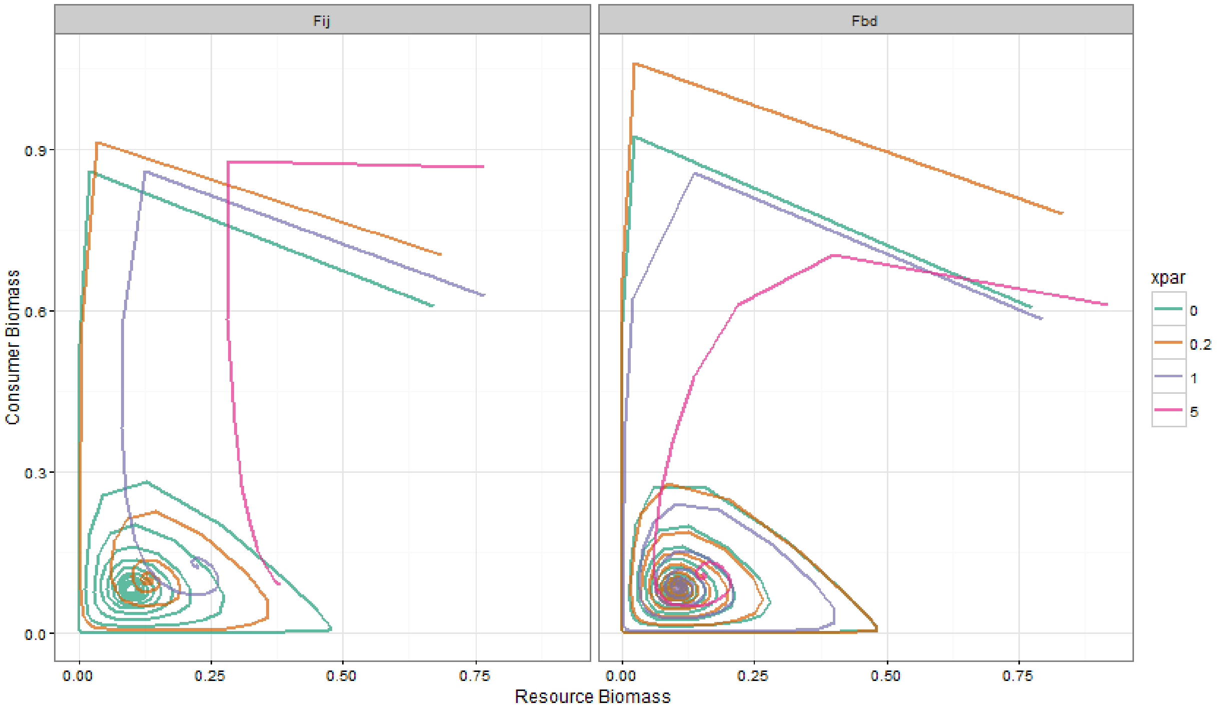


**Figure 4.5:** One example of a species by species adjacency matrix for a simulated niche model food web with 20 species and a connectance of 0.15, constrained to have 5 basal species. A black square indicates that the row species is predated by the column species. This is interpreted as a matrix of 0s (white squares) and 1s (black squares) and then used to define the trophic structure for the bioenergetic model.

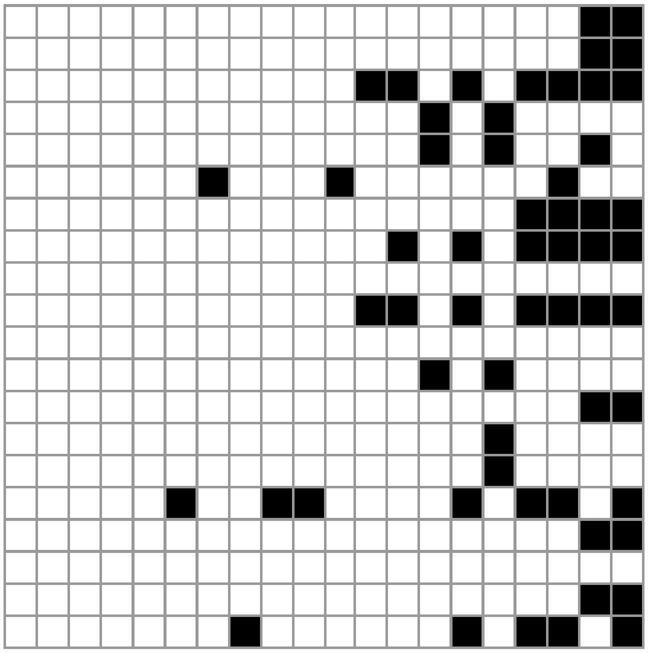


**Figure 4.6:** Simulated biomass dynamics through time of a niche model web with *S* = 20, *C* = 0.15, and 5 basal species. Inset shows an expanded view of the last 25 time steps. Simulation parameters are: *states = NULL, t = 1:200, G = Gi, method = CRmod, FuncRes = Fij, K = 1, x.i = 0.5, yij = 6, eij = 1, xpar = 0.2, B.o = 0.5, r = NULL, ext = goExtinct.*

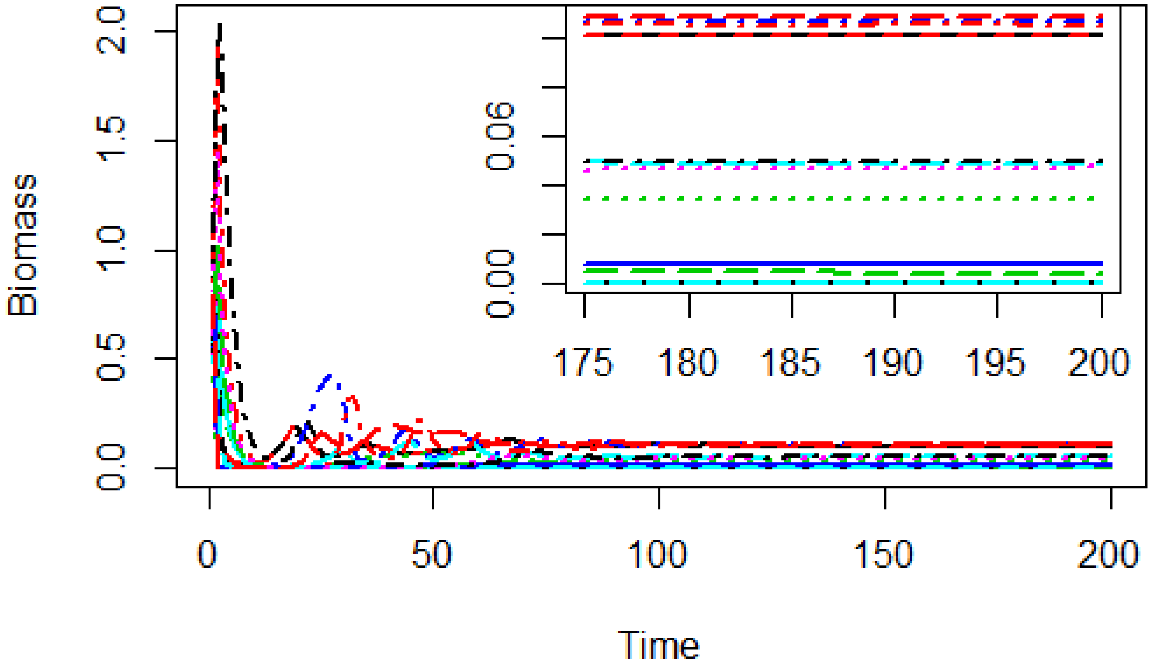


**Figure 4.7:** Plot of changes in a set of nine food web indices as a result of biomass dynamics on a niche model food web with *S* = 20, *C* = 0.15, and 5 basal species; number of species (*N*), number of links (*Ltot*), link density (*LD*), connectance (*C*), diameter (*D*), average path length (*APL*), clustering coefficient (*CC*), modularity (*M*), and number of modules (*nMod*). Simulation parameters are: *states = NULL, t = 1:200, G = Gi, method = CRmod, FuncRes = Fij, K = 1, x.i = 0.5, yij = 6, eij = 1, xpar = 0.2, B.o = 0.5, r = NULL, ext = goExtinct*.

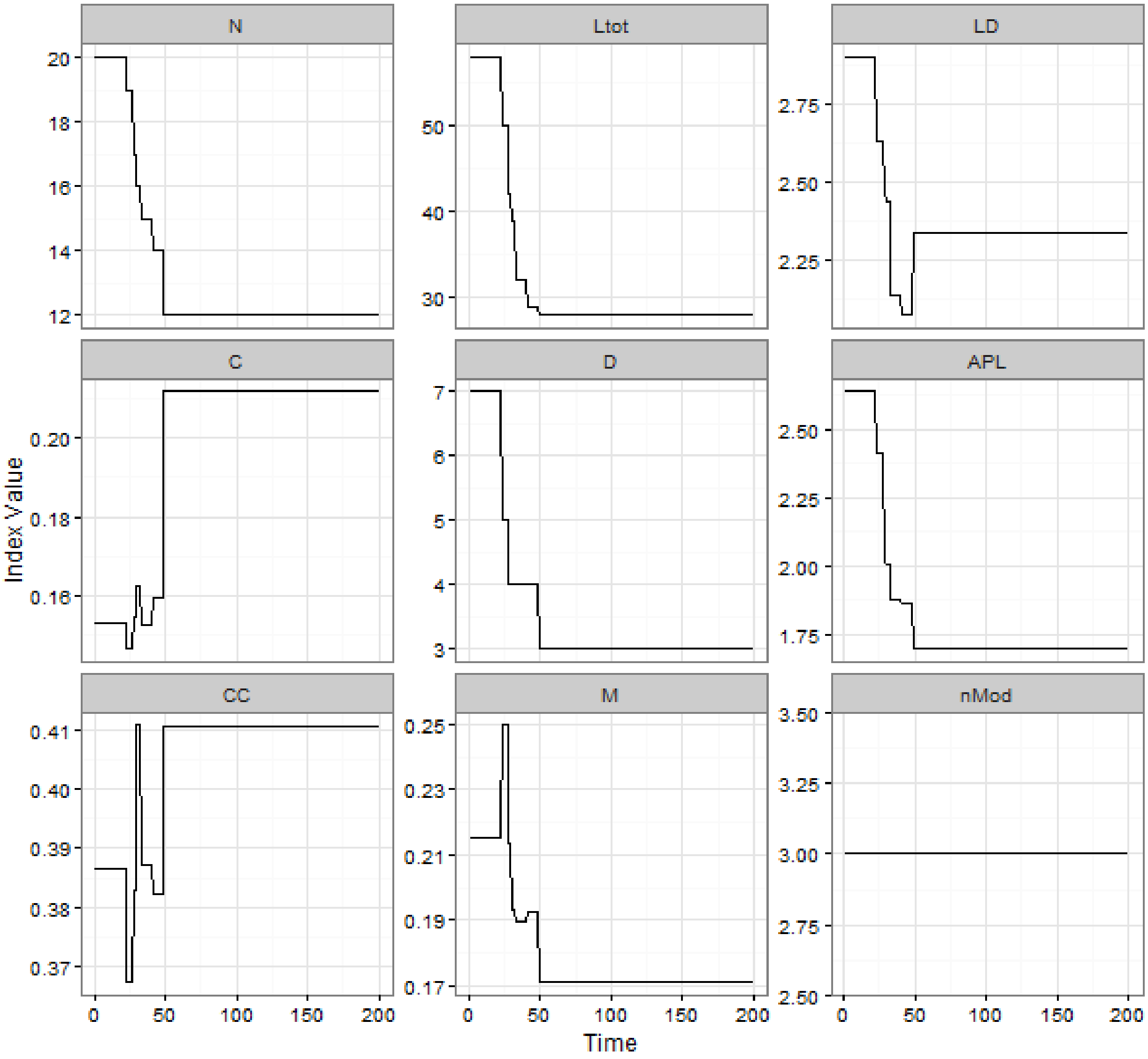


**Figure 4.8:** Plot of changes in the trophic position of the species (each panel representing a single species) in a niche model food web with $S$ = 20, $C$ = 0.15, and 5 basal species as a result of biomass dynamics. Simulation parameters are: *states = NULL, t = 1:200, G = Gi, method = CRmod, FuncRes = Fij, K = 1, x.i = 0.5, yij = 6, eij = 1, xpar = 0.2, B.o = 0.5, r = NULL, ext = goExtinct*.

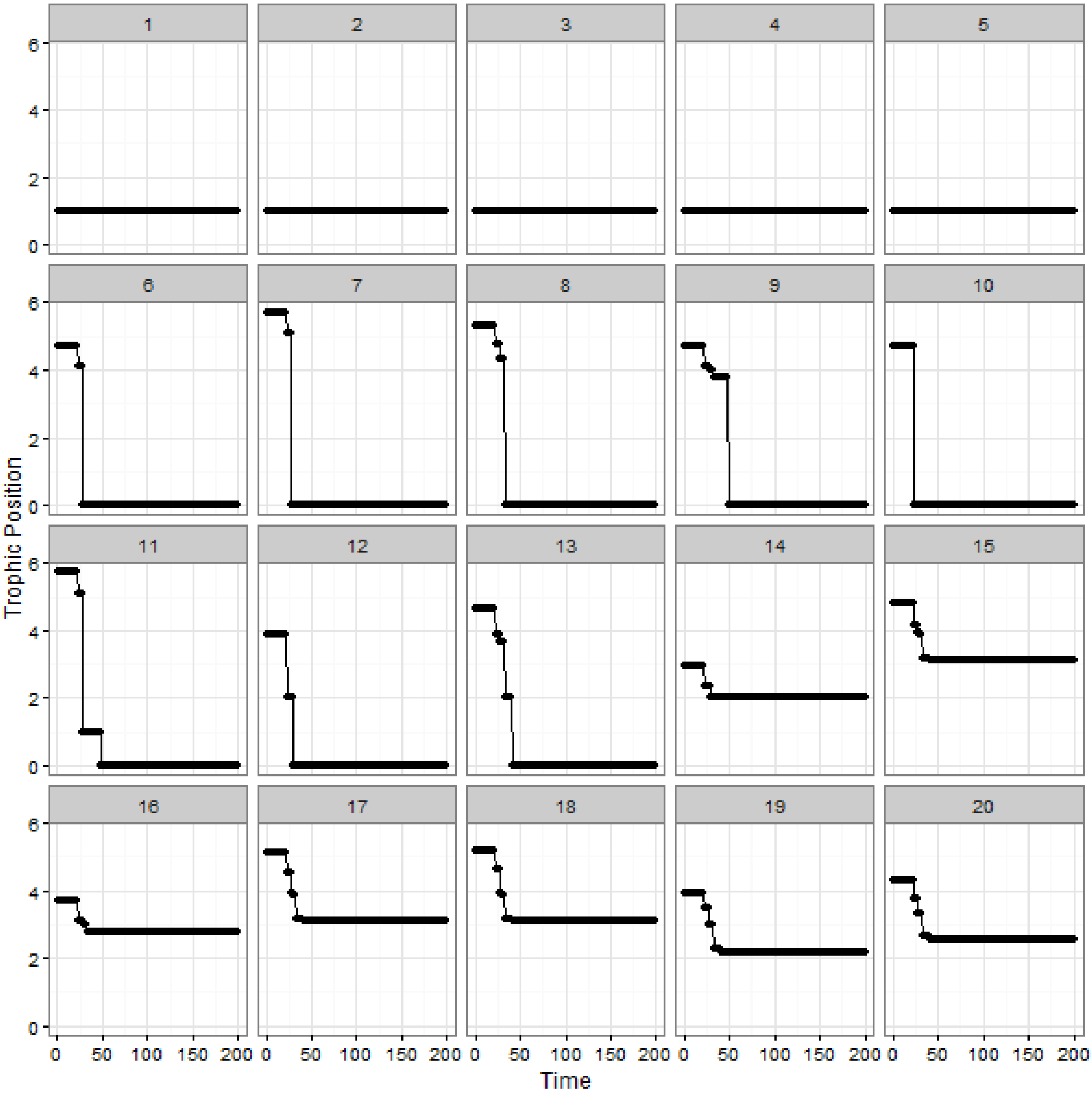


**Figure 4.9:** Plot of changes in the number of each of the thirteen three-species trophic configurations (each panel representing a different three-species configuration) in a niche model food web with *S* = 20, *C* = 0.15, and 5 basal species as a result of biomass dynamics. Simulation parameters are: *states = NULL, t = 1:200, G = Gi, method = CRmod, FuncRes = Fij, K = 1, x.i = 0.5, yij = 6, eij = 1, xpar = 0.2, B.o = 0.5, r = NULL, ext = goExtinct*.

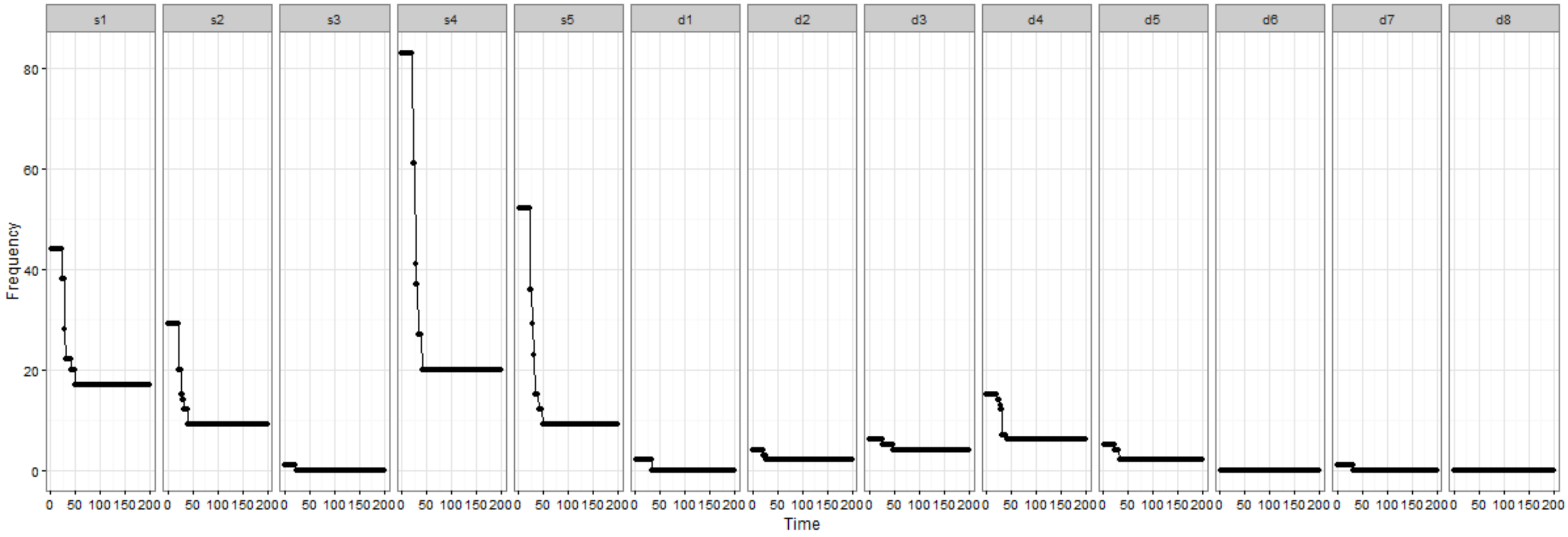


**Table 4.1:** Definitions of the parameters of the *CRsimulator* function

| Parameter | Definition |
|---|---|
| Adj | Adjacency matrix |
| states | Biomass values for the species |
| 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 |
| r | Vector of growth rates (1 for basal, 0 otherwise) |
| ext | Function describing extinction events during the simulation |
| plot | Whether or not to generate a plot of biomass against time |

**Table 4.2:** Default parameter values input to the *CRsimulator* function, adopted from Williams and Martinez (2005).

| Parameter | Value |
|---|---:|
| K | 1 |
| x.i | 0.5 |
| yij | 6 |
| eij | 1 |
| xpar | 0.2 |
| B.o | 0.5 |

**Table 4.3:** Description of the nine food web indices measured by the *WEBind* function; number of species (*N*), number of links (*Ltot*), link density (*LD*), connectance (*C*), diameter (*D*), average path length (*APL*), clustering coefficient (*CC*), modularity (*M*), and number of modules (*nMod*).

| **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 |

**Table 4.4:** Sample output, the first six rows, of the *CRsimulator* function.

| time | 1 | 2 |
|---:|---:|---:|
| 1 | 0.7754 | 0.733 |
| 2 | 0.0304 | 0.988 |
| 3 | 0.0053 | 0.6214 |
| 4 | 0.0029 | 0.3801 |
| 5 | 0.0025 | 0.2318 |
| 6 | 0.0028 | 0.1414 |

# Chapter 5: Influence of topology on the dynamic disassembly and assembly of food webs

## Abstract

A longstanding question in community ecology is what makes for a stable community. The study of food webs has sought to answer this question by assessing how different food web architectures impact community stability. Food web structure not only impacts the dynamics of the community, the population dynamics in the community, in turn, alter food web structure. Structure may change as a result of altered interactions through prey switching or through the local extinction/invasion of participating species. Previous work has suggested that some food web structures (e.g., short trophic chains, and motifs) are so commonly observed because they are more likely to be stable and thus less likely to be lost via extinction of participating species, or be preferentially retained during community assembly, but to date, this hypothesis has not been tested. I assessed how species deletions and introductions alter food web structure and determined how those changes affected either the persistence of the web or the success of the invasion using principle components regression. I found that the model structure and parameterization caused substantial variation in the results. The properties of individual species had significant effects on the outcome, and a change in the motif profile had significant impacts for both persistence following deletion as well as for invasion success.

## Introduction

Ecologists have been interested in the link between the complex structure of natural communities and their dynamics for over sixty years, beginning in the 50's when ecologists such as McArthur (1955), Elton (1958), and Paine (Paine 1966, 1969) hypothesized that complex communities should be more stable as a result of redundancies in the system. This simple observation was turned on its head when May (1972) demonstrated that there was, mathematically, no reason to suspect a link between increased complexity and stability. In fact, the opposite pattern was true in *in silico* model ecosystems constructed at random (May 1972, 1973), and not constrained to have a feasible equilibrium (Roberts 1974, Christianou and Kokkoris 2008). May suggested that ecologists should seek to "elucidate the devious strategies which make for stability in enduring natural systems" (May 1973, p 72), launching forty years of research on ecological networks (e.g., Cohen and Newman 1985, 1988, Lockwood et al. 1997, Dunne et al. 2002, Guimarães et al. 2007, Vermaat et al. 2009, Bascompte 2009, Eklöf et al. 2013).

Research into the link between structure and stability has taken three major paths: description of the structure of empirical networks (e.g., Cohen and Newman 1985, Dunne et al. 2002, Vermaat et al. 2009), analyses of random matrices (May 1972, Pimm and Lawton 1977, Allesina and Pascual 2008, Allesina and Tang 2012, Tang and Allesina 2014), and analysis of the behavior of dynamical models of species interactions (McCann et al. 1998, 2005, Drossel et al. 2004, Williams and Martinez 2005, Romanuk et al. 2009, Petchey et al. 2010). The majority of research has historically been focused on food webs, networks of feeding interactions. Recently, however, there has been increased interest in other interaction types such as mutualisms (Olesen et al. 2008, Alarcón et al. 2008, Albrecht et al. 2010, Rohr et al. 2014), parasitism (Lafferty et al. 2006, 2008, Dunne et al. 2013), competition (Allesina and Levine 2011, Allesina and Tang 2012), and mixed interactions (Proulx et al. 2005, Ings et al. 2009, Kéfi et al. 2012).

Early studies of food webs focused on their topological structure, especially the relationship between the number of species and the number of links in the web. The first collections of food web datasets were used to demonstrate scale-invariance, that the average number of links per species remains constant with increasing numbers of species (Briand 1983, Briand and Cohen 1984). Following the collection of food webs with higher resolution and more taxa (e.g., Martinez 1991, Polis 1991), the notion of scale-invariance was scrapped. Many other food web properties have been explored, including predator-prey ratios , proportion of top, intermediate, and basal species as well as proportions of links between each group, intervality in consumer diets, and food chain length (Post 2002, Dunne 2005, Stouffer et al. 2006).

These studies of food web structure guided the development of phenomenological models like the cascade (Cohen and Newman 1985), niche (Williams and Martinez 2000), and nested-hierarchy (Cattin et al. 2004) models. Each of these models takes as input the number of species and the connectance (proportion of realized links) and uses a set of heuristic rules to govern the placement of links within a modeled food web. Typically species are arranged along some hypothetical niche axis, and then prey are chosen (with rules dependent on the model) from species whose niche value is (typically) lower. Stouffer et al. (2005) proved that there were two key components required to predict food web properties using these models. First, the niche values of the species must form an ordered set, and second species have a probability of preying on species with a lower niche value (drawn from an exponential distribution).

Random matrices link the equilibrium dynamics of ecological communities to their structure, and research in this area has expanded on the observations of May (1972). The matrix of interest is the Jacobian, or community matrix, which describes how each species impacts the growth of the other species. Finding the eigenvalues of this matrix allows one to determine how the community will respond to a small perturbation away from equilibrium. If all eigenvalues are negative, the system will return to equilibrium. The magnitude and signs of the values in the Jacobian matrix are defined by both the equations governing the dynamics of the system and by the structure of the interactions among species. The behavior of the community around the equilibrium can then be described by randomly filling in the Jacobian matrix and finding the distribution of eigenvalues. Recent research has shown that increasing the proportion of predator-prey interactions (relative to other interaction types) can help to stabilize communities (Allesina and Pascual 2008, Allesina and Tang 2012), and that correlations between the strengths of reciprocal interactions may be more important to stability than the structure of the network (Tang et al. 2014). However, the use of random matrices may lead to underestimation of the stability of real communities (James et al. 2015).

Where random matrices can be used to explore the link between structure and dynamics around equilibrium, the application of dynamical models to ecological communities allows for an understanding under non-equilibrium conditions. Using this approach ecologists can determine the impact of the type of dynamics (Drossel et al. 2004) and model parameterization (McCann et al. 1998) on food web stability. Additionally, models of population dynamics can be paired with models of community assembly (Pawar 2009), metacommunity dynamics (Baiser et al. 2012), and evolution (Loeuille and Loreau 2005). In simulations of assembly, food webs grow and change with introductions of new species and loss through extinction and/or migration. Over time these simulated food webs reach an equilibrium number of species and their final structure can be compared to empirical food webs.  Different dynamical model equations and parameterizations may yield equilibrium webs that differ from what is observed in nature (Drossel et al. 2001, 2004, Drossel and Mckane 2003, Allhoff and Drossel 2013).

The hypothesis of selection on stability suggests that the dynamics of interacting populations creates a selective process that leads to observed food web structure via the preferential loss of unstable configurations (Borrelli and Ginzburg 2014, Borrelli 2015, Borrelli et al. 2015). As food web structure changes via the introduction of new species through migration and invasion, or through loss of species from local extinction, new food web structures either will or will not persist through time.

Two patterns found in real food webs have been hypothesized to have been produced by selection on stability: short food chains (Borrelli and Ginzburg 2014) and three-species motif profiles (Borrelli 2015). These patterns represent a signature of the selection process because they are structures that both have a high probability of being stable and a high frequency in observed food webs. Much stronger evidence that these patterns are generated via selection on stability of food webs would come from dynamical models applied to food webs. If this selective process is occurring, the observed patterns should emerge from the dynamics of food webs whose structure is generated from a reasonable mechanism.

I applied a bioenergetic model of multispecies predator-prey dynamics to food webs generated from disassembly (species removal) and assembly (species introduction) processes. Patterns of short chains and three species motifs could result from the preferential establishment of species contributing to them, or the preferential loss of species that do not contribute to these patterns.  I looked at the relative importance of the loss of species from destabilizing links, the loss of species that contribute more towards short food chains and three-species configurations, or the addition of species that are more likely to create stable links, in overall food web stability.

## Methods

### Population Dynamics Model

I took advantage of a well-studied model of multispecies predator-prey dynamics (McCann et al. 1998, Williams and Martinez 2005, Romanuk et al. 2009). This model was originally proposed by Yodzis and Innes (1992) and later extended to more than two species (McCann et al. 1998). The impacts of varying the model structure and parameterization on food web persistence has also been explored (Williams and Martinez 2005). Williams and Martinez (2005) and Romanuk (Romanuk et al. 2009) also suggest a simplified version of this model to reduce the size of parameter space that needs to be explored. It is this simplified version that I used to investigate food web disassembly and assembly. The model takes the form,

$$\frac{dB_i(t)}{dt} = G_i(B) - x_i B_i(t) + \sum_{j=1}^{n} (x_i y_{ij} F_{ij}(B) B_i(t) - x_j y_{ji} F_{ji}(B) B_j(t)/e_{ij}) \qquad (eq. 5.1)$$

Here basal species increase in biomass according to $G_i$, which is density-dependent growth expressed as $r_iB_i(1\text{-}(B_i/K_i))$. For non-basal species, $r_i$ is equal to 0. All species in the model die at a rate equal to $x_i$, which represents the metabolic rate of species *i*. Consumers increase in total biomass with consumption, and intermediate consumers lose total biomass from being consumed. In this model consumption can be modeled using one of two functional responses: one based on Holling's Type II and III (Holling 1959a, b), and another based on consumer interference (Beddington 1975, Skalski and Gilliam 2001). The Holling functional response, $F_{Hij}$,

$$F_{Hij}(B) = \frac{B_j^{1+q}}{\sum_k B_k^{1+q} + B_0^{1+q}} \qquad (eq. 5.2),$$

can be tuned to represent either the standard Type II ($q$ = 0) or Type III ($q$ = 1), with other values of $q$ > 0 being varying strength of the Type III response (i.e., larger values of $q$ represent larger prey refugia). Alternatively, the Beddington-DeAngelis functional response, $F_{BDij}$,

$$F_{BDij}(B) = \frac{B_j}{\sum_{k=1}^{n} B_k(t) + (1 + c_{ij}B_i(t)B_{0ji})} \qquad (eq\ 5.3),$$

takes the shape of a Holling Type II functional response when $c$ = 0, but the strength of intra-specific consumer interference can be increased by setting $c$ > 0. Additional constraints on consumption come from the the maximum ingestion rate ($y_{ij}$), the half-saturation constant ($B_0$), and energetic efficiency ($e_{ij}$). The energetic efficiency parameter here is used to convert the biomass lost by species $j$ into biomass of species $i$.

This bioenergetic model offers several benefits over alternative dynamical model structures. For example, many studies of food web assembly and dynamics have used a multi-species Lotka-Volterra model (e.g., Baiser et al. 2010). Drossel et al. (2001, 2004) showed that population dynamics models with non-linear functional responses allow for the generation of more realistic food webs via evolution-based models of assembly than models with linear functional responses (e.g., Holling Type I). The preference of predator $j$ for prey $i$ can also be incorporated into a bioenergetic type model relatively easily as well, although for this study I assumed that predators had no prey preferences.

**Simulation**

A 1000 species regional food web with connectance of 0.15 was generated using the niche model (Williams and Martinez 2000). I sampled 200 local food webs by randomly selecting 50 species from the regional food web. Local webs were constrained to have 5 basal species and no unconnected species, meaning that all consumers were required to have at least one resource and all basal species had at least one consumer. Population dynamics of the species in each local web were simulated for 1000 time steps using the bioenergetic model described above. Dynamics were simulated with either a Holling functional response ($F_{Hij}$) or one with consumer interference ($F_{BDij}$). For each model structure, three values of the tuning parameter ($q$ and $c$ respectively) were applied: 0, 0.2, and 1. Additional model parameters were kept constant (Table 5.1) following Williams and Martinez (2005) and Romanuk (2009). This gave me six scenarios, one for each combination of model structure and tuning parameter. The number of resulting equilibrium webs ("equilibrium webs" refers to the local web at the end of the 1000 time steps) that had no unconnected species were 19, 133, 94, 30, 53, and 76 respectively. Food web disassembly and assembly processes were applied to each of these resultant communities. The code used for each part of this simulation is provided in Appendix 4.

*Disassembly*

Disassembly was simulated by removing each species from the initial equilibrium food web one at a time. Following the deletion, the new web was compared to the full web to determine the change in a set of 13 food web properties (described in Table 5.2) and 13 three-species subgraphs. In addition to the change in food web properties, I measured the generality, vulnerability, trophic position, and omnivory index of the deleted species. The dynamics of the remaining species were simulated using the bioenegetic model described above with the same functional response and tuning parameter combination that was used for the initial simulation that generated equilibrium webs from local webs. Initial abundances for each species were set to their equilibrium abundance found at the end of the initial 1000 time steps. Simulations then proceeded for 500 time steps, after which I determined the persistence of the web (the number of species whose biomass was greater than 0). All species whose biomass fell below $10^{-10}$ were considered extinct and their biomass was set to 0.

*Assembly*

A new species was randomly chosen from the regional food web (the original 1000 species niche model food web) to invade the equilibrium food webs. The newly introduced species was required to have at least one resource existing in the community, unless it was a basal species, in which case it had to have at least one consumer in the web. This ensured that all introduced species had a non-zero chance of becoming established. Following the selection of a species and introducing it into the web, I compared the invaded web to the pre-invasion initial equilibrium web to determine how the introduced species altered a set of 13 food web properties (Table 5.2) and the frequency of 13 three-species configurations. For each initial food web, 300 different species introductions were made.

The bioenergetics model was then applied to the invaded web, with the same structure and tuning parameter combinations as were used to generate the initial web. The simulated dynamics were used to determine if the introduced species survived for all 500 time steps (establishment). If it did not survive for 500 time steps, I recorded how long the introduced species persisted (time to extinction). Additionally, I determined how many secondary extinctions occurred as a result of the introduction as the change in species from the initial invaded to the final web. I also recorded the introduced species' trophic specialization, vulnerability, trophic position, and omnivory index.

## Analysis

To determine how deleted or introduced species impacted the structure of the food web, I computed the change in a set of food web properties; number of species (N), number of links (L), connectance (C), mean trophic generality (meanGen), mean vulnerability (meanVul), standard deviation of generality (sdGen), standard deviation of vulnerability (sdVul), mean trophic position (meanTP), standard deviation of trophic position (sdTP), mean omnivory index (meanOI, where omnivory index is the standard deviation of the trophic position of all prey for the target species), standard deviation of omnivory index (sdOI), average path length (APL), and diameter (D).  In addition to the food web properties, I computed the change in the substructure of the web as the change in frequencies of three-species configurations.

To get a general overview of how species' roles affected food web disassembly, I split the data into two groups: deletions that led to no change in persistence (no secondary extinctions), and deletions that reduced persistence. For these two groups I compared the mean change in food web properties, mean change in three-species modules, and the mean value for deleted species characteristics.

Species introductions in food web assembly could result in one of four outcomes. The introduced species could either become established in the web or go extinct, and it could either cause other species to go extinct or not. When the introduced species was able to establish in the food web and cause no extinctions the food web grew. If the introduced species established and caused extinctions, the food web either shrank or stayed the same (depending on the number of extinctions). Alternatively if the introduced species went extinct, it could either cause additional extinctions while it persisted, or it could go extinct without causing additional extinctions. To determine if the change in web properties or substructure affected the result of species introduction, I compared the mean change in these properties across all four possible scenarios. I also compared the mean characteristics of successful and unsuccessful invaders.

Many of the food web properties I measured are correlated. In order to compensate for the inflated variance caused by multicollinearity, I used principle components regression with linear mixed effects models to determine the effect of species' roles on the dynamics of the system (persistence in the case of disassembly and invasion success for assembly). Principle components were separately determined for the 13 food web properties and for the 13 three-species configuration frequencies.

I tried to predict the results of species addition or subtraction for each of the six combinations of functional response and tuning parameters.  Three separate sets of predictor variables were used, including changes to whole web properties, changes to subgraph structure, or properties of the deleted/invaded species. For each set of predictor variables, a principle components analysis was used to determine the linear combination of weighted predictor variables that explained the largest amount of the variance in the predictors. The first principle component of web properties ($PC_{web}$), subgraph structure ($PC_{sub}$), and species properties of removoved or added species ($PC_{sp}$) was then used to predict either persistence of remaining species, in the case of species removals, or invasion success, in the case of species additions. The identity of the local web was included in each model as a random effect. This resulted in a total of 36 models (3 subsets of predictors x 2 model structures x 3 parameterizations x 2 simulated scenarios).

## Results

### *Disassembly*

Most webs were persistent following deletions that had large effects on most food web properties.  There were few consistent differences among the relatively few webs where secondary extinctions occurred and webs that were unaffected by species deletions. However, secondary extinctions were more common when deletions altered food web structure in specific ways. In particular, when there was a large change in the frequency of tritrophic chains following the initial removal of a species, the resulting webs were not persistent (Figure 5.1). In scenarios with weak consumer interference, persistent webs were those that lost more modules with double links (Figure 3.1) after species deletion. Under conditions of strong consumer interference, webs that were persistent lost more competition (apparent and exploitative) modules.  The attributes of the deleted species were consistently different between persistent and non-persistent food webs (Figure 5.2). Specifically, secondary extinctions followed the deletion of species that had more predators (higher vulnerability) and occupied lower trophic levels. Non-persistent webs also resulted from the deletion of species that had more prey (higher generality) and less omnivory, although these results were more variable. There was also an apparently large effect of functional response and parameterization (Figures 5.1-3), as the effect of changes in each of the measured properties varied across scenarios.

$PC_{web\_del}$ varied among the six different bioenergetic model combinations, and explained 80-99% of the variation in these properties. The change in links was present in the first axis for all combinations, and was the sole variable for 3 combinations. Of the remaining combinations, mean generality and vulnerability contributed to the first principle component, and in two of the combinations diameter was included. The loadings on $PC_{sub\_del}$ were more consistent across combinations: with the tritrophic chain, apparent competition, and exploitative competition consistently included. The intraguild predation module was also correlated with the first principle component in four of the six combinations. This principle component explained 48-65% of the variation. $PC_{sp\_del}$ was comprised of the removed species' generality and trophic position, and explained 60-83% of the variation,

$PC_{web\_del}$ did not have a significant effect on persistence in most cases (Table 5.3). The only exception being a statistically significant negative effect of $PC_{web\_del}$ on persistence when consumer interference in the functional response was very weak ($c$ = 0). $PC_{sub\_del}$ had a positive effect on persistence given a weak Type III functional response ($q$ = 0.2) or weak consumer interference ($c$ = 0, $c$ = 0.2). The removed species' properties had the most consistent positive effect on persistence. When both subgraph structure and species properties were found to have effects on persistence ($F_{Hij}$, 0.2; $F_{BDij}$, 0.2), the model based on subgraph structure explained more of the variation. In one of those two cases ($F_{BDij}$, 0.2) the subgraph structure model had the lowest AIC as well.

*Assembly*

Successful invaders added more links to the web than unsuccessful ones in all scenarios, except when consumer interference was weak (Figure 5.4). Additionally, successful invaders increased mean generality and vulnerability, as well as the variability in those properties. When there was a Type II or weak Type III functional response unsuccessful invaders increased mean trophic position more, although in other scenarios the same effect was shown by successful invaders. Unsuccessful invaders always caused a larger increase in the mean and standard deviation of the omnivory index, as well as increased average path length and diameter.

Successful invaders in scenarios with a consumer interference functional response contributed more tritrophic chains, exploitative and apparent competition, and intraguild predation modules (Figure 5.5). Scenarios with a weak Type III or Type II functional response had the opposite pattern for tritrophic chains, but similar patterns for exploitative, apparent competition, and intraguild predation. In some scenarios successful invaders added more modules that include double links (e.g., d2, d6, and d7 with strong consumer interference; Figure 5.5), although in most cases invaders adding more of these modules was unsuccessful.

The characteristics of successful invaders also varied with the functional response and parameterization. One consistent difference was that a successful invasion was accompanied by fewer secondary extinctions than unsuccessful invasions (Figure 5.6). With Type II or weak Type III functional responses invaders also had lower trophic positions when they were successful, although with strong Type III and moderate consumer interference the reverse was true. Successful invaders were also always less omnivorous than unsuccessful ones, although their generality did not show consistent patterns. With either a strong Type III or at least some consumer interference, successful invaders had more prey species and fewer predators.

One difference between successful invaders that cause secondary extinctions and those that do not was that the latter tended to add more links to the web. Invaders that caused extinctions also tended to increase the number of tritrophic, apparent competition, exploitative competition, and intraguild predation modules in the web. In most scenarios invaders that did not cause extinctions tended to increase the average path length, diameter, and mean trophic position.  Every unsuccessful invader caused secondary extinctions.

$PC_{web\_inv}$ was most strongly correlated with the change in the number of links for all scenarios. In the two scenarios with functional responses in which $q = c = 0$, mean generality, vulnerability and the standard deviation in generality were also correlated with the first principle component. Tritrophic chains, intraguild predation, apparent competition, and exploitative competition were all correlated with the first principle component for the change in subgraph structure for all scenarios. This component explained 47-70% of the variation in the predictors. Variation explained by $PC_{sp\_inv}$ ranged from 50-55%. Introduced species' generality and trophic position were correlated with the first principle component across all scenarios. Vulnerability of the introduced species was correlated with the first principle component in four of the six scenarios, and omnivory only in one.

Introduced species' success was consistently significantly related to changes in web structure or characterstics of invading species, with the exception of the model of subgraph structure for the Holling functional response with $c = 0$ (Table 5.3). Among the 18 models predicting success of the introduced species, the lowest AIC values (best fit) were those based on properties of the introduced species ($PC_{sp\_inv}$). Models using $PC_{web\_inv}$ as a predictor had the second lowest AIC values while those using $PC_{sub\_inv}$ had the highest. However, models using $PC_{sub\_inv}$ as a predictor consistently explained a greater proprotion of the variation (higher $r^2$).

The effect of species properties were dependent upon the form of the functional response. In scenarios with a Holling-based functional response, regardless of parameterization, $PC_{sp\_inv}$ had a negative effect on establishment, but in scenarios with consumer interference, the effect was positive. $PC_{sub\_inv}$ typically had a negative effect on establishment and in most scenarios, increases in $PC_{web\_inv}$ led to an increase in the introduced species success. However, with a scenario of a weak Type III functional response ($q = 0.2$) the effects of $PC_{sub\_inv}$ and $PC_{web\_inv}$ were positive and negative respectively.

## Discussion

Based on the idea of selection on stability, the presence of species that contribute to food web configurations believed to be more stable should increase whole food web stability. Thus, their loss from an equilibrium food web should have a larger effect on persistence of remaining species than would the removal of other species. These species likewise should have a higher chance of becoming integrated into the food web as successful invaders and cause fewer secondary extinctions after invasion. The results of the simulations described here were actually far more complex as the impact of food web structure on stability was dependent on both the functional response type and model parameterization.

Model food webs with shorter food chains have a higher probability of being stable (Borrelli and Ginzburg 2014).  As a result, when species are deleted from the higher trophic levels there should be smaller effects on persistence because larger reductions in mean trophic position, average path length, and food web diameter should result in more persistent webs. In this study, the removal of species at higher trophic positions did typically produce webs that were persistent. When a weak type II functional response was used in the simulation, persistent webs were also associated with a larger decrease in mean trophic position, matching expectations. In other scenarios, however, there were no differences between persistent and non-persistent webs with respect to the reduction in mean trophic position. There were no differences between persistent and non-persistent webs with respect to the reduction in diameter or average path length in any scenarios tested.

I also expected that invading species at low trophic levels should be more likely to become established. Comparing successful and unsuccessful invasions, introduced species that increased the average path length (by either participating at higher trophic levels or causing other species to increase their trophic position) were less likely to become established in the community. However, in some cases, successful introduced species still increased mean trophic position and had higher generality and vulnerability, suggesting that these successful introductions were mid-level well-connected consumers with both many prey and many predators. This pattern matches the ambivalent results of the regression models based on invader species' properties. In scenarios with Holling-based dynamics, greater trophic position, vulnerability, and generality decreased invasion success. However, in scenarios with consumer interference these properties increased success.

The motif profile of food webs has also been suggested to arise via selection on stability (Borrelli 2015). A selective process could occur via either the disassembly of webs with the preferential loss of species contributing to unstable three-species configurations, or from assembly processes favoring the establishment of species whose interactions increase the occurrence of stable subgraphs (Borrelli et al. 2015). The configurations that I expected to increase the stability of the web are the tritrophic chain, apparent competition, and exploitative competition (Prill et al. 2005, Borrelli 2015).

In this study, webs were less persistent with the deletion of species participating in many tritrophic chains. For apparent and exploitative competition, the removal of species from these modules had a variable effect on persistence, depending on the dynamical model structure. With Holling dynamics, non-persistent webs typically lost more exploitative competition modules, and when there was Type II or weak Type III functional response, they lost fewer apparent competition modules. With consumer interference, non-persistent webs tended to lose fewer apparent and exploitative competition modules. Although modules with double links were rare, persistent webs often lost more of them, which is what I expect because webs with fewer of them should be more stable.

The results of principle components regression were affected by the type of dynamics as well. There was a significant positive effect of tritrophic chain, apparent and exploitative competition modules with a weak Type III functional response and a negative effect with weak consumer interference. In other scenarios the loss of some of the stable modules may not have had a large impact on the stability of the whole web because of the relative rarity of unstable modules. Thus losing some stable modules would not have substantially altered the ratio of stable to unstable modules.

$PC_{sub\_inv}$ included tritrophic chains, apparent competition, exploitative competition and intraguild predation. With increasing the frequency of these stable modules, I expected the invader to have increased chances of successful establishment, but this was only the case with a weak Type III functional response. For most scenarios there was a negative effect of this principle component on invader success. Comparing the differences in the mean change in frequencies, however, in most scenarios with a consumer interference functional response successful invaders added more tritrophic chains, intraguild predation, and apparent and exploitative competition modules. However, contrary to expectation, in some scenarios successful invaders added more of the unstable modules. This again could result from the module frequencies being heavily weighted towards those that are stable, compared to those that are unstable.

The success or failure of establishment for an introduced species in a community is not the only possible outcome of invasion. Three out of the four possible impacts of invaders were represented in these simulations: establishment with no secondary extinctions, establishment with secondary extinctions, and no establishment with secondary extinctions. Whenever the introduced species failed to establish it always caused additional species to go extinct. Once again, however, comparing the mean change in frequencies across scenarios, there was no clear pattern distinguishing invaders that did or did not cause extinctions, or successful and unsuccessful invaders that caused extinctions.

The lack of any apparent pattern in the differences and effects of the frequencies of three-species modules may reflect the fact that food webs are more than the sum of their parts (Cohen et al. 2009). There may be emergent properties resulting from the interactions among the modules, for example, that alter their probability of being stable. Thus it may not be best to determine whether the addition of a module may lead to increased success based on its probability of stability in isolation, rather it may depend on both its internal stability and the context in which it is embedded.

For example, the intraguild predation module in isolation may be either stable or unstable (Borrelli 2015), and has been part of a larger debate over the stabilizing or destabilizing influence of omnivory in food webs (Pimm and Lawton 1978). Conventional wisdom states that when two consumers feed on the same resource and one consumer (the intraguild predator) feeds on its competitor (the intraguild prey), the only way for that configuration to be stable is if the consumed predator is a more efficient consumer of the shared resource. Kondoh (2008), however, demonstrated that the context in which an intraguild predation module is embedded, whether there are external interactions that are favorable for the intraguild predator or the intraguild prey, will impact its stability.

Moreover, there was a lack of consistency in my results for how changes in whole web properties affected stability. One possibility is that correlations among the different web properties have an affect on their relationship with food web stability. For example, increased mean generality or vulnerability of species in a web is directly a result of increased exploitative and apparent competition. Likewise, increased mean trophic position and average path length are related to the number of tritrophic chains in the food web.

Other studies that have attempted to predict the success or failure of invasions using similar models have found results similar to those found here. These studies have focused on either the properties of the web that is being invaded or the properties of the invader itself, rather than the change in web structure as a result of the introduction (Romanuk et al. 2009, Baiser et al. 2010). Baiser et al. (2010) found that connectance of the food web has a variable effect on invasion success that is dependent upon the trophic level of the introduced species. Alternatively, Romanuk et al. (2009) demonstrated that the best explanation for invasion success tended to be properties such as generality and vulnerability, although in some cases the particular properties of the invader made for better predictors. Unlike these previous studies, I did not categorize my predictions depending on the trophic category of the invader (basal, herbivore, carnivore), which may have caused some of the complex results. Additionally, where other studies used only a single model of dynamics, I looked at a range of model structures and parameterizations to gauge whether there were any differences caused by how dynamics are simulated. I found that model structure and parameterization can influence the properties that are the best predictors of invasion success.

Principle components regression, applied here, is a first step. Like previous studies that have utilized logistic regression (Baiser et al. 2010) and discriminant analysis (Romanuk et al. 2009), I found the best combination of variables to be used to predict an outcome. The indices that are being measured, however, are clearly related to one another (Vermaat et al. 2009). A more useful approach may be to use structural equation modeling. This would allow one to tease apart the various correlations among the measured indices to determine which are truly important for the stability of the food web during disassembly and assembly and how each one influences how the others affect the outcome. It would also allow for better predictive ability, as principle components as predictor variables in a model are not easily interpreted.

Future work could also expand on this study by incorporating additional realism into the bioenergetic model. The model used here is simplified in order to be more computationally tractable. Parameters such as metabolic rate, energetic efficiency, and the tuning parameter were kept constant for all species. The model could be modified to include variation in these parameters, possibly based on allometric relationships. Alternatively, additional parameters could be incorporated to relax the assumption of no prey preferences among predators. It is possible that incorporating additional complexity to the model could help to reconcile some of the conflicting results.

**Figure 5.1:** The mean change in three-species module frequency following the deletion of a species for webs that were persistent following species removal (red) and webs that were not (blue). Error bars represent 95% confidence intervals. Modules (rows) identified with an "s" are those with single directional links only, and include tritrophic chains (s1), intraguild predation (s2), three-species loops (s3), apparent competition (s4), and exploitative competition (s5). Modules with a "d" are the eight different modules that include bi-directional links (e.g., A eats B and B eats A). Columns represent the two functional response types used in the simulation; Holling (*Fij*) and consumer interference (*Fbd*). The x-axis is the value used for the tuning parameter (*q* or *c*).

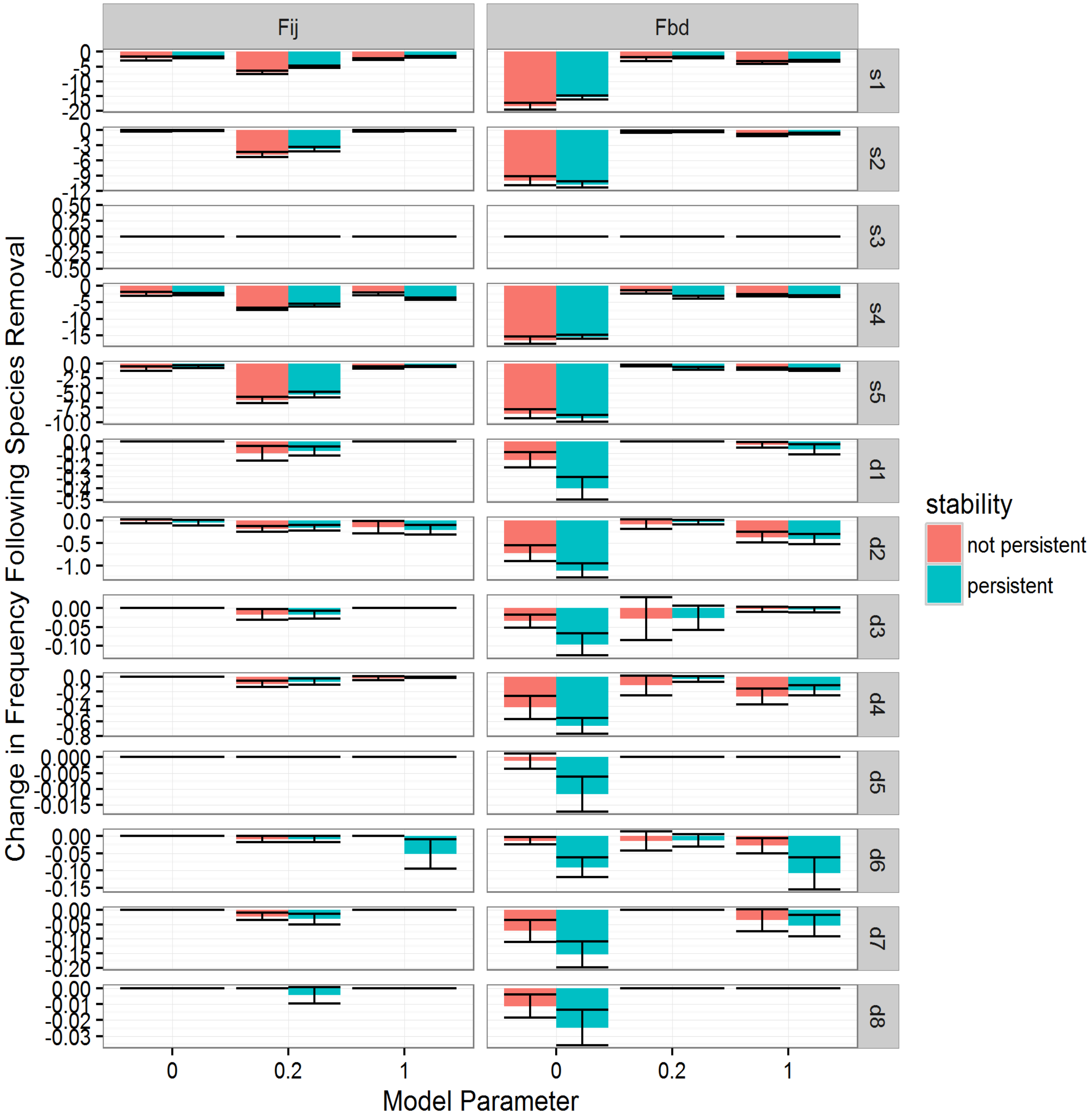


**Figure 5.2:** The mean species-level properties of the removed species for webs that were persistent following species removal (blue) and webs that were not (red). Error bars represent 95% confidence intervals. Measured species characteristics (rows) include generality (delGen), vulnerability (delVul), trophic position (delTP), and omnivory index (delOI). Figure columns represent the two functional response types used in the simulation; Holling (*Fij*) and consumer interference (*Fbd*). The x-axis is the value used for the tuning parameter (*q* or *c*).

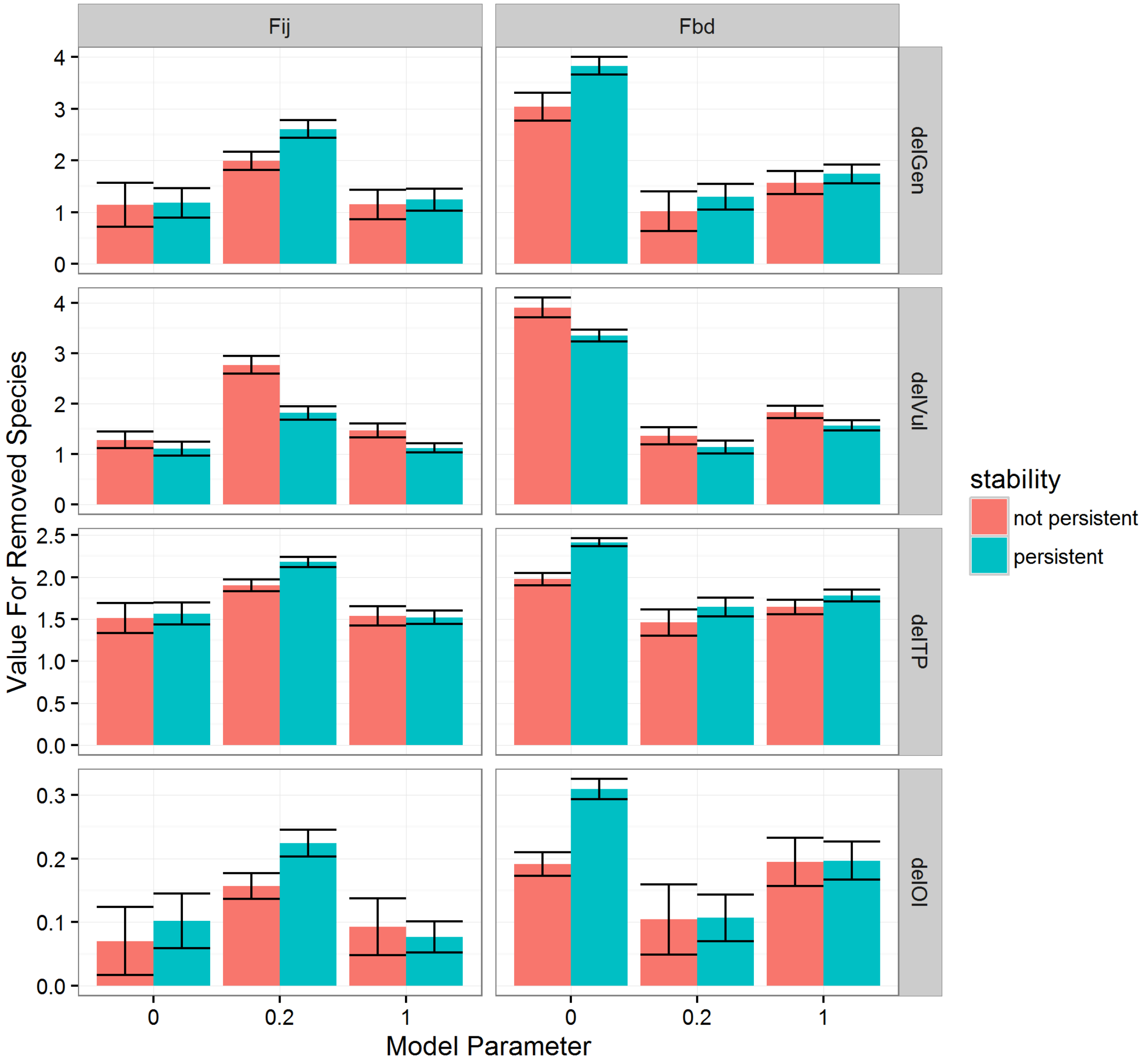


**Figure 5.3:** The mean change in food web properties following the deletion of a species for webs that were persistent following species removal and webs that were not. Error bars represent 95% confidence intervals. Food web properties measured (rows) are the number of links (L), mean generality (meanGen), mean vulnerability (meanVul), standard deviation of generality (sdGen), standard deviation of vulnerability (sdVul), mean trophic position (meanTP), standard deviation of trophic position (sdTP), mean omnivory index (meanOI), standard deviation of the omnivory index (sdOI), average path length (APL), and diameter (D). Figure columns represent the two functional response types used in the simulation; Holling (*Fij*) and consumer interference (*Fbd*). The x-axis is the value used for the tuning parameter (*q* or *c*).

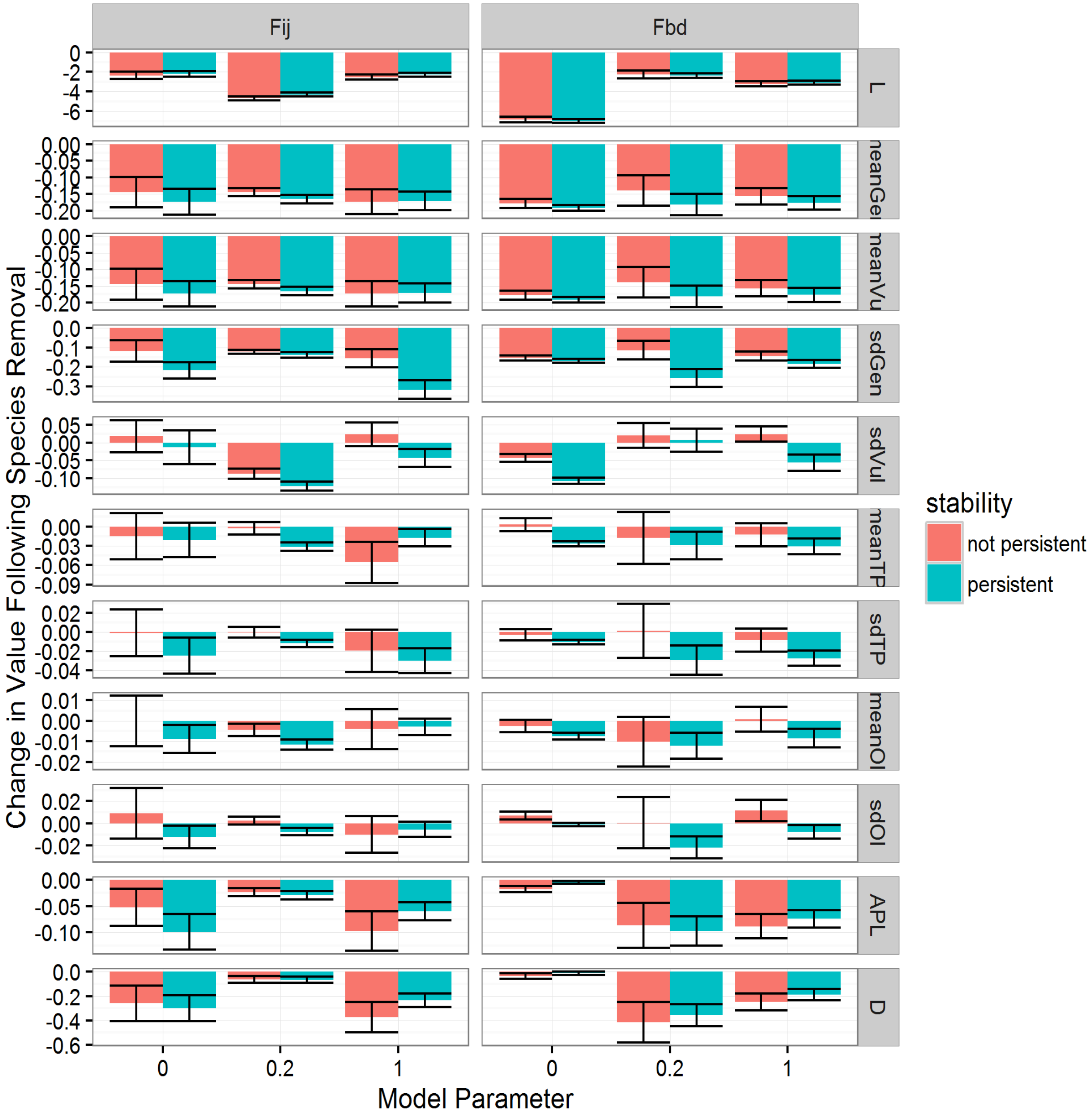


**Figure 5.4:** The mean change in food web properties following the introduction of a new species for successful (blue) and unsuccessful invasions (red). Error bars represent 95% confidence intervals. Food web properties measured (rows) are the number of links (L), mean generality (meanGen), mean vulnerability (meanVul), standard deviation of generality (sdGen), standard deviation of vulnerability (sdVul), mean trophic position (meanTP), standard deviation of trophic position (sdTP), mean omnivory index (meanOI), standard deviation of the omnivory index (sdOI), average path length (APL), and diameter (D). Figure columns represent the two functional response types used in the simulation; Holling (*Fij*) and consumer interference (*Fbd*). The x-axis is the value used for the tuning parameter (*q* or *c*).

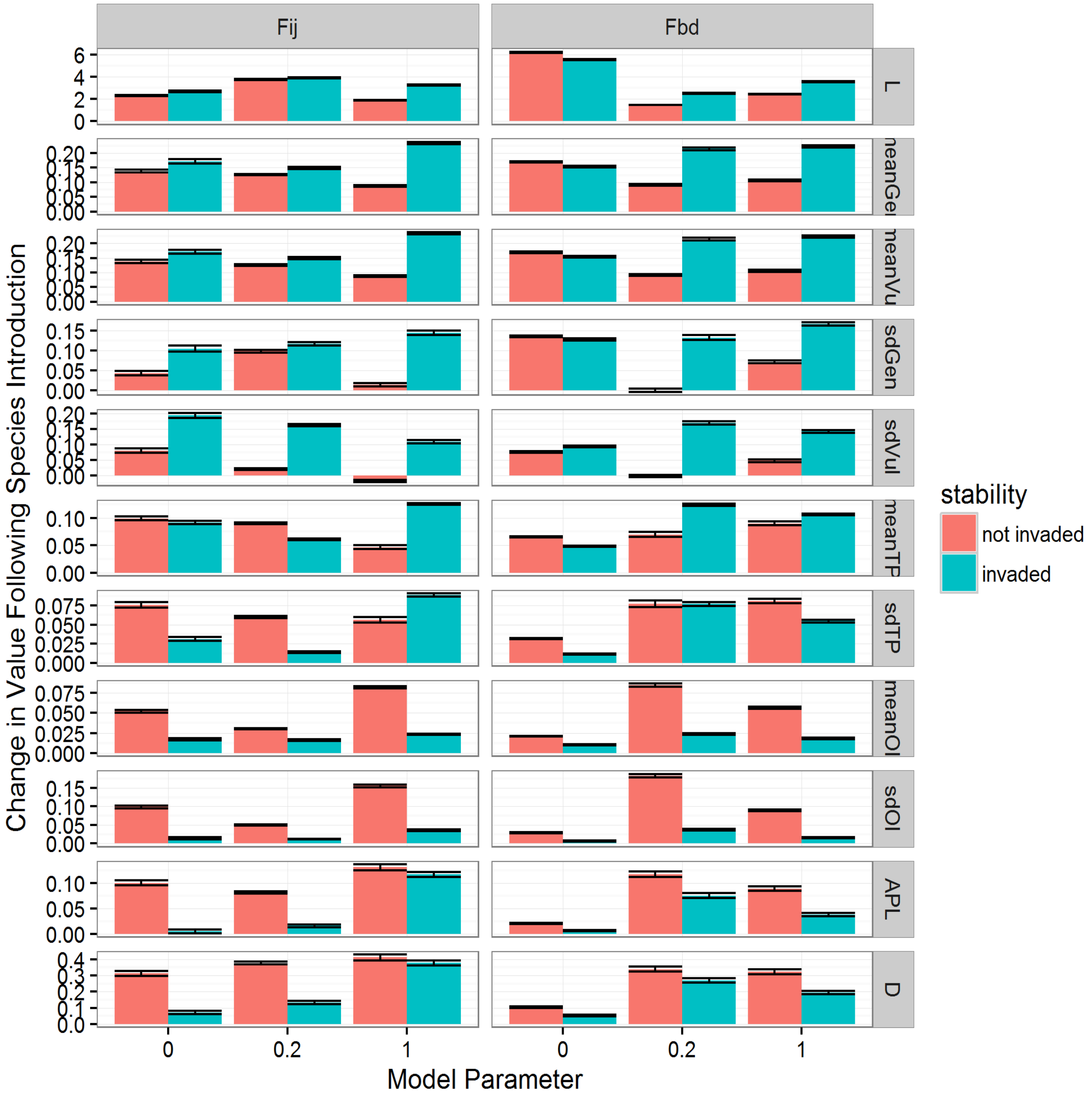


**Figure 5.5:** The mean change in three-species module frequency following the introduction of a species for successful (blue) and unsuccessful invasions (red). Error bars represent 95% confidence intervals. Modules (rows) identified with an "s" are those with single directional links only, and include tritrophic chains (s1), intraguild predation (s2), three-species loops (s3), apparent competition (s4), and exploitative competition (s5). Modules with a "d" are those that include bi-directional links (e.g., A eats B and B eats A). Figure columns represent the two functional response types used in the simulation; Holling (*Fij*) and consumer interference (*Fbd*). The x-axis is the value used for the tuning parameter (*q* or *c*).

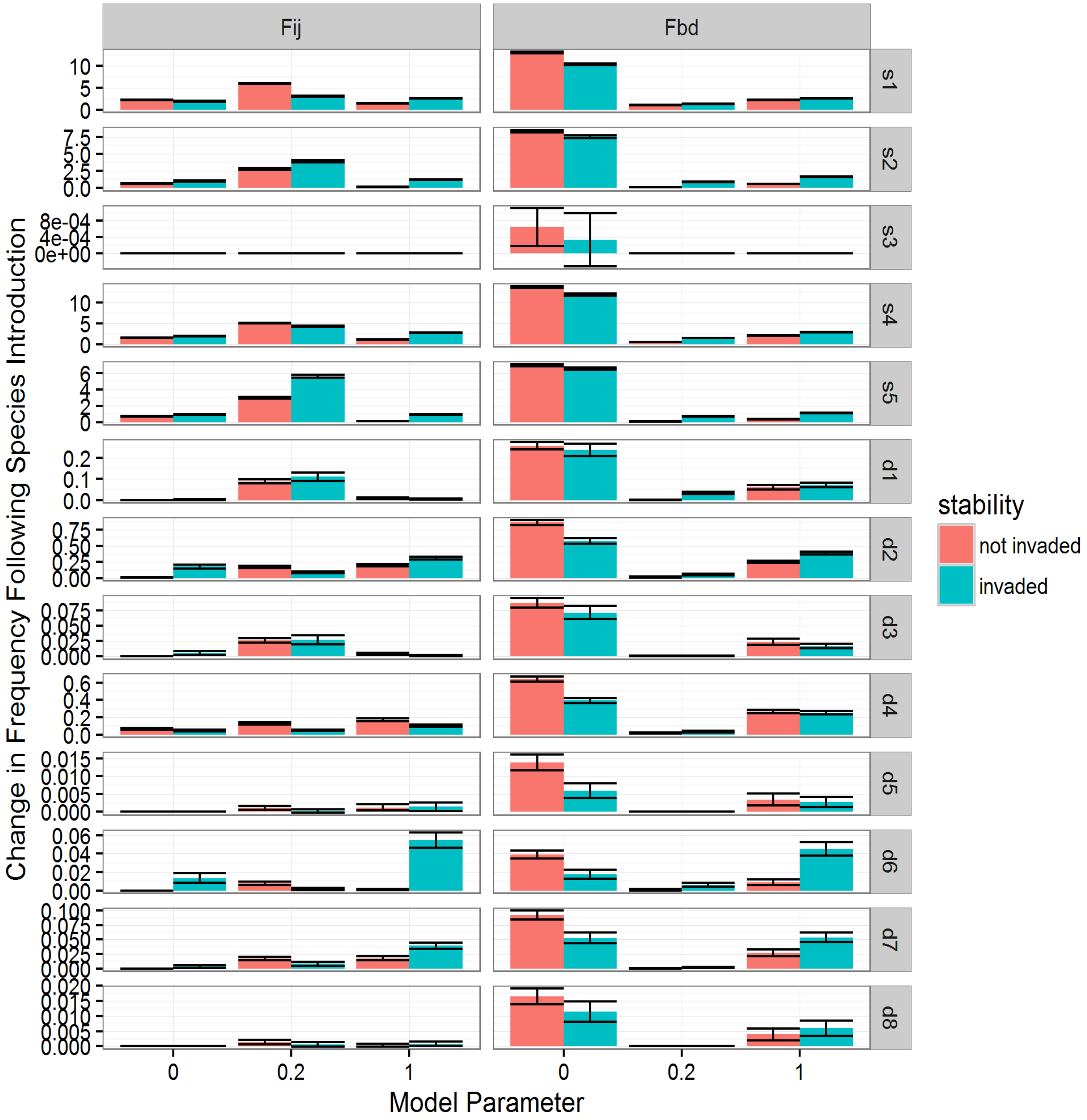


**Figure 5.6:** The mean introduced species characteristics for successful (blue) and unsuccessful invaders (red). Error bars represent 95% confidence intervals. Measured species characteristics (rows) include generality (delGen), vulnerability (delVul), trophic position (delTP), omnivory index (delOI), and number of secondary extinctions (dN). Figure columns represent the two functional response types used in the simulation; Holling (*Fij*) and consumer interference (*Fbd*). The x-axis is the value used for the tuning parameter (*q* or *c*).

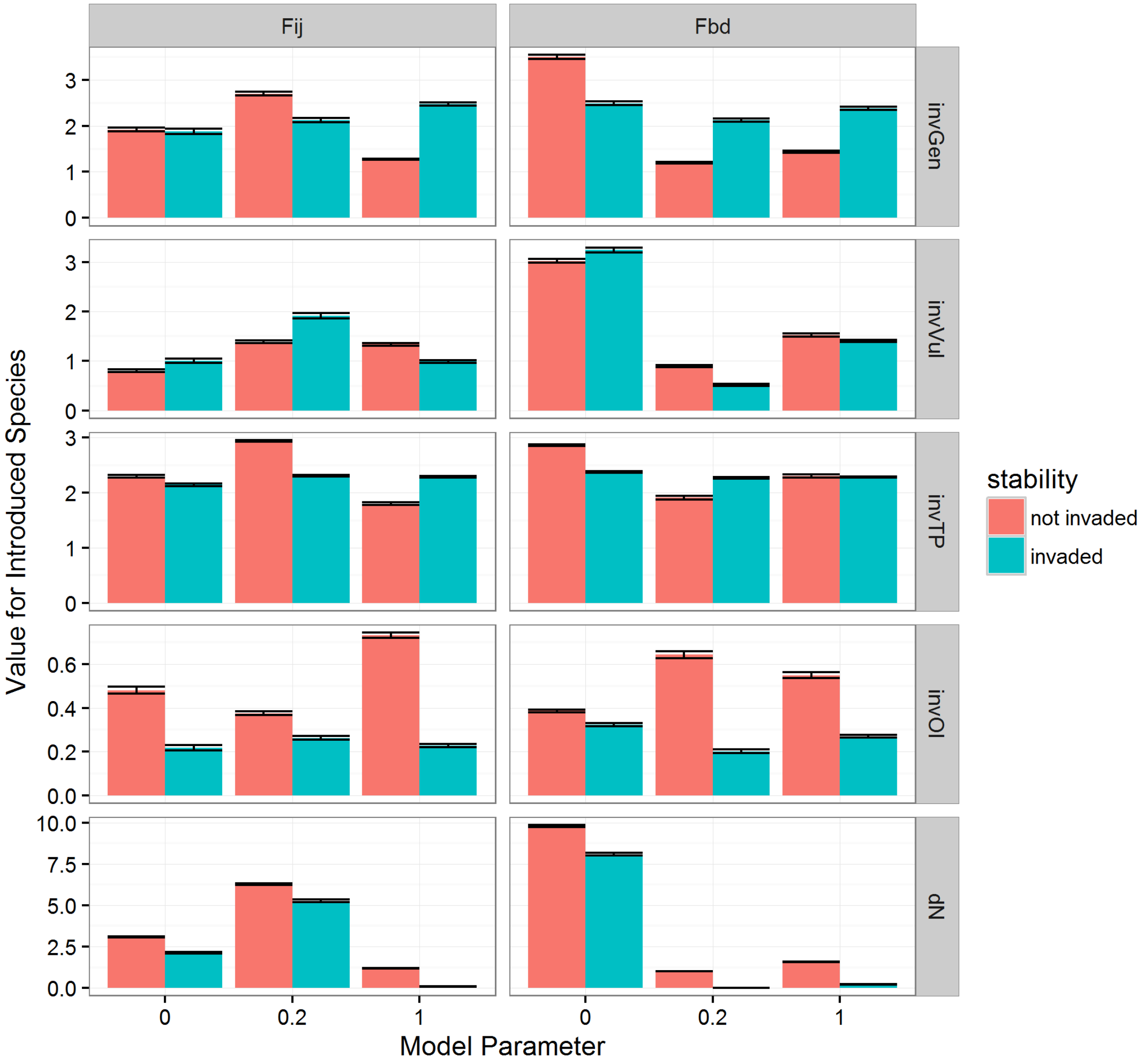
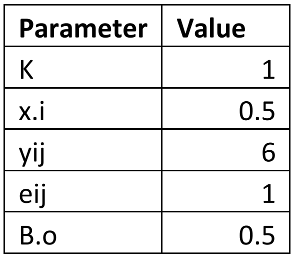


**Table 5.1:** Fixed parameters for all simulations using the bioenergetics model.



| Parameter | Value |
|---|---:|
| K | 1 |
| x.i | 0.5 |
| yij | 6 |
| eij | 1 |
| B.o | 0.5 |

**Table 5.2:** Definitions of measured food web properties.

| **Property** | **Definition** |
|---|---|
| Links | The number of connections made between the focal species and all other species in the food web. |
| Generality | The number of prey species a focal species has |
| Vulnerability | The number of predators a focal species has |
| Trophic Position | One plus the average trophic position of a species' prey, such that basal species have a trophic position of one. |
| Omnivory Index | The standard deviation of the trophic positions of a species' prey |
| Average Path Length | The average number of links between two species |
| Diameter | The longest shortest path from a basal species to a top predator |

**Table 5.3:** Outcomes of all 36 principle components regressions with linear mixed effects models (significant effects are in bold). Rows are grouped by functional response and parameter combination (*q* for Holling or *c* for consumer interference) scenario by shading.

| Simulation | Principle Components Predictor | Estimate | P value | AIC | $r^2$ | Functional Response | q or c |
|---|---|---|---|---|---|---|---|
| Deletion | Subgraph | -0.0716 | 0.3011 | 285 | 0.784 | Fij | 0 |
| Deletion | Web Property | -0.0873 | 0.3764 | 285 | 0.7991 | Fij | 0 |
| Deletion | Species Property | 0.1086 | 0.149 | 284 | 0.801 | Fij | 0 |
| Deletion | Subgraph | **0.0035** | 0.0042 | 8472 | 0.9054 | Fij | 0.2 |
| Deletion | Web Property | -0.0056 | 0.3302 | 8479 | 0.9077 | Fij | 0.2 |
| Deletion | Species Property | **0.0964** | 0 | 8236 | 0.8685 | Fij | 0.2 |
| Deletion | Subgraph | -0.0044 | 0.1128 | 5018 | 0.7398 | Fij | 1 |
| Deletion | Web Property | 0.0093 | 0.2792 | 5019 | 0.7396 | Fij | 1 |
| Deletion | Species Property | **0.2261** | 0 | 4458 | 0.5879 | Fij | 1 |
| Deletion | Subgraph | **0.1037** | 0.0281 | 421 | 0.8216 | Fbd | 0 |
| Deletion | Web Property | **-0.0244** | 0 | 426 | 0.8406 | Fbd | 0 |
| Deletion | Species Property | **0.082** | 0 | 424 | 0.8375 | Fbd | 0 |
| Deletion | Subgraph | **-0.1872** | 0 | 703 | 0.6808 | Fbd | 0.2 |
| Deletion | Web Property | -0.0667 | 0.1842 | 726 | 0.6928 | Fbd | 0.2 |
| Deletion | Species Property | **0.2046** | 0.0001 | 710 | 0.6452 | Fbd | 0.2 |
| Deletion | Subgraph | -0.0007 | 0.9611 | 1700 | 0.7442 | Fbd | 1 |
| Deletion | Web Property | -0.0322 | 0.1714 | 1698 | 0.7378 | Fbd | 1 |
| Deletion | Species Property | **0.1231** | 0 | 1674 | 0.7064 | Fbd | 1 |
| Invasion | Subgraph | 0.0239 | 0.0822 | 6612 | 0.8821 | Fij | 0 |
| Invasion | Web Property | **0.0906** | 0.0001 | 6599 | 0.8801 | Fij | 0 |
| Invasion | Species Property | **-0.1185** | 0 | 6593 | 0.8804 | Fij | 0 |
| Invasion | Subgraph | **0.0072** | 0 | 30940 | 0.9333 | Fij | 0.2 |
| Invasion | Web Property | **-0.0541** | 0 | 30865 | 0.9304 | Fij | 0.2 |
| Invasion | Species Property | **-0.1759** | 0 | 30193 | 0.9057 | Fij | 0.2 |
| Invasion | Subgraph | **-0.0146** | 0 | 18497 | 0.8987 | Fij | 1 |
| Invasion | Web Property | **0.0403** | 0 | 18501 | 0.8983 | Fij | 1 |
| Invasion | Species Property | **-0.2267** | 0 | 18055 | 0.8744 | Fij | 1 |
| Invasion | Subgraph | **-0.6154** | 0 | 10051 | 0.8135 | Fbd | 0 |
| Invasion | Web Property | **1.2347** | 0 | 9157 | 0.7247 | Fbd | 0 |
| Invasion | Species Property | **1.2816** | 0 | 8870 | 0.6984 | Fbd | 0 |
| Invasion | Subgraph | **-0.3765** | 0 | 15772 | 0.7927 | Fbd | 0.2 |
| Invasion | Web Property | **0.8308** | 0 | 14885 | 0.7446 | Fbd | 0.2 |
| Invasion | Species Property | **1.3282** | 0 | 13624 | 0.6639 | Fbd | 0.2 |
| Invasion | Subgraph | **-0.1536** | 0 | 17699 | 0.8626 | Fbd | 1 |
| Invasion | Web Property | **0.5911** | 0 | 16358 | 0.7798 | Fbd | 1 |
| Invasion | Species Property | **0.7479** | 0 | 16294 | 0.7784 | Fbd | 1 |

# Chapter 6: Conclusion

The idea of selection on stability is one of three components of a theoretical framework for understanding similarity among different systems. First, there must be some pattern generating mechanism. In food webs this is likely the familiar processes of invasion, migration (speciation and extinction over longer timescales), and adaptive foraging altering the configuration of the food web. Second, there must be some selective process. Because population dynamics and stability have been major areas of research on food webs, a likely selection process would be selection against unstable configurations. Finally, this selective process should lead to observed patterns. Comparing food web structures that are more likely to be stable, those with a higher degree of quasi sign-stability, seem also to be those that are more commonly observed in empirical networks.

Model food webs composed of short food chains on average are more quasi sign-stable than those composed of longer food chains, all else being equal (Chapter 2). Given that food chains of varying lengths are generated via community assembly, species participating in longer food chains should be more likely to go extinct as they are in less stable configurations. Selection against unstable longer food chain lengths should then tend to generate a higher frequency of short food chains. Analysis of a collection of fifty empirically described food webs showed that species tend to have trophic positions, one plus the mean trophic position of their prey, between three and five, and none had a trophic position higher than six. Likewise, food webs most often had a diameter, the longest shortest path from basal species to top predator, less than 5 (Chapter 2).

Food webs may also be decomposed into smaller *n*-species components. The most dynamically interesting components (that are still computationally tractable) are three-species configurations. Of the thirteen possible three-species modules, three are very likely to be stable, one is equally likely to be stable or not, and the rest are very unlikely to be stable. Using the collection of 50 empirical food webs, I assessed whether those modules that were more quasi sign-stable are also those that are more common than expected by chance, motifs. Using two null models of varying constraint I demonstrated that the stable modules are also the motifs in empirical food webs (Chapter 3).

That there is a link between the stability of certain food web components and their frequency in real food webs does not exactly constitute proof that selection on stability generates the observed patterns. It may, however, be a signature of such a process, as the predictions made by the theoretical framework are matched by the data. The observed patterns may also be generated by alternative mechanisms. For example, some three-species configurations may be more likely to occur in networks with some kind of hierarchy or ordering, like size structure in food webs (Paulau et al. 2015). A null model that incorporates these probabilities, like the niche model, is essential in differentiating the roles of these mechanisms. In order further explore the signature of selection on stability, I chose to analyze dynamical models of multispecies predator-prey systems. As there are currently no freely available resources to apply any of the many I developed an R package to simulate the biomass dynamics of species in food webs (Chapter 4).

Selection on stability presupposes two potential mechanisms to generate food web patterns that can then be selected, the extinction and introduction of new species. Given that the signature of the selective process is the prevalence of a given pattern over others, species that contribute to food web structures that have low stability should be less important to the stability of the whole web. In other words, if the loss of species leads to a food web configuration that is composed of stable elements, there should be fewer secondary extinctions. Alternatively, new species that are added to the food web should be more likely to become established in the community if they contribute to food web components that are stable.

Based on Chapters 2 and 3, both short food chains and network motifs exhibit the signature of selection on stability. Thus, when species removals lead to longer chains or fewer stable subgraphs, the expectation is that the resulting web should be less stable. Likewise, newly introduced species that lead to shorter average food chains and stable subgraph frequency should be more likely to successfully invade the community. In Chapter 5 I tested the idea that changes in food web properties, like average chain length and module frequency, had an effect on food web persistence or the success of invasion.

There were substantial differences in the effects of changes in food web properties that were dependent on the type and parameterization of functional response. The removal of species from higher trophic levels and reducing mean trophic position often resulted in persistent webs, which matches the prediction. When the functional response included interference, the removal of species contributing to unstable modules was also associated with persistent food webs. Successful invaders often had a higher trophic position, but did not increase average path length as much, suggesting that they acted to increase the frequency of short chains as well. However, the expectation with respect to three-species modules did not match predictions; successful invaders added fewer stable modules and occasionally more unstable modules.

A mismatch between predictions and results for three-species modules may be caused by context dependence. Unstable configurations may be supported by being embedded within a web of stable configurations. Kondoh (2008) demonstrated that three-species modules that are not internally persistent (not all species have positive equilibrium densities), may persist given appropriate external interactions. These unstable configurations may also be able to persist if their constituent interactions are weak relative to stronger interactions in stable configurations. To test this would require food webs with measured interaction strengths. The subset of the web with the stronger interactions should be composed only of stable three-species configurations, while subsets with weak interactions may include configurations that are less likely to be stable.

It may be worthwhile to reexamine these results in the context of species' roles (Stouffer et al. 2012) rather than via module frequency. Stouffer et al. (2012) showed that species' roles, quantified as both how they are embedded in the network and their dynamical importance, are conserved spatially and phylogenetically. Moreover, in their examination of 32 food webs they found that in many webs relatively few unique roles were represented, suggesting that redundancy may play a part in stabilizing the community. Species with dynamically important roles in food webs may be more likely to successfully invade. Alternatively, a community's resistance to invasion may depend on the diversity of roles it contains.

In the future, I intend to further develop the theoretical foundation of selection on stability in food webs by both addressing the limitations of the model used in this dissertation and expanding on my work. Many of the parameters in the bioenergetic model described in Chapter 4 and used in Chapter 5 are constants instead of varying among the different species. For example, all basal species have the same growth rate ($r_i$), and all species have the same metabolic rate ($x_i$) and extinction threshold. These parameters could instead be based on either species' trophic position or niche value (if based on the niche model). For example, species higher in the food web or with larger niche value (i.e., body size) could have a higher threshold for extinction such that they would be considered extinct at higher biomasses. Additionally predators do not have prey preferences, which could be included in either functional response used in Chapters 4 and 5. I think it will be important to explore how additional complexity can affect the predictions made in Chapter 5.

Moving forward I will also incorporate stochasticity into the bioenergetic model. Currently, the R package described in Chapter 4 only allows for the deterministic model. Adding stochasticity will allow me to generate results as probabilistic outcomes as opposed to binary success or failures. A stochastic model would also allow me to determine the risk of extinction for species based on how they contribute to food web structure. I will expand on the results of Chapter 3 using the stochastic model, by first assessing extinction risk for species based on their position in each of the three-species configurations. The next step would then be to assess risk for species in four-species configurations to determine how risk changes when species participate in larger networks. Based on the results of Chapter 3, I would expect risk to be higher for species in unstable three-species configurations, and lower in tritrophic chains, apparent competition, and exploitative competition. Likewise in larger webs when species participate primarily in stable three-species subgraphs they should have lower risk.

In Chapter 5 species were added or removed from equilibrium food webs to determine how food web structure emerges from selection on stability. A better approach may be to use community assembly models. Starting from one or few species, new species would be added to the community from a larger species pool. By varying the order of introductions I could examine how the context a species is introduced into may alter the probability of a successful invasion or lead to alternative equilibrium communities. I would expect that despite leading to communities composed of different species, selection on the stability of these communities should generate similar structures among them.

The idea of selection on stability can be a simple, yet powerful tool for explaining ecological similarity. Moreover, that there is some selection process operating based on the stability of community configurations is widespread in ecology, but has only recently been formalized into a theoretical framework (Borrelli et al. 2015). Examples can be found at multiple scales in ecology. At the level of predator-prey pairs, selection on stability may be responsible for the observed lack of combinations of high attack rates and high handling times (Kidd and Amarasekare 2012, Novak 2013, Johnson and Amarasekare 2015). Such high-high combinations in models of predator-prey dynamics lead to increased oscillations in abundances (Rosenzweig 1971). Oscillatory dynamics lead to increased risk of stochastic extinctions during periods of low abundance. Thus pairs of species exhibiting this behavior should be observed fairly infrequently (Borrelli et al. 2015), a prediction that is borne out by empirical data (Gilpin 1975, Kendall et al. 1998).

Within populations, selection on stability may be used to explain the generation of social hierarchies. Many species exhibit a particular type of hierarchy in small groups. This hierarchy of dominance among individuals is typically linear in structure. There is a dominant individual to whom all others are subordinate, and among the subordinates there is a dominant individual, and this continues throughout the group (Chase et al. 2002, Chase and Lindquist 2009, Lindquist and Chase 2009). Considering triads of individuals, linear hierarchies are transitive in structure. Experimental evidence demonstrates that when triads become intransitive (i.e., a subordinate attacks a more dominant individual) the intransitivity does not last long and a linear hierarchy is rapidly recovered (Chase and Lindquist 2009). Thus the observation of linear dominance hierarchies may be a result of the preferential loss of unstable configurations (Borrelli et al. 2015).

While the idea of selection on stability offers an enticing theoretical framework, and indeed is often implicitly recognized in ecology, there is yet to be clear empirical evidence supporting it. The large amount of variability found in the modeling approach that I used in Chapter 5 suggests that these ideas should be tested against empirical observations. Unfortunately, this seems to be a serious obstacle due to the difficulty in obtaining food web data. More promising avenues may be through investigating selection on stability operating on networks of plants and their pollinators, for which there are richer temporal data (e.g., Olesen et al. 2008, Petanidou et al. 2008, Dupont et al. 2009). Using time-series network data we may be able more explicitly test whether species that are more prone to extinction or invasion depending on how they alter the structure of the system. For example, species contributing to the nestedness of plant-pollinator webs may be less likely to go extinct, as nestedness is thought to increase the feasibility and stability of the system (Rohr et al. 2014).

# Bibliography

Abrams, P. A., and L. R. Ginzburg. 2000. The nature of predation: prey dependent, ratio dependent or neither? Trends in Ecology & Evolution 15:337–341.

Alarcón, R., N. M. Waser, and J. Ollerton. 2008. Year-to-year variation in the topology of a plant-pollinator interaction network. Oikos 117:1796–1807.

Albrecht, M., M. Riesen, and B. Schmid. 2010. Plant-pollinator network assembly along the chronosequence of a glacier foreland. Oikos 119:1610–1624.

Allesina, S., and J. M. Levine. 2011. A competitive network theory of species diversity. Proceedings of the National Academy of Sciences 108:5638–5642.

Allesina, S., and M. Pascual. 2008. Network structure, predator–prey modules, and stability in large food webs. Theoretical Ecology 1:55–64.

Allesina, S., and S. Tang. 2012. Stability criteria for complex ecosystems. Nature 483:205–208.

Allhoff, K. T., and B. Drossel. 2013. When do evolutionary food web models generate complex networks? Journal of Theoretical Biology 334:122–129.

Angulo, M. T., Y. Liu, and J. Slotine. 2015. Network motifs emerge from interconnections that favor stability. Science 11:848–852.

Arditi, R., and L. R. Ginzburg. 2012. How Species Interact: Altering the Standard View on Trophic Ecology. Oxford University Press, New York, New York, USA.

Arim, M., and P. A. Marquet. 2004. Intraguild predation: a widespread interaction related to species biology. Ecology Letters 7:557–564.

Baird, D., and R. Ulanowicz. 1989. The seasonal dynamics of the Chesapeake Bay ecosystem. Ecological Monographs 59:329–364.

Baiser, B., H. L. Buckley, N. J. Gotelli, and A. M. Ellison. 2012. Predicting food-web structure with metacommunity models. Oikos 122:492–506.

Baiser, B., G. J. Russell, and J. L. Lockwood. 2010. Connectance determines invasion success via trophic interactions in model food webs. Oikos 119:1970–1976.

Bascompte, J. 2009. Disentangling the web of life. Science 325:416–419.

Bascompte, J. 2010. Structure and dynamics of ecological networks. Science 329:765–766.

Bascompte, J., P. Jordano, C. J. Melián, and J. M. Olesen. 2003. The nested assembly of plant-animal mutualistic networks. Proceedings of the National Academy of Sciences 100:9383–9387.

Bascompte, J., and C. J. Melián. 2005. Simple trophic modules for complex food webs. Ecology 86:2868–2873.

Bastolla, U., M. A. Fortuna, A. Pascual-García, A. Ferrera, B. Luque, and J. Bascompte. 2009.

The architecture of mutualistic networks minimizes competition and increases biodiversity. Nature 458:1018–1020.

Beaudoin, C. P., E. E. Prepas, W. M. Tonn, L. I. Wassenaar, and B. G. Kotak. 2001. A stable carbon and nitrogen isotope study of lake food webs in Canada’s Boreal Plain. Freshwater Biology 46:465–477.

Beddington, J. 1975. Mutual interference between parasites or predators and its effect on searching efficiency. Journal of Animal Ecology 44:331–340.

Berlow, E. L., A.-M. Neutel, J. E. Cohen, P. C. de Ruiter, B. Ebenman, M. Emmerson, J. W. Fox, V. A. A. Jansen, J. I. Jones, G. D. Kokkoris, D. O. Logofet, A. J. McKane, J. M. Montoya, and O. Petchey. 2004. Interaction strengths in food webs: issues and opportunities. Journal of Animal Ecology 73:585–598.

Borrelli, J. J. 2015. Selection against instability: stable subgraphs are most frequent in empirical food webs. Oikos 124:1583–1588.

Borrelli, J. J., S. Allesina, P. Amarasekare, R. Arditi, I. Chase, J. Damuth, R. D. Holt, D. O. Logofet, M. Novak, R. P. Rohr, A. G. Rossberg, M. Spencer, J. K. Tran, and L. R. Ginzburg. 2015. Selection on stability across ecological scales. Trends in Ecology & Evolution 30:1–9.

Borrelli, J. J., and L. R. Ginzburg. 2014. Why there are so few trophic levels: selection against instability explains the pattern. Food Webs 1:10–17.

Borrett, S. R., and M. K. Lau. 2014. enaR : An R package for Ecosystem Network Analysis. Methods in Ecology and Evolution 5:1206–1213.

Briand, F. 1983. Environmental control of food web structure. Ecology 64:253–263.

Briand, F., and J. Cohen. 1984. Community food webs have scale-invariant structure. Nature 307:264–267.

Brose, U., R. J. Williams, and N. D. Martinez. 2006. Allometric scaling enhances stability in complex food webs. Ecology Letters 9:1228–1236.

Caldarelli, G., P. Higgs, and A. McKane. 1998. Modelling coevolution in multispecies communities. Journal of Theoretical Biology 193:345–358.

Camacho, J., D. B. Stouffer, and L. A. N. Amaral. 2007. Quantitative analysis of the local structure of food webs. Journal of Theoretical Biology 246:260–268.

Cattin, M., L. Bersier, C. Banašek-Richter, R. Baltensperger, and J. P. Gabriel. 2004. Phylogenetic constraints and adaptation explain food-web structure. Nature 427:835–839.

Chase, I. D., and W. B. Lindquist. 2009. Dominance Hierarchies. *in* P. Hedström and P. Bearman, editors. The Oxford Handbook of Analytical Sociology. Oxford University Press, Oxford.

Chase, I. D., C. Tovey, D. Spangler-Martin, and M. Manfredonia. 2002. Individual differences versus social dynamics in the formation of animal dominance hierarchies. Proceedings of the National Academy of Sciences 99:5744–5749.

Christensen, V., and D. Pauly. 1992. ECOPATH II—a software for balancing steady-state ecosystem models and calculating network characteristics. Ecological Modelling 61:169–185.

Christian, R. R., and J. J. Luczkovich. 1999. Organizing and understanding a winter's seagrass foodweb network through effective trophic levels. Ecological Modelling 117:99–124.

Christianou, M., and G. D. Kokkoris. 2008. Complexity does not affect stability in feasible model communities. Journal of Theoretical Biology 253:162–169.

Cohen, J. E., F. Briand, and C. M. Newman. 1986. A stochastic theory of community food webs III: predicted and observed lengths of food chains. Proceedings of the Royal Society B 228:317–353.

Cohen, J. E., and C. M. Newman. 1985. A stochastic theory of community food webs: I. Models and aggregated data. Proceedings of the Royal Society B 224:421–448.

Cohen, J. E., D. N. Schittler, D. G. Raffaelli, and D. C. Reuman. 2009. Food webs are more than the sum of their tritrophic parts. Proceedings of the National Academy of Sciences 106:22335–22340.

Cohen, J., and C. Newman. 1988. Dynamic basis of food web organization. Ecology 69:1655–1664.

Côté, P., and L. Parrott. 2006. Controlling food web structure by optimization of a community assembly model. Ecological Informatics 1:125–131.

Csárdi, G., and T. Nepusz. 2006. The igraph software package for complex network research. InterJournal Complex Systems:1695.

DeAngelis, D. L., R. a. Goldstein, and R. V. O'Neill. 1975. A model for trophic interaction. Ecology 56:881–892.

Donohue, I., O. L. Petchey, J. M. Montoya, A. L. Jackson, L. McNally, M. Viana, K. Healy, M. Lurgi, N. E. O'Connor, and M. C. Emmerson. 2013. On the dimensionality of ecological stability. Ecology Letters 16:421–429.

Dormann, C. F., J. Frund, N. Bluthgen, and B. Gruber. 2009. Indices, graphs and null models: analyzing bipartite ecological networks. The Open Ecology Journal 2:7–24.

Dormann, C. F., B. Gruber, and J. Fründ. 2008. Introducing the bipartite package: analysing ecological networks. R News 8:8–11.

Doulcier, G., and D. B. Stouffer. 2015. Rnetcarto: Fast network modularity and roles computation by simulated annealing.

Drossel, B., P. G. Higgs, and A. J. McKane. 2001. The influence of predator-prey population dynamics on the long-term evolution of food web structure. Journal of Theoretical Biology 208:91–107.

Drossel, B., and A. J. McKane. 2003. Modelling food webs. Pages 218–247 *in* S. Bornholdt and H. G. Schuster, editors. Handbook of Graphs and Networks: From the Genome to the Internet. Wiley-VCH Verlag GmbH & Co. KGaA, Weinheim, FRG.

Drossel, B., A. J. McKane, and C. Quince. 2004. The impact of nonlinear functional responses on the long-term evolution of food web structure. Journal of Theoretical Biology 229:539–548.

Dunne, J. A. 2005. The network structure of food webs. Pages 27–86 *in* M. Pascual and J. A. Dunne, editors. Ecological Networks: Linking Structure to Dynamics in Food Webs. Oxford University Press, Oxford.

Dunne, J. A., K. D. Lafferty, A. P. Dobson, R. F. Hechinger, A. M. Kuris, N. D. Martinez, J. P. McLaughlin, K. N. Mouritsen, R. Poulin, K. Reise, D. B. Stouffer, D. W. Thieltges, R. J. Williams, and C. D. Zander. 2013. Parasites affect food web structure primarily through increased diversity and complexity. PLoS Biology 11:e1001579.

Dunne, J. A., and R. J. Williams. 2009. Cascading extinctions and community collapse in model food webs. Philosophical Transactions of the Royal Society of London. Series B, Biological sciences 364:1711–23.

Dunne, J. A., R. J. Williams, and N. D. Martinez. 2002. Food-web structure and network theory: the role of connectance and size. Proceedings of the National Academy of Sciences 99:12917–12922.

Dupont, Y. L., B. Padrón, J. M. Olesen, and T. Petanidou. 2009. Spatio-temporal variation in the structure of pollination networks. Oikos 118:1261–1269.

Eklöf, A., U. Jacob, J. Kopp, J. Bosch, R. Castro-Urgal, N. P. Chacoff, B. Dalsgaard, C. de Sassi, M. Galetti, P. R. Guimarães, S. B. Lomáscolo, A. M. Martín González, M. A. Pizo, R. Rader, A. Rodrigo, J. M. Tylianakis, D. P. Vázquez, and S. Allesina. 2013. The dimensionality of ecological networks. Ecology Letters 16:577–583.

Elton, C. 1927. Animal Ecology. Macmillan Co., New York, New York, USA.

Elton, C. 1958. The Ecology Of Invasions By Plants And Animals. Methuen, London.

Fahimipour, A. K., and A. M. Hein. 2014. The dynamics of assembling food webs. Ecology Letters 17:606–613.

Fawcett, T. W., and A. D. Higginson. 2012. Heavy use of equations impedes communication among biologists. Proceedings of the National Academy of Sciences 109:11735–11739.

Fortuna, M. A., D. B. Stouffer, J. M. Olesen, P. Jordano, D. Mouillot, B. R. Krasnov, R. Poulin, and J. Bascompte. 2010. Nestedness versus modularity in ecological networks: two sides of the same coin? Journal of Animal Ecology 79:811–817.

Fridley, J., J. Stachowicz, S. Naeem, D. Sax, E. Seabloom, M. Smith, T. Stohlgren, D. Tilman, and B. V. Holle. 2007. The invasion paradox: reconciling pattern and process in species invasions. Ecology 88:3–17.

Gilpin, M. E. 1975. Group Selection in Predator-Prey Communities. Princeton University Press, Princeton, NJ.

Ginzburg, L., and H. Akçakaya. 1992. Consequences of ratio-dependent predation for steady-state properties of ecosystems. Ecology 73:1536–1543.

Glazier, D. S. 2012. Temperature affects food-chain length and macroinvertebrate species richness in spring ecosystems. Freshwater Science 31:575–585.

Goldwasser, L., and J. Roughgarden. 1993. Construction and analysis of a large Carribean food web. Ecology 74:1216–1233.

Guimarães, P. R., G. Machado, M. A. M. de Aguiar, P. Jordano, J. Bascompte, A. Pinheiro, and S. F. Dos Reis. 2007. Build-up mechanisms determining the topology of mutualistic networks. Journal of theoretical biology 249:181–189.

Hall, S., and D. Raffaelli. 1991. Food-web patterns: lessons from a species-rich web. Journal of Animal Ecology 60:823–841.

Havens, K. 1992. Scale and structure in natural food webs. Science 257:1107–1109.

Hechinger, R. F., K. D. Lafferty, J. P. McLaughlin, B. L. Fredensborg, T. C. Huspeni, J. Lorda, P. K. Sandhu, J. C. Shaw, M. E. Torchin, K. L. Whitney, and A. M. Kuris. 2011. Food webs including parasites, biomass, body sizes, and life stages for three California/Baja California estuaries. Ecology 92:791.

Holling, C. S. 1959a. The components of predation as revealed by a study of small-mammal predation of the European Pine Sawfly. The Canadian Entomologist 91:293–320.

Holling, C. S. 1959b. Some characteristics of simple types of predation and parasitism. The Canadian Entomologist 91:385–398.

Holt, R. 1996. Food webs in space: an island biogeographic perspective. Pages 313–323 *in* G. Polis and K. O. Winemiller, editors. Food Webs. Chapman and Hall.

Holt, R. D. 2002. Food webs in space: On the interplay of dynamic instability and spatial processes. Ecological Research 17:261–273.

Holyoak, M., and S. Sachdev. 1998. Omnivory and the stability of simple food webs. Oecologia 117:413–419.

Hudson, L. N., R. Emerson, G. B. Jenkins, K. Layer, M. E. Ledger, D. E. Pichler, M. S. A. Thompson, E. J. O’Gorman, G. Woodward, and D. C. Reuman. 2013. Cheddar: analysis and visualisation of ecological communities in R. Methods in Ecology and Evolution 4:99–104.

Hutchinson, G. E. 1959. Homage to Santa Rosalia or why are there so many kinds of animals? The American Naturalist 93:145–159.

Iacus, S. M. 2015. sde: Simulation and Inference for Stochastic Differential Equations.

Ings, T. C., J. M. Montoya, J. Bascompte, N. Blüthgen, L. Brown, C. F. Dormann, F. Edwards, D. Figueroa, U. Jacob, J. I. Jones, R. B. Lauridsen, M. E. Ledger, H. M. Lewis, J. M. Olesen, F. J. F. van Veen, P. H. Warren, and G. Woodward. 2009. Ecological networks--beyond food webs. Journal of Animal Ecology 78:253–269.

Ives, A. R., and S. R. Carpenter. 2007. Stability and diversity of ecosystems concepts of stability. Science 317:58–62.

Jaarsma, N. G., S. M. de Boer, C. R. Townsend, R. M. Thompson, and E. D. Edwards. 1998. Characterising food-webs in two New Zealand streams. New Zealand Journal of Marine and Freshwater Research 32:271–286.

James, A., M. J. Plank, A. G. Rossberg, J. Beecham, M. Emmerson, and J. W. Pitchford. 2015. Constructing random matrices to represent real ecosystems. The American Naturalist 185:680-692.

Johnson, C. A., and P. Amarasekare. 2013. Competition for benefits can promote the persistence of mutualistic interactions. Journal of Theoretical Biology 328:54–64.

Johnson, C. A., and P. Amarasekare. 2015. A metric for quantifying the oscillatory tendency of consumer-resource interactions. The American Naturalist 185:87–99.

Johnson, S., V. Domínguez-García, L. Donetti, and M. a. Muñoz. 2014. Trophic coherence determines food-web stability. Proceedings of the National Academy of Sciences 111:17923–17928.

Kéfi, S., E. L. Berlow, E. A. Wieters, S. A. Navarrete, O. L. Petchey, S. A. Wood, A. Boit, L. N. Joppa, K. D. Lafferty, R. J. Williams, N. D. Martinez, B. A. Menge, C. A. Blanchette, A. C. Iles, and U. Brose. 2012. More than a meal… integrating non-feeding interactions into food webs. Ecology Letters 15:291–300.

Kendall, Prendergast, and Bjørnstad. 1998. The macroecology of population dynamics: taxonomic and biogeographic patterns in population cycles. Ecology Letters 1:160–164.

Kidd, D., and P. Amarasekare. 2012. The role of transient dynamics in biological pest control: insights from a host-parasitoid community. Journal of Animal Ecology 81:47–57.

Kondoh, M. 2008. Building trophic modules into a persistent food web. Proceedings of the National Academy of Sciences 105:16631–16635.

Kondoh, M., S. Kato, and Y. Sakato. 2010. Food webs are built up with nested subwebs.

Ecology 91:3123–3130.

Kones, J. K., K. Soetaert, D. van Oevelen, and J. Owino. 2009. Are network indices robust indicators of food web functioning? A Monte Carlo approach. Ecological Modelling 220:370–382.

Krause, A. E., K. A. Frank, D. M. Mason, R. E. Ulanowicz, and W. W. Taylor. 2003. Compartments revealed in food-web structure. Nature 426:282–285.

Lafferty, K. D., S. Allesina, M. Arim, C. J. Briggs, G. De Leo, A. P. Dobson, J. A. Dunne, P. T. J. Johnson, A. M. Kuris, D. J. Marcogliese, N. D. Martinez, J. Memmott, P. A. Marquet, J. P. McLaughlin, E. A. Mordecai, M. Pascual, R. Poulin, and D. W. Thieltges. 2008. Parasites in food webs: the ultimate missing links. Ecology Letters 11:533–546.

Lafferty, K. D., A. P. Dobson, and A. M. Kuris. 2006. Parasites dominate food web links. Proceedings of the National Academy of Sciences 103:11211–11216.

Lafferty, K. D., and A. M. Kuris. 2009. Parasites reduce food web robustness because they are sensitive to secondary extinction as illustrated by an invasive estuarine snail. Philosophical Transactions of the Royal Society of London. Series B, Biological sciences 364:1659–63.

Lindeman, R. 1942. The trophic-dynamic aspect of ecology. Ecology 23:399–417.

Lindquist, W. B., and I. D. Chase. 2009. Data-based analysis of winner-loser models of hierarchy formation in animals. Bulletin of Mathematical Biology 71:556–584.

Link, J. 2002. Does food web theory work for marine ecosystems? Marine Ecology Progress Series 230:1–9.

Lockwood, J. L., R. D. Powell, M. P. Nott, and S. L. Pimm. 1997. Assembling ecological communities in time and space. Oikos 80:549–553.

Loeuille, N., and M. Loreau. 2005. Evolutionary emergence of size-structured food webs. Proceedings of the National Academy of Sciences 102:5761–5766.

Loeuille, N., and M. Loreau. 2010. Emergence of complex food web structure in community evolution models. *in* H. A. Verhoef and P. J. Morin, editors. Community Ecology. Oxford University Press, Oxford.

Long, Z. T., J. F. Bruno, and J. E. Duffy. 2011. Food chain length and omnivory determine the stability of a marine subtidal food web. Journal of Animal Ecology 80:586–594.

Lurgi, M., N. Galiana, B. C. López, L. N. Joppa, and J. M. Montoya. 2014. Network complexity and species traits mediate the effects of biological invasions on dynamic food webs. Frontiers in Ecology and Evolution 2:1–11.

MacArthur, R. 1955. Fluctuations of animal populations and a measure of community stability. Ecology 36:533–536.

Martinez, N. 1991. Artifacts or attributes? Effects of resolution on the Little Rock Lake food web. Ecological Monographs 61:367–392.

Martinez, N., B. Hawkins, H. Dawah, and B. Feifarek. 1999. Effects of sampling effort on characterization of food-web structure. Ecology 80:1044–1055.

May, R. M. 1972. Will a large complex system be stable? Nature 238:413–414.

May, R. M. 1973a. Stability and Complexity in Model Ecosystems. Princeton University Press, Princeton, NJ.

May, R. M. 1973b. Qualitative stability in model ecosystems. Ecology 54:638–641.

McCann, K., A. Hastings, and G. R. Huxel. 1998. Weak trophic interactions and the balance of nature. Nature 395:794–798.

McCann, K. S., J. B. Rasmussen, and J. Umbanhowar. 2005. The dynamics of spatially coupled food webs. Ecology Letters 8:513–523.

Medan, D., R. P. J. Perazzo, M. Devoto, E. Burgos, M. G. Zimmermann, H. Ceva, and A. M. Delbue. 2007. Analysis and assembling of network structure in mutualistic systems. Journal of Theoretical Biology 246:510–521.

Memmott, J., N. D. Martinez, and J. E. Cohen. 2000. Predators, parasitoids and pathogens: species richness, trophic generality and body sizes in a natural food web. Journal of Animal Ecology 69:1–15.

Milo, R., S. Shen-Orr, S. Itzkovitz, N. Kashtan, D. Chklovskii, and U. Alon. 2002. Network motifs: simple building blocks of complex networks. Science 298:824–827.

Morin, P., and S. Lawler. 1996. Effects of food chain length and omnivory on population dynamics in experimental food webs. Pages 218–230 *in* G. A. Polis and K. O. Winemiller, editors. Food Webs. Chapman and Hall, New York, New York, USA.

Mouritsen, K. N., R. Poulin, J. P. McLaughlin, and D. W. Thieltges. 2011. Food web including metazoan parasites for an intertidal ecosystem in New Zealand. Ecology 92:2006.

Neutel, A. M., J. A. P. Heesterbeek, and P. C. de Ruiter. 2002. Stability in real food webs: weak links in long loops. Science 296:1120–1123.

Novak, M. 2013. Trophic omnivory across a productivity gradient: intraguild predation theory and the structure and strength of species interactions. Proceedings of the Royal Society B: Biological Sciences 280:20131415.

Olesen, J. M., J. Bascompte, H. Elberling, and P. Jordano. 2008. Temporal dynamics in a pollination network. Ecology 89:1573–1582.

Opitz, S. 1996. Trophic Interactions in Caribbean Coral Reefs. ICALRM Tech, Makati City, Philippines.

Paine, R. T. 1966. Food web complexity and species diversity. The American Naturalist 100:65–75.

Paine, R. T. 1969. A note on trophic complexity and community stability. The American Naturalist 103:91–93.

Paine, R. T. 1980. Food webs: linkage, interaction strength and community infrastructure. Journal of Animal Ecology 49:666–685.

Paulau, V. P., C. Feenders, and B. Blasius. 2015. Motif analysis in directed ordered networks and applications to food webs. Scientific Reports 5:11926.

Pawar, S. 2009. Community assembly, stability and signatures of dynamical constraints on food web structure. Journal of Theoretical Biology 259:601–612.

Perdomo, G., R. Thompson, and P. Sunnucks. submitted. foodweb: an open-source program for the visualisation and analysis of compilations of complex food webs. Environmental Modelling and Software.

Petanidou, T., A. S. Kallimanis, J. Tzanopoulos, S. P. Sgardelis, and J. D. Pantis. 2008. Long-term observation of a pollination network: fluctuation in species and interactions, relative invariance of network structure and implications for estimates of specialization. Ecology Letters 11:564–75.

Petchey, O. L., U. Brose, and B. C. Rall. 2010. Predicting the effects of temperature on food web connectance. Philosophical Transactions of the Royal Society of London. Series B, Biological Sciences 365:2081–2091.

Pimm, S. L. 1982. Food Webs. University of Chicago Press, Chicago, Illinois, U.S.A.

Pimm, S. L. 1984. The complexity and stability of ecosystems. Nature 307:321–326.

Pimm, S. L., and J. Lawton. 1978. On feeding on more than one trophic level. Nature 275:542–544.

Pimm, S. L., and J. H. Lawton. 1977. Number of trophic levels in ecological communities. Nature 268:329–331.

Plein, M., L. Längsfeld, E. Neuschulz, C. Schultheiß, L. Ingmann, T. Töpfer, K. Böhning-Gaese, and M. Schleuning. 2013. Constant properties of plant-frugivore networks despite fluctuations in fruit and bird communities in space and time. Ecology 94:1296–1306.

Poisot, T. 2015. betalink: Beta-Diversity of Species Interactions.

Polis, G. A. 1991. Complex trophic interactions in deserts: an empirical critique of food-web theory. The American Naturalist 138:123–155.

Polis, G. A., W. B. Anderson, and R. D. Holt. 1997. Toward an integration of landscape and food web ecology : the dynamics of spatially subsidized food webs. Annual Review of Ecology and Systematics 28:289–316.

Post, D. M. 2002. The long and short of food-chain length. Trends in Ecology & Evolution 17:269–277.

Post, D. M. 2007. Testing the productive-space hypothesis: rational and power. Oecologia 153:973–984.

Post, D. M., M. L. Pace, and N. G. Hairston. 2000. Ecosystem size determines food-chain length in lakes. Nature 405:1047–1049.

Poulin, R. 2010. Network analysis shining light on parasite ecology and diversity. Trends in Parasitology 26:492–498.

Preston, D. L., S. A. Orlofske, J. P. McLaughlin, and P. T. Johnson. 2012. Food web including infectious agents for a California freshwater pond. Ecology 93:1760.

Prill, R. J., P. A. Iglesias, and A. Levchenko. 2005. Dynamic properties of network motifs contribute to biological network organization. PLoS Biology 3:1881–1892.

Proulx, S. R., D. E. L. Promislow, and P. C. Phillips. 2005. Network thinking in ecology and evolution. Trends in Ecology & Evolution 20:345–353.

R Core Team, R. 2014. R: A Language and Environment for Statistical Computing. R Foundation for Statistical Computing, Vienna, Austria.

Rezende, E. L., E. M. Albert, M. A. Fortuna, and J. Bascompte. 2009. Compartments in a marine food web associated with phylogeny, body mass, and habitat structure. Ecology Letters 12:779–788.

Rip, J. M. K., K. S. McCann, D. H. Lynn, and S. Fawcett. 2010. An experimental test of a fundamental food web motif. Proceedings of the Royal Society B: Biological Sciences 277:1743–1749.

Roberts, A. 1974. The stability of a feasible random ecosystem. Nature 251:607–608.

Rohr, R. P., S. Saavedra, and J. Bascompte. 2014. On the structural stability of mutualistic systems. Science 345:1253497–1253497.

Romanuk, T. N., Y. Zhou, U. Brose, E. L. Berlow, R. J. Williams, and N. D. Martinez. 2009. Predicting invasion success in complex ecological networks. Philosophical Transactions of the Royal Society of London. Series B, Biological Sciences 364:1743–1754.

Roopnarine, P. D., and R. Hertog. 2012. Detailed food web networks of three Greater Antillean coral reef systems: the Cayman Islands, Cuba, and Jamaica. Dataset Papers in Ecology 2013:857470. http://dx.doi.org/10.7167/2013/857470.

Rosenzweig, M. L. 1971. Paradox of enrichment: destabilization of exploitation ecosystems in ecological time. Science 171:385–387.

Saavedra, S., R. P. Rohr, J. M. Olesen, and J. Bascompte. 2016. Nested species interactions promote feasibility over stability during the assembly of a pollinator community. Ecology and Evolution 6:997–1007.

Schoener, T. W. 1989. Food webs from the small to the large: the Robert H. MacArthur award lecture. Ecology 70:1559–1589.

Skalski, G. T., and J. F. Gilliam. 2001. Functional responses with predator interference: viable alternatives to the Holling type II model. Ecology 82:3083–3092.

Soetaert, K., T. Petzoldt, and R. W. Setzer. 2010. Package deSolve: solving initial value differential equations in R. Journal Of Statistical Software 33:1–25.

Spencer, M., and P. Warren. 1996. The effects of habitat size and productivity on food web structure in small aquatic microcosms. Oikos 75:419–430.

Stegen, J. C., R. Ferriere, and B. J. Enquist. 2012. Evolving ecological networks and the emergence of biodiversity patterns across temperature gradients. Proceedings of the Royal Society B: Biological Sciences 279:1051–1060.

Sterner, R. W., A. Bajpai, and T. Adams. 1997. The enigma of food chain length: absence of theoretical evidence for dynamic constraints. Ecology 78:2258–2262.

Stouffer, D. B. 2010. Scaling from individuals to networks in food webs. Functional Ecology 24:44–51.

Stouffer, D. B., and J. Bascompte. 2011. Compartmentalization increases food-web persistence. Proceedings of the National Academy of Sciences 108:3648–3652.

Stouffer, D. B., J. Camacho, and L. A. N. Amaral. 2006. A robust measure of food web intervality. Proceedings of the National Academy of Sciences 103:19015–19020.

Stouffer, D. B., J. Camacho, R. Guimera, C. A. Ng, and L. A. N. Amaral. 2005. Quantitative patterns in the structure of model and empirical food webs. Ecology 86:1301–1311.

Stouffer, D. B., J. Camacho, W. Jiang, and L. A. N. Amaral. 2007. Evidence for the existence of a robust pattern of prey selection in food webs. Proceedings of the Royal Society B: Biological Sciences 274:1931–1940.

Strona, G., D. Nappo, F. Boccacci, S. Fattorini, and J. San-Miguel-Ayanz. 2014. A fast and unbiased procedure to randomize ecological binary matrices with fixed row and column totals. Nature Communications 5:4114.

Takimoto, G., D. Spiller, and D. Post. 2008. Ecosystem size, but not disturbance determines food chain length on islands of the Bahamas. Ecology 89:3001–3007.

Tang, S., and S. Allesina. 2014. Reactivity and stability of large ecosystems. Frontiers in Ecology and Evolution 2:1–8.

Tang, S., S. Pawar, and S. Allesina. 2014. Correlation between interaction strengths drives stability in large ecological networks. Ecology Letters 17:1094–1100.

Thébault, E., and C. Fontaine. 2010. Stability of ecological communities and the architecture of mutualistic and trophic networks. Science 329:853–856.

Thieltges, D., K. Reise, and K. Mouritsen. 2011. Food web including metazoan parasites for a tidal basin in Germany and Denmark. Ecology 92:2005.

Thompson, R., and E. Edwards. 2001. Allocation of effort in stream food-web studies: the best compromise? Marine Freshwater Research 52:339–345.

Thompson, R., and M. Hemberg. 2009. The ubiquity of omnivory. Verh. Internat. Verein. Limnol. 30:761–764.

Thompson, R. M., M. Hemberg, B. M. Starzomski, and J. B. Shurin. 2007. Trophic levels and trophic tangles: the prevalence of omnivory in real food webs. Ecology 88:612–617.

Thompson, R. M., and C. R. Townsend. 2000. Is resolution the solution?: The effect of taxonomic resolution on the calculated properties of three stream food webs. Freshwater Biology 44:413–422.

Thompson, R., and C. Townsend. 1999. The effect of seasonal variation on the community structure and food-web attributes of two streams: implications for food-web science. Oikos 87:75–88.

Thompson, R., and C. Townsend. 2003. Impacts on stream food webs of native and exotic forest: an intercontinental comparison. Ecology 84:145–161.

Thompson, R., and C. Townsend. 2005. Energy availability, spatial heterogeneity and ecosystem size predict food-web structure in streams. Oikos 108:137–148.

Townsend, C. R., R. M. Thompson, A. R. Mcintosh, C. Kilroy, E. Edwards, and M. R. Scarsbrook. 1998. Disturbance, resource supply, and food-web architecture in streams. Ecology Letters 1:200–209.

Ulanowicz, R. E., R. D. Holt, and M. Barfield. 2013. Limits on ecosystem trophic complexity: insights from ecological network analysis. Ecology Letters 17:127–136.

Vermaat, J. E., J. A. Dunne, and A. J. Gilbert. 2009. Major dimensions in food-web structure properties. Ecology 90:278–282.

Waide, R., and W. Reagan. 1996. The Food Web of a Tropical Rainforest. University of Chicago. University of Chicago Press, Chicago, Illinois, U.S.A.

Warren, P. 1989. Spatial and temporal variation in the structure of a freshwater food web. Oikos 55:299–311.

Williams, R. J., and N. D. Martinez. 2000. Simple rules yield complex food webs. Nature

404:180–183.

Williams, R. J., and N. D. Martinez. 2004a. Stabilization of chaotic and non-permanent food-web dynamics. European Physical Journal B 38:297–303.

Williams, R. J., and N. D. Martinez. 2004b. Limits to trophic levels and omnivory in complex food webs: theory and data. The American Naturalist 163:458–468.

Williams, R. J., and N. D. Martinez. 2005. Diversity, complexity, and persistence in large model ecosystems. Pages 163–186 *in* M. Pascual and J. A. Dunne, editors. Ecological Networks: Linking Structure to Dynamics in Food Webs. Oxford University Press, Oxford.

Yodzis, P. 1981. The structure of assembled communities. Journal of Theoretical Biology 92:103–117.

Yodzis, P. 1998. Local trophodynamics and the interaction of marine mammals and fisheries in the Benguela ecosystem. Journal of Animal Ecology 67:635–658.

Yodzis, P. 2000. Diffuse effects in food webs. Ecology 81:261–266.

Yodzis, P., and S. Innes. 1992. Body size and consumer-resource dynamics. The American Naturalist 139:1151–1175.

Young, H. S., D. J. McCauley, R. B. Dunbar, M. S. Hutson, A. M. Ter-Kuile, and R. Dirzo. 2013. The roles of productivity and ecosystem size in determining food chain length in tropical terrestrial ecosystems. Ecology 94:692–701.

Vander Zanden, M., B. Shuter, N. Lester, and J. Rasmussen. 1999. Patterns of food chain length in lakes: a stable isotope study. The American Naturalist 154:406–416.

Zander, C. D., N. Josten, K. C. Detloff, R. Poulin, J. P. McLaughlin, and D. W. Thieltges. 2011. Food web including metazoan parasites for a brackish shallow water ecosystem in Germany and Denmark. Ecology 92:2007.

# Appendix 1: Data and annotated code for Chapter 1

## Data Sources for Chapter 1

Three food webs were downloaded from the Dryad Digital Repository (Roopnarine & Hertog 2012a, 2012b). Another seven were available from Ecological Archives (Hechinger et al. 2011; Mouritsen et al. 2011; Thieltges et al. 2011; Zander et al. 2011; Preston et al. 2012). Fourteen webs were provided by Jennifer Dunne of the PEaCE Lab (Baird & Ulanowicz 1989; Warren 1989; Polis 1991; Hall & Raffaelli 1991; Martinez 1991; Christensen & Pauly 1992; Havens 1992; Goldwasser & Roughgarden 1993; Opitz 1996; Waide & Reagan 1996; Yodzis 1998, 2000; Christian & Luczkovich 1999; Martinez et al. 1999; Memmott et al. 2000; Link 2002) that were analyzed in (Dunne et al. 2002, 2004). The remaining 26 food webs were downloaded from the Interaction Web Database (Jaarsma et al. 1998; Townsend et al. 1998; Thompson & Townsend 1999, 2000, 2003, 2005; Thompson & Edwards 2001). We also used a subset of sixteen ecosystem networks provided by Robert Ulanowicz through his website, and used most recently in Ulanowicz et al. 2014.

## References

Baird, D., and R. Ulanowicz. 1989. The seasonal dynamics of the Chesapeake Bay ecosystem. Ecological Monographs 59:329–364.

Christensen, V., and D. Pauly. 1992. ECOPATH II—a software for balancing steady-state ecosystem models and calculating network characteristics. Ecological Modelling 61:169–185.

Christian, R. R., and J. J. Luczkovich. 1999. Organizing and understanding a winter’s seagrass foodweb network through effective trophic levels. Ecological Modelling 117:99–124.

Dunne, J. A., R. J. Williams, and N. D. Martinez. 2002. Food-web structure and network theory: The role of connectance and size. Proceedings of the National Academy of Sciences 99:12917–12922.

Dunne, J., R. Williams, and N. Martinez. 2004. Network structure and robustness of marine food webs. Marine Ecology Progress Series 273:291–302.

Goldwasser, L., and J. Roughgarden. 1993. Construction and analysis of a large Carribean food web. Ecology 74:1216–1233.

Hall, S., and D. Raffaelli. 1991. Food-web patterns: lessons from a species-rich web. Journal of Animal Ecology 60:823–841.

Havens, K. 1992. Scale and structure in natural food webs. Science 257:1107–1109.

Hechinger, R. F., K. D. Lafferty, J. P. McLaughlin, B. L. Fredensborg, T. C. Huspeni, J. Lorda, P. K. Sandhu, J. C. Shaw, M. E. Torchin, K. L. Whitney, and A. M. Kuris. 2011. Food webs including parasites, biomass, body sizes, and life stages for three California/Baja California estuaries. Ecology 92:791.

Jaarsma, N. G., S. M. de Boer, C. R. Townsend, R. M. Thompson, and E. D. Edwards. 1998. Characterising food-webs in two New Zealand streams. New Zealand Journal of Marine and Freshwater Research 32:271–286.

Link, J. 2002. Does food web theory work for marine ecosystems? Marine Ecology Progress Series 230:1–9.

Martinez, N. 1991. Artifacts or attributes? Effects of resolution on the Little Rock Lake food web. Ecological Monographs 61:367–392.

Martinez, N., B. Hawkins, H. Dawah, and B. Feifarek. 1999. Effects of sampling effort on characterization of food-web structure. Ecology 80:1044–1055.

Memmott, J., N. D. Martinez, and J. E. Cohen. 2000. Predators, parasitoids and pathogens: species richness, trophic generality and body sizes in a natural food web. Journal of Animal Ecology 69:1–15.

Mouritsen, K. N., R. Poulin, J. P. McLaughlin, and D. W. Thieltges. 2011. Food web including metazoan parasites for an intertidal ecosystem in New Zealand. Ecology 92:2006.

Opitz, S. 1996. Trophic Interactions in Caribbean Coral Reefs. ICALRM Tech, Makati City, Philippines.

Polis, G. A. 1991. Complex trophic interactions in deserts: an empirical critique of food-web theory. The American Naturalist 138:123–155.

Preston, D. L., S. A. Orlofske, J. P. McLaughlin, and P. T. Johnson. 2012. Food web including infectious agents for a California freshwater pond. Ecology 93:1760.

Roopnarine, P. D., and R. Hertog. 2012. Detailed food web networks of three Greater Antillean coral reef systems: the Cayman Islands, Cuba, and Jamaica. Dataset Papers in Ecology 2013:857470.

Thieltges, D. W., K. Reise, K. N. Mouritsen, J. P. McLaughlin, and R. Poulin. 2011. Food web including metazoan parasites for a tidal basin in Germany and Denmark. Ecology 92:2005.

Thompson, R., and E. Edwards. 2001. Allocation of effort in stream food-web studies: the best compromise? Marine Freshwater Research 52:339–345.

Thompson, R. M., and C. R. Townsend. 2000. Is resolution the solution?: the effect of taxonomic resolution on the calculated properties of three stream food webs. Freshwater Biology 44:413–422.

Thompson, R., and C. Townsend. 1999. The effect of seasonal variation on the community structure and food-web attributes of two streams: implications for food-web science. Oikos 87:75–88.

Thompson, R., and C. Townsend. 2003. Impacts on stream food webs of native and exotic forest: an intercontinental comparison. Ecology 84:145–161.

Thompson, R., and C. Townsend. 2005. Energy availability, spatial heterogeneity and ecosystem size predict food-web structure in streams. Oikos 108:137–148.

Townsend, C. R., R. M. Thompson, A. R. Mcintosh, C. Kilroy, E. Edwards, and M. R. Scarsbrook. 1998. Disturbance, resource supply, and food-web architecture in streams. Ecology Letters 1:200–209.

Ulanowicz, R. E., R. D. Holt, and M. Barfield. 2014. Limits on ecosystem trophic complexity: insights from ecological network analysis. Ecology Letters 17:127–136.

Waide, R., and W. Reagan. 1996. The Food Web of a Tropical Rainforest. University of Chicago. University of Chicago Press, Chicago, Illinois, U.S.A.

Warren, P. 1989. Spatial and temporal variation in the structure of a freshwater food web. Oikos 55:299–311.

Yodzis, P. 1998. Local trophodynamics and the interaction of marine mammals and fisheries in the Benguela ecosystem. Journal of Animal Ecology 67:635–658.

Yodzis, P. 2000. Diffuse effects in food webs. Ecology 81:261–266.

Zander, C. D., N. Josten, K. C. Detloff, R. Poulin, J. P. McLaughlin, and D. W. Thieltges. 2011. Food web including metazoan parasites for a brackish shallow water ecosystem in Germany and Denmark. Ecology 92:2007.

**Code to assess impact of interaction strength on the degree of omnivory**

Required libraries

```
library(igraph)
library(NetIndices)
library(data.table)
library(ggplot2)
library(reshape2)
```

Import the Ulanowicz data

Determine the threshold flow weight above which we call "strong" interactions. Here we subset the Ulanowicz webs by taking just the top 50%, 40%, 30%, 20%, and 10%.

```
q50 <- sapply(ulanEDGE, function(x){quantile(x[,3], .5)})
q60 <- sapply(ulanEDGE, function(x){quantile(x[,3], .6)})
q70 <- sapply(ulanEDGE, function(x){quantile(x[,3], .7)})
q80 <- sapply(ulanEDGE, function(x){quantile(x[,3], .8)})
q90 <- sapply(ulanEDGE, function(x){quantile(x[,3], .9)})
```

```
uEDGE.50 <- list()
uEDGE.60 <- list()
uEDGE.70 <- list()
uEDGE.80 <- list()
uEDGE.90 <- list()
for(i in 1:length(ulanEDGE)){
  uEDGE.50[[i]] <- ulanEDGE[[i]][which(ulanEDGE[[i]][,3] >= q50[i]),]
  uEDGE.60[[i]] <- ulanEDGE[[i]][which(ulanEDGE[[i]][,3] >= q60[i]),]
  uEDGE.70[[i]] <- ulanEDGE[[i]][which(ulanEDGE[[i]][,3] >= q70[i]),]
  uEDGE.80[[i]] <- ulanEDGE[[i]][which(ulanEDGE[[i]][,3] >= q80[i]),]
  uEDGE.90[[i]] <- ulanEDGE[[i]][which(ulanEDGE[[i]][,3] >= q90[i]),]
}

uGRAPH <- lapply(uEDGE.50, function(x){graph.edgelist(as.matrix(x[,1:2]))})
uGRAPH6 <- lapply(uEDGE.60, function(x){graph.edgelist(as.matrix(x[,1:2]))})
uGRAPH7 <- lapply(uEDGE.70, function(x){graph.edgelist(as.matrix(x[,1:2]))})
uGRAPH8 <- lapply(uEDGE.80, function(x){graph.edgelist(as.matrix(x[,1:2]))})
uGRAPH9 <- lapply(uEDGE.90, function(x){graph.edgelist(as.matrix(x[,1:2]))})

uMAT <- lapply(uGRAPH, get.adjacency, sparse = F)
uMAT6 <- lapply(uGRAPH6, get.adjacency, sparse = F)
uMAT7 <- lapply(uGRAPH7, get.adjacency, sparse = F)
uMAT8 <- lapply(uGRAPH8, get.adjacency, sparse = F)
uMAT9 <- lapply(uGRAPH9, get.adjacency, sparse = F)

tind <- lapply(uMAT, TrophInd)
tind6 <- lapply(uMAT6, TrophInd)
tind7 <- lapply(uMAT7, TrophInd)
tind8 <- lapply(uMAT8, TrophInd)
tind9 <- lapply(uMAT9, TrophInd)

prop.omn <- sapply(tind, function(x){(1 - sum(x$OI == 0)/nrow(x)) * 100})
prop.omn6 <- sapply(tind6, function(x){(1 - sum(x$OI == 0)/nrow(x)) * 100})
prop.omn7 <- sapply(tind7, function(x){(1 - sum(x$OI == 0)/nrow(x)) * 100})
prop.omn8 <- sapply(tind8, function(x){(1 - sum(x$OI == 0)/nrow(x)) * 100})
prop.omn9 <- sapply(tind9, function(x){(1 - sum(x$OI == 0)/nrow(x)) * 100})

p.omn <- matrix(c(prop.omn, prop.omn6, prop.omn7, prop.omn8, prop.omn9), ncol = 5)
colnames(p.omn) <- seq(50, 90, 10)
```

```
omn.dat <- data.frame(prop = seq(50, 90, 10), omn = colMeans(p.omn),
                      upper = apply(p.omn, 2,
                                    function(x){mean(x) + 1.96 * (sd(x)/sqrt(length(x)))}),
                      lower = apply(p.omn, 2,
                                    function(x){mean(x) - 1.96 * (sd(x)/sqrt(length(x)))}))
```

```
ggplot(omn.dat, aes(x = prop, y = omn)) + geom_errorbar(aes(x = prop, ymax = upper, ymin =
lower), width = 1) + geom_point(data = data.frame(melt(p.omn)), aes(x = Var2, y = value),
alpha = .25, position = "jitter") + geom_point() + theme_bw() + xlab("Flow Weight Percentile
Cutoff (percentage)") + ylab("Percent Omnivorous Species")
```

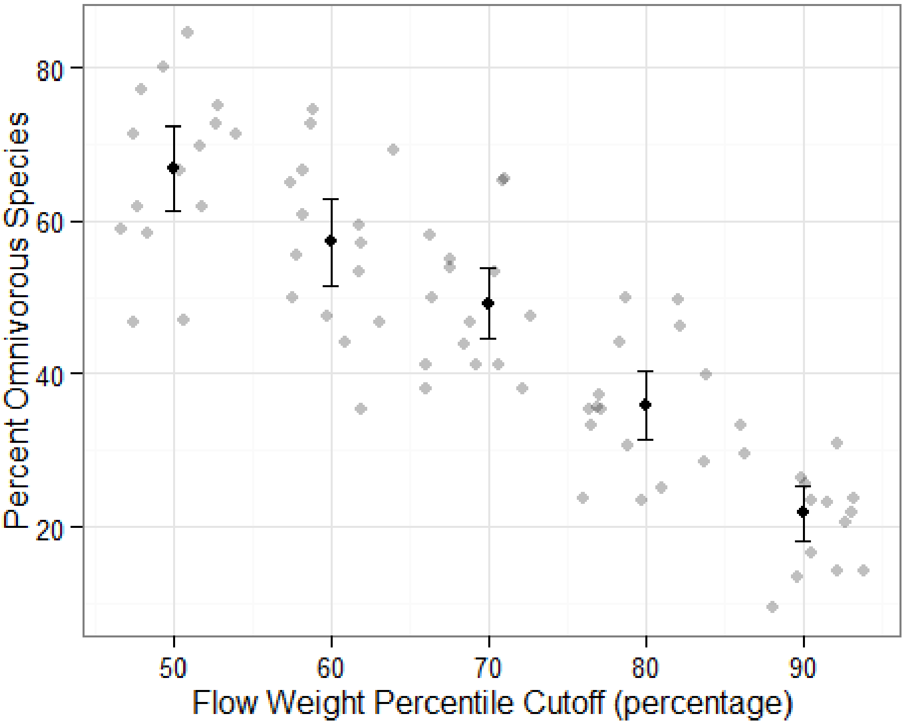


**Figure A1.1** The proportion of omnivorous species in 15 ecosystem flow networks that have been subset by flow weight. Values along the x-axis represent removal of the bottom 50, 60, 70, 80, and 90 percentiles of flow weights. Data for making this figure are available from http://www.cbl.umces.edu/~ulan.

Note that there is still at least 20% omnivory even when only looking at the top 10% of interactions. I say "at least" because this simple analysis does not remove from consideration species who are disconnected from the web (i.e. they do not have any interactions whose flow is above the threshold).

**Code for Chapter 1 simulations**

All code here can also be found on GitHub

Loading required packages

```
library(RCurl)
library(igraph)
library(reshape2)
library(ggplot2)
library(grid)
library(data.table)
library(devtools)
library(NetIndices)
```

**Functions for eigenvalue analysis and food web generation**

The function `analyze.eigen` randomly fills a signed matrix and calculates the eigenvalue with the largest real part. This function takes in a sign matrix (a matrix of +1s, -1s, and 0s) and replaces the $a_{ij}$ with values randomly drawn from predefined uniform distributions. This is the function used for the simple food webs.

```
analyze.eigen<-function(m){
  for(i in 1:nrow(m)){
    for (j in 1:nrow(m)){
      ifelse(m[i,j]==1,m[i,j]<-runif(1,0,10),NA)
      ifelse(m[i,j]==-1,m[i,j]<-runif(1,-1,0),NA)
    }
  }
  for(i in 1:nrow(m)){
    if(m[i,i]<0){m[i,i]<--1}
  }
  ev<-eigen(m)$values[1]
  return(ev)
}
```

The following functions are used for the random and niche model webs.

The `ran.unif` function fills in a sign structured matrix with random values drawn from a random uniform distribution. The impact of the prey on the predator population is drawn from a distribution between 0 and `pred`, while the impact of the predator on the prey is distributed between `prey` and 0.

```
ran.unif <- function(motmat, pred = 10, prey = -1, random = F){
  newmat <- apply(motmat, c(1,2), function(x){
    if(x==1){runif(1, 0, pred)}else if(x==-1){runif(1, prey, 0)} else{0}
  })
  if(random){
    diag(newmat) <- runif(length(diag(newmat)), -1, 0)
  }else{diag(newmat) <- -1}

  return(newmat)
}
```

The maxRE computes the largest eigenvalue and returns the real part.

```
maxRE <- function(rmat){
  lam.max <- eigen(rmat)$values[which.max(Re(eigen(rmat)$values))]
  return(lam.max)
}
```

The eig.analysis function takes an imput of a list of matrices and randomly fills it according to ran.unif (above) n times and computes maxRE for each iteration.

```
eig.analysis <- function(n, matrices, params){
  require(data.table)
  dims <- dim(matrices[[1]])
  cols <- length(matrices)
  rows <- n
  eigenMATRIX.re <- matrix(nrow = rows, ncol = cols)
  eigenMATRIX.im <- matrix(nrow = rows, ncol = cols)
  samps <- list()
  for(i in 1:n){
    ranmat <- lapply(matrices, ran.unif, pred = params[,1],
                     prey = params[,2], random = T)
    sampvals <- matrix(nrow = length(ranmat), ncol = dims[1]^2)
    for(j in 1:length(ranmat)){
      sampvals[j,] <- ranmat[[j]]
    }
    eigs <- sapply(ranmat, maxRE)
    eigenMATRIX.re[i,] <- Re(eigs)
    eigenMATRIX.im[i,] <- Im(eigs)
    samps[[i]] <- as.data.frame(sampvals)
  }
  svals <- cbind(web = rep(1:length(matrices), n), n = rep(1:n, each = cols), rbindlist(samps))
  return(list(samples = svals, ematrix.re = eigenMATRIX.re, ematrix.im = eigenMATRIX.im))
}
```

The conversion function takes in an adjacency matrix and converts it into sign matrix, assuming all interactions are predator/prey (+/-).

```
conversion <- function(tm){
  for(i in 1:nrow(tm)){
    for(j in 1:ncol(tm)){
      if(tm[i,j] == 1){tm[j,i] <- -1}
    }
  }
  return(tm)
}
```

The randomWEBS function generates numweb random webs of S species and total interactions. The connectance of these webs would then be total/S^2. The chain parameter can be modified to change the maximum chain length in the web. For example, the default chain = 9 means that the webs have at least one chain of ten species.

```
randomWEBS <- function(S = 10, numweb = 200, chain = 9, total = 14){
  require(NetIndices)
  require(igraph)
  mywebs <- list()
  for(j in 1:numweb){

    check <- 1
    while(!check == 0){
      myweb <- matrix(0, nrow = S, ncol = S)
      for(i in 1:chain){
        myweb[i,i+1] <- 1
      }
      tophalf <- which(myweb[upper.tri(myweb)] == 0)
      newones <- sample(tophalf, total-chain)
      myweb[upper.tri(myweb)][newones] <- 1
      mywebs[[j]] <- myweb

      indeg <- apply(myweb, 1, sum)
      outdeg <- apply(myweb, 2, sum)
      deg <- indeg + outdeg
      if(sum(deg == 0) >= 1){check <- 1}else{check <- 0}

    }

  }
  return(mywebs)
}
```

The two functions below are the niche model (niche.model), and a function that uses the niche model code to generate a list of niche model food webs with a given connectance C and and number of species S (niche_maker).

```
niche.model<-function(S,C){
  require(igraph)
  connected = FALSE
  while(!connected){
    new.mat<-matrix(0,nrow=S,ncol=S)
    ci<-vector()
    niche<-runif(S,0,1)
    r<-rbeta(S,1,((1/(2*C))-1))*niche

    for(i in 1:S){
      ci[i]<-runif(1,r[i]/2,niche[i])
    }

    r[which(niche==min(niche))]<-.00000001

    for(i in 1:S){

      for(j in 1:S){
        if(niche[j]>(ci[i]-(.5*r[i])) && niche[j]<(ci[i]+.5*r[i])){
          new.mat[j,i]<-1
        }
      }
    }

    new.mat<-new.mat[,order(apply(new.mat,2,sum))]

    connected <- is.connected(graph.adjacency(new.mat))
  }
  return(new.mat)
}

niche_maker <- function(n, S, C){
  niche.list <- list()
  for (i in 1:n){
    niche.list[[i]]<- niche.model(S, C)
  }
  return(niche.list)
}
```

The randomQSS function takes in a list of webs (as adjacency matrices) with the mywebs paramater, and a 1 row by 2 column matrix of parameters to feed into the eig.analysis function. It then outputs a list of two data frames. The first, web.dat contains information on quasi sign-stability, max, mean and median trophic level, standard deviation of trophic level, and the diameter (longest food chain) of the web. The second data frame, iter.dat contains information on each random sampling of each matrix. iter.dat has each sampled value for each link, and the real and imaginary parts of the largest eigenvalue.

```
randomQSS <- function(mywebs, params){
  require(NetIndices)
  require(igraph)

  mywebs1 <- lapply(mywebs, conversion)
  myweb.tl <- lapply(mywebs, TrophInd)
  emat <- eig.analysis(1000, mywebs1, params)

  qss <- apply(emat$ematrix.re, 2, function(x){sum(x<0)/1000})
  maxtl <- sapply(myweb.tl, function(x){max(x$TL)})
  meantl <- sapply(myweb.tl, function(x){mean(x$TL)})
  medtl <- sapply(myweb.tl, function(x){median(x$TL)})
  sdtl <- sapply(myweb.tl, function(x){sd(x$TL)})
  diam <- sapply(lapply(mywebs, graph.adjacency), diameter)

  web.dat <- data.frame(qss, diam, maxtl, meantl, medtl, sdtl)
  iter.dat <- cbind(par = rep(paste(params, collapse = "_"), nrow(emat$samples)),
                    emat$samples, reals = as.vector(emat$ematrix.re),
                    im = as.vector(emat$ematrix.im))
  return(list(web.dat, iter.dat))
}
```

The getQSS function is the main function for the simulation, putting together all previous functions. The webiter, maxchain, and totalINT paramaters are fed into the functions that generate the food webs. The params parameter should be a matrix with two columns, one for the impact of the prey on the predator, and one for the impact of the predator on the prey. This function is designed to run in parallel, and will run a set of webs on each core. The output is written to file according to filepath. This function will generate either random webs or niche model webs according to whether niche is TRUE or FALSE. It calls a separate script robustnessFUNC.R that holds the functions described above and allows each node of the cluster to complete the analysis.

```
getQSS <- function(webiter = 100, maxchain = 9, totalINT = 14, params, filepath, niche = FALSE){
  require(doSNOW)
  require(parallel)
  require(data.table)

  #make the cluster
  cl <- makeCluster(detectCores()-1)
  registerDoSNOW(cl)

  RESULT <- foreach(i = 1:maxchain) %dopar% {
    source("C:/Users/jjborrelli/Desktop/GitHub/Food-Chain-Length/robustnessFUNC.R")

    #cat(i, "\n")
    if(!niche){
      rwebs <- randomWEBS(S = 10, numweb = webiter, chain = i, total = totalINT)
    }else{
      rwebs <- niche_maker(15, 10, totalINT)
    }

    rqss <- list()
    for(j in 1:nrow(params)){
      rqss[[j]] <- randomQSS(mywebs = rwebs, params = params[j,1:2])

      rqss[[j]][[1]] <- cbind(C = rep(totalINT, nrow(rqss[[j]][[1]])),
                                       mxch = rep(i, nrow(rqss[[j]][[1]])),
                               rqss[[j]][[1]],
                              par = rep(paste(params[j,], collapse = "/"),
                                        nrow(rqss[[j]][[1]])))

      rqss[[j]][[2]] <- cbind(C = rep(totalINT, nrow(rqss[[j]][[2]])),
                              mxch = rep(i, nrow(rqss[[j]][[2]])),
                              rqss[[j]][[2]],
                              par = rep(paste(params[j,], collapse = "/"),
                                        nrow(rqss[[j]][[2]])))
      cat("--", j, "\n")
    }
    rqss <- unlist(rqss, recursive = F)
    web.dat.ls <- rbindlist(rqss[seq(1, length(rqss), 2)])
    iter.dat.ls <- rbindlist(rqss[seq(2, length(rqss), 2)])

    return(list(web.dat.ls, iter.dat.ls))

  }

  stopCluster(cl)


  RESULT <- unlist(RESULT, recursive = F)
  chain.data <- rbindlist(RESULT[seq(1, length(RESULT), 2)])
  iter.data <- rbindlist(RESULT[seq(2, length(RESULT), 2)])
  rm(RESULT)
  write.csv(chain.data, file = paste(filepath, "/webdata-",
                                     totalINT, ".csv", sep = ""),
            row.names = F)

  #Code to generate dataframes of sampled matrices (note this can create very large files)
  #write.csv(iter.data, file = paste(filepath, "/iterdata-",
  #                                   totalINT, ".csv", sep = ""),
  #          row.names = F)

  return(chain.data)
}
```

### Code for simulation of food webs

Simple webs

The code below will create the sign matrix structure for each perturbed chain of 2, 3, 4, 5, and 6 levels. A -1 indicates the impact of a predator on its prey (negative), while a 1 indicates the impact of the prey on the predator (positive).

```
sign2<-matrix(c(-1,-1,1,0),nrow=2,ncol=2)
diag(sign2)<--1

sign3<-matrix(c(-1,-1,-1,1,0,-1,1,1,0),nrow=3,ncol=3)
diag(sign3)<--1

sign4<-matrix(nrow=4,ncol=4)
sign4[lower.tri(sign4)]<--1
sign4[upper.tri(sign4)]<-1
diag(sign4)<--1

sign5<-matrix(nrow=5,ncol=5)
sign5[lower.tri(sign5)]<--1
sign5[upper.tri(sign5)]<-1
diag(sign5)<--1

sign6<-matrix(nrow=6,ncol=6)
sign6[lower.tri(sign6)]<--1
sign6[upper.tri(sign6)]<-1
diag(sign6)<--1
```

These matrices are combined into a list for simplicity:

```
sign.matrices<-list(sign2,sign3,sign4,sign5,sign6)
names(sign.matrices)<-c("2 sp","3 sp","4 sp","5 sp","6 sp")
sign.matrices
```

The following code applies the analyze.eigen function to each of the 5 sign matrices created above. It then stores the max(Re(lambda)), or the eigen value with the largest real part in the eigenvalues list. Quasi sign-stability (qss) can then be calculated by determining the proportion of the max(Re(lambda)) that are negative out of the 10000 that are calculated.

```
eigenvalues<-list()
qss<-list()
for(i in 1:5){
  eigenvalues[[i]]<-replicate(10000,analyze.eigen(sign.matrices[[i]]))
  qss[[i]]<-sum(Re(eigenvalues[[i]])<0)/10000
}

names(eigenvalues)<-c("2 sp","3 sp","4 sp","5 sp","6 sp")
names(qss)<-c("2 sp","3 sp","4 sp","5 sp", "6 sp")
qss
```

**Stability analysis of random and niche model webs**

Set the parameters for the simulation, and the desired levels of connectance.

```
pars <- data.frame(pred = c(10, 10, 10, 5, 5, 5, 1, 1, 1), prey = c(-1, -5, -10, -1, -5, -10, -1, -5, -
10))
ints <- c(12, 16, 20, 24, 28)
```

Use the getQSS function to create webs and determine quasi sign-stability for a range of parameters.

**For random webs**

```
for(con in 1:length(ints)){
  getQSS(25, maxchain =7, totalINT = ints[con], params = pars,
         filepath = myfilepath1, niche = F)
  cat(con/length(ints)*100, "%", "\n")
}
```

**For niche model webs**

```
for(con in 1:length(ints)){
  getQSS(25, maxchain =7, totalINT = ints[con]/100, params = pars,
         filepath = myfilepath2, niche = T)
  cat(con/length(ints)*100, "%", "\n")
}
```

**Code for Chapter 1 figures**

Code for Figure 1

Code for Figure 1a

```
path <- getURL("https://raw.githubusercontent.com/jjborrelli/Food-Chain-
Length/master/Tables/webDiameters.csv",
              ssl.verifypeer=0L, followlocation=1L)

web.diameters <- read.csv(text = path, row.names = 1)
```

```
diam.plot <- ggplot(web.diameters, aes(x = Diameter + 1))
diam.plot <- diam.plot + geom_histogram(binwidth = 1)
diam.plot <- diam.plot + theme(axis.title.x = element_text(size = 20))
diam.plot <- diam.plot + theme(axis.title.y = element_text(size = 20))
diam.plot <- diam.plot + theme(axis.text.x = element_text(size = 15))
diam.plot <- diam.plot + theme(axis.text.y = element_text(size = 15))
diam.plot <- diam.plot + scale_y_continuous(name = "Density", limits = c(0,.55))
diam.plot <- diam.plot + scale_x_continuous(name = "Longest Chain Length", breaks = 0:9) +
theme_bw()
```

Code for Figure 1b

```
path2 <- getURL("https://raw.githubusercontent.com/jjborrelli/Food-Chain-
Length/master/Tables/NodeProperties.csv", ssl.verifypeer=0L, followlocation=1L)
trophic.properties <- read.csv(text = path2, row.names = 1)

consumers <- which(round(trophic.properties$TL, 6) >= 2)

# ggplot of distribution of trophic positions equal or higher than 2
tc.plot <- ggplot(trophic.properties[consumers,], aes(x = TL)) + theme_bw()
tc.plot <- tc.plot + geom_histogram(binwidth = 1) + xlab("Trophic Position") + ylab("Density")
tc.plot <- tc.plot + scale_x_continuous(limits = c(2,6),name = "Trophic Position") +
scale_y_continuous(name = "Density", limits = c(0,.55))
tc.plot
```

Code for Figure 2

The simple web matrices can be visualized with the following code (note: requires the igraph library):

But first the sign matrices need to be converted to graph objects

```
graph.chains<-lapply(sign.matrices, graph.adjacency)
```

The layout is defined for each node of each chain:

```
twospec2<-matrix(c(1,1,
                 2,2),nrow=2,ncol=2,byrow=T)
threespec2<-matrix(c(1,1,
                   3,1,
                   2,2),nrow=3,ncol=2,byrow=T)
```

```
fourspec2<-matrix(c(1,1,
                    2,2,
                    0,2,
                    1,3),nrow=4,ncol=2,byrow=T)
fivespec2<-matrix(c(2,1,
                    3,2,
                    1,2,
                    3,3,
                    1,3),nrow=5,ncol=2,byrow=T)
sixspec2<-matrix(c(2,1,
                   3,2,
                   1,2,
                   3,3,
                   1,3,
                   2,4),nrow=6,ncol=2,byrow=T)

layouts<-list(twospec2,threespec2,fourspec2,fivespec2,sixspec2)
```

Setting the plotting options to highlight the longest chain in each web:

```
for(i in 1:5){
  E(graph.chains[[i]])$color = "darkslategray4"
  E(graph.chains[[i]], path = c(1:(i+1)))$color = "black"
}
```

Create the plot

```
par(mfrow=c(5, 1),mar=c(.1, .1, .1, .1))
for(i in 1:5){
  plot.igraph(graph.chains[[i]], layout = layouts[[i]],
              vertex.size = 40,
              vertex.color = "white",
              vertex.label.color = "black",
              vertex.label.cex = .8,
              edge.width = 1,
              edge.arrow.size = .35,
              frame = T,
              margin = 0)
}
```

Code for Figure 3

```
qss.plot <- qplot(2:6, unlist(qss), xlab = "Longest Chain Length", ylab = "Quasi Sign-Stability",
margin = T)
qss.plot <- qss.plot + geom_point(size = 2)
qss.plot <- qss.plot + geom_line() + theme_bw()
qss.plot
```

Code for Figure 4

First import the data from the random webs

```
web.files <- list.files(path = filepath1, pattern = "web")
temp.ls <- list()
for(i in 1:length(web.files)){
  temp.ls[[i]]<- fread(paste(filepath1,
                              web.files[i], sep = ""))
}
webdata <- rbindlist(temp.ls)
webdata$C <- factor(webdata$C)

# Formulas for upper and lower confidence limits
sem.l <- function(x){mean(x) - 1.96*sqrt(var(x)/length(x))}
sem.u <- function(x){mean(x) + 1.96*sqrt(var(x)/length(x))}

sub1 <- subset(webdata, par == "1/-10" | par == "10/-1" | par == "1/-5" | par == "5/-1")
sub2 <- subset(webdata, par == "5/-10" | par == "10/-5")
sub3 <- subset(webdata, par == "1/-1" | par == "10/-10" | par == "5/-5")
```

Plots of the different subsets of data. The final plot is what appears in the paper.

```
ggplot(sub1, aes(x = factor(diam+1), y = qss)) +
  geom_point(alpha = .25, position = position_jitter(w=0.2), col = "grey58") +
  stat_summary(fun.y="mean", geom="point") +
  stat_summary(fun.ymin = sem.l, fun.y = "mean", fun.ymax = sem.u,
               geom="errorbar", width = .2) +
  facet_grid(par~C) + theme_bw() +
  xlab("Longest Food Chain Length") + ylab("Quasi sign-stability")


ggplot(sub2, aes(x = factor(diam+1), y = qss)) +
  geom_point(alpha = .25, position = position_jitter(w=0.2), col = "grey58") +
  stat_summary(fun.y="mean", geom="point") +
  stat_summary(fun.ymin = sem.l, fun.y = "mean", fun.ymax = sem.u,
               geom="errorbar", width = .2) +
  facet_grid(par~C) + theme_bw() +
  xlab("Longest Food Chain Length") + ylab("Quasi sign-stability")


ggplot(sub3, aes(x = factor(diam+1), y = qss)) +
  geom_point(alpha = .25, position = position_jitter(w=0.2), col = "grey58") +
  stat_summary(fun.y="mean", geom="point") +
  stat_summary(fun.ymin = sem.l, fun.y = "mean", fun.ymax = sem.u,
               geom="errorbar", width = .2) +
  facet_grid(par~C) + theme_bw() +
  xlab("Longest Food Chain Length") + ylab("Quasi sign-stability")


ggplot(webdata, aes(x = factor(diam+1), y = qss)) +
  geom_point(alpha = .25, position = position_jitter(w=0.2), col = "grey58") +
  stat_summary(fun.y="mean", geom="point") +
  stat_summary(fun.ymin = sem.l, fun.y = "mean", fun.ymax = sem.u,
               geom="errorbar", width = .2) +
  facet_grid(par~C) + theme_bw() +
  xlab("Longest Food Chain Length") + ylab("Quasi sign-stability")
```

Code for Figure 4

First import the data from the niche model webs

```
web.files <- list.files(path = filepath2, pattern = "web")
temp.ls <- list()
for(i in 1:length(web.files)){
  temp.ls[[i]]<- fread(paste(filepath2,
                             web.files[i], sep = ""))
}
webdata <- rbindlist(temp.ls)
webdata$C <- factor(webdata$C)

# Formulas for upper and lower confidence limits
sem.l <- function(x){mean(x) - 1.96*sqrt(var(x)/length(x))}
sem.u <- function(x){mean(x) + 1.96*sqrt(var(x)/length(x))}
```

```
sub1 <- subset(webdata, par == "1/-10" | par == "10/-1" | par == "1/-5" | par == "5/-1")
sub2 <- subset(webdata, par == "5/-10" | par == "10/-5")
sub3 <- subset(webdata, par == "1/-1" | par == "10/-10" | par == "5/-5")
```

Plots of the different subsets of data. The final plot is what appears in the paper.

```
ggplot(sub1, aes(x = factor(diam+1), y = qss)) +
 geom_point(alpha = .25, position = position_jitter(w=0.2), col = "grey58") +
 stat_summary(fun.y="mean", geom="point") +
 stat_summary(fun.ymin = sem.l, fun.y = "mean", fun.ymax = sem.u,
              geom="errorbar", width = .2) +
 facet_grid(par~C) + theme_bw() +
 xlab("Longest Food Chain Length") + ylab("Quasi sign-stability")
```

```
ggplot(sub2, aes(x = factor(diam+1), y = qss)) +
 geom_point(alpha = .25, position = position_jitter(w=0.2), col = "grey58") +
 stat_summary(fun.y="mean", geom="point") +
 stat_summary(fun.ymin = sem.l, fun.y = "mean", fun.ymax = sem.u,
              geom="errorbar", width = .2) +
 facet_grid(par~C) + theme_bw() +
 xlab("Longest Food Chain Length") + ylab("Quasi sign-stability")
```

```
ggplot(sub3, aes(x = factor(diam+1), y = qss)) +
 geom_point(alpha = .25, position = position_jitter(w=0.2), col = "grey58") +
 stat_summary(fun.y="mean", geom="point") +
 stat_summary(fun.ymin = sem.l, fun.y = "mean", fun.ymax = sem.u,
              geom="errorbar", width = .2) +
 facet_grid(par~C) + theme_bw() +
 xlab("Longest Food Chain Length") + ylab("Quasi sign-stability")
```

```
ggplot(webdata, aes(x = factor(diam+1), y = qss)) +
 geom_point(alpha = .25, position = position_jitter(w=0.2), col = "grey58") +
 stat_summary(fun.y="mean", geom="point") +
 stat_summary(fun.ymin = sem.l, fun.y = "mean", fun.ymax = sem.u,
              geom="errorbar", width = .2) +
 facet_grid(par~C) + theme_bw() +
 xlab("Longest Food Chain Length") + ylab("Quasi sign-stability")
```

# Appendix 2: Annotated code for Chapter 2 simulations

### Generating a Subgraph Library

The following code defines the sign matrix for each of the thirteen possible three-node subgraphs. Here, the "s" in the object name indicates that only single links are used, while a "d" indicates the presence of double links.

```
s1 <- matrix(c(0, 1, 0, -1, 0, 1, 0, -1, 0), nrow = 3, ncol = 3)
s2 <- matrix(c(0, 1, 1, -1, 0, 1, -1, -1, 0), nrow = 3, ncol = 3)
s3 <- matrix(c(0, 1, -1, -1, 0, 1, 1, -1, 0), nrow = 3, ncol = 3)
s4 <- matrix(c(0, 1, 1, -1, 0, 0, -1, 0, 0), nrow = 3, ncol = 3)
s5 <- matrix(c(0, 0, 1, 0, 0, 1, -1, -1, 0), nrow = 3, ncol = 3)

d1 <- matrix(c(0, 1, 1, -1, 0, 1, -1, 1, 0), nrow = 3, ncol = 3)
d2 <- matrix(c(0, 1, 1, 1, 0, 1, -1, -1, 0), nrow = 3, ncol = 3)
d3 <- matrix(c(0, 1, -1, 1, 0, 0, 1, 0, 0), nrow = 3, ncol = 3)
d4 <- matrix(c(0, 1, 1, -1, 0, 0, 1, 0, 0), nrow = 3, ncol = 3)
d5 <- matrix(c(0, 1, 1, -1, 0, 1, 1, -1, 0), nrow = 3, ncol = 3)
d6 <- matrix(c(0, 1, 1, 1, 0, 1, 1, 1, 0), nrow = 3, ncol = 3)
d7 <- matrix(c(0, 1, 1, 1, 0, 1, 1, -1, 0), nrow = 3, ncol = 3)
d8 <- matrix(c(0, 1, 1, 1, 0, 0, 1, 0, 0), nrow = 3, ncol = 3)

mot.lst <- list(s1, s2, s3, s4, s5, d1, d2, d3, d4, d5, d6, d7, d8)
names(mot.lst) <- c("s1", "s2", "s3", "s4", "s5", "d1", "d2", "d3", "d4", "d5",
    "d6", "d7", "d8")
```

### Code to define required functions

Functions for counting motifs

The `motif_counter` function takes in a list of graph objects and applies `triad.census` to each. It returns a data frame of the frequency of each connected three-node digraph.

```
motif_counter <- function(graph.lists) {
    require(igraph)

    if (!is.list(graph.lists)) {
        stop("The input should be a list of graph objects")
    }

    triad.count <- lapply(graph.lists, triad.census)
    triad.matrix <- matrix(unlist(triad.count), nrow = length(graph.lists),
        ncol = 16, byrow = T)
    colnames(triad.matrix) <- c("empty", "single", "mutual", "s5", "s4", "s1",
        "d4", "d3", "s2", "s3", "d8", "d2", "d1", "d5", "d7", "d6")

    triad.df <- as.data.frame(triad.matrix)

    motif.data.frame <- data.frame(s1 = triad.df$s1, s2 = triad.df$s2, s3 = triad.df$s3,
        s4 = triad.df$s4, s5 = triad.df$s5, d1 = triad.df$d1, d2 = triad.df$d2,
        d3 = triad.df$d3, d4 = triad.df$d4, d5 = triad.df$d5, d6 = triad.df$d6,
        d7 = triad.df$d7, d8 = triad.df$d8)

    rownames(motif.data.frame) <- names(graph.lists)
    return(motif.data.frame)
}
```

The Curveball algorithm is available as supplemental information as part of the original publication. *Note if you want to use this code please cite the paper in which it was introduced:*

Strona, G. et al. 2014. A fast and unbiased procedure to randomize ecological binary matrices with fixed row and column totals. -Nat. Comm. 5: 4114. doi: 10.1038/ncomms5114

Their function takes a matrix and makes a swap (process descirbed in their paper) and returns the new matrix.

```
curve_ball <- function(m) {
    RC = dim(m)
    R = RC[1]
    C = RC[2]
    hp = list()
    for (row in 1:dim(m)[1]) {
        hp[[row]] = (which(m[row, ] == 1))
    }
    l_hp = length(hp)
    for (rep in 1:(5 * l_hp)) {
        AB = sample(1:l_hp, 2)
        a = hp[[AB[1]]]
        b = hp[[AB[2]]]
        ab = intersect(a, b)
        l_ab = length(ab)
        l_a = length(a)
        l_b = length(b)
        if ((l_ab %in% c(l_a, l_b)) == F) {
            tot = setdiff(c(a, b), ab)
            l_tot = length(tot)
            tot = sample(tot, l_tot, replace = FALSE, prob = NULL)
            L = l_a - l_ab
            hp[[AB[1]]] = c(ab, tot[1:L])
            hp[[AB[2]]] = c(ab, tot[(L + 1):l_tot])
        }

    }
    rm = matrix(0, R, C)
    for (row in 1:R) {
        rm[row, hp[[row]]] = 1
    }
    rm
}
```

The curving function is used to iteratively apply the Curveball algorithm to a single matrix. This function takes an adjacency matrix and number of iterations as inputs and returns a dataframe of motif frequencies.

```
curving <- function(adjmat, n) {
    mot <- motif_counter(list(graph.adjacency(adjmat)))
    newmat <- adjmat

    for (i in 1:n) {
        newmat <- curve_ball(newmat)
        m <- motif_counter(list(graph.adjacency(newmat)))
        mot <- rbind(mot, m)
    }
    return(mot[-1, ])
}
```

I added the additional constraints of maintaining the number of single, double, and self links in each matrix to the original curveball algorithm in the function dblcan.curve. This function takes a matrix and desired number of iterations as inputs and returns a dataframe of motif frequencies.

```
library(plyr)

nd <- function(gl) nrow(gl) - nrow(unique(aaply(gl, 1, sort)))
# determines the number of double links
```

```
dblcan.curve <- function(mat, iter) {
    mot <- motif_counter(list(graph.adjacency(mat)))

    el <- get.edgelist(graph.adjacency(mat))
    Ne <- nrow(el)
    dbl <- nd(el)
    can <- sum(diag(mat))

    for (i in 1:iter) {
        ed = TRUE
        dub = TRUE
        ca = TRUE
        while (ed || dub || ca) {
            mat2 <- curve_ball(mat)

            el2 <- get.edgelist(graph.adjacency(mat2))
            el3 <- unique(el2)
            Ne2 <- nrow(el3)
            dbl2 <- nd(el3)
            can2 <- sum(diag(mat2))

            ed <- Ne != Ne2
            dub <- dbl != dbl2
            ca <- can != can2

        }
        mat <- mat2

        mot <- rbind(mot, motif_counter(list(graph.adjacency(mat))))
    }
    return(M = mot[-1, ])
}
```

Functions for determining quasi sign-stability

There are two main functions for determining quasi sign-stability, and a third that wraps them together to generate the desired number of iterations.

The function ran.unif takes an input of a signed matrix. It will then check each cell to see if there is a 1 or -1. Each 1 will be replaced by a value drawn from the random uniform distribution between 0 and 10, while each -1 is replaced by a value from the random uniform distribution between -1 and 0. The ran.unif function also assigns values to the diagonal from a random uniform distribition between -1 and 0. The resulting randomly sample matrix is returned.

```
ran.unif <- function(motmat) {
    newmat <- apply(motmat, c(1, 2), function(x) {
        if (x == 1) {
            runif(1, 0, 10)
        } else if (x == -1) {
            runif(1, -1, 0)
        } else {
            0
        }
    })
    diag(newmat) <- runif(3, -1, 0)
    return(newmat)
}
```

Given the input matrix maxRE will compute the eigenvalues and return the largest real part.

```
maxRE <- function(rmat) {
    lam.max <- max(Re(eigen(rmat)$values))
    return(lam.max)
}
```

The above two functions are combined in eig.analysis. Given the number of desired sampling iterations, n, and a list of sign matrices to analyze, matrices, the eig.analysis function will return an n by length(matrices) matrix of eigenvalues. Specifically it is returning the max(Re(λ)) for each sampled matrix. From this matrix quasi sign-stability can be calculated as the proportion of values in each column that are negative.

```
eig.analysis <- function(n, matrices) {
    cols <- length(matrices)
    rows <- n
    eigenMATRIX <- matrix(0, nrow = rows, ncol = cols)
    for (i in 1:n) {
        ranmat <- lapply(matrices, ran.unif)
        eigs <- sapply(ranmat, maxRE)
        eigenMATRIX[i, ] <- eigs
    }
```

```
  return(eigenMATRIX)
}
```

### Code for motif counting

Load required packages

```
library(igraph)
library(ggplot2)
library(reshape2)
library(parallel)
library(doSNOW)
```

Determining motif frequency

Load in web data from GitHub. Click here to download the .Rdata file. This file is a list of igraph graph objects for each of the 50 webs used in the analysis. Once you have downloaded the file into your working directory:

```
load(paste(getwd(), "webGRAPHS.Rdata", sep = "/"))
```

The frequencies of each of the different subgraphs can now be determined easily with motif_counter.

```
motfreq <- motif_counter(web.graphs)
kable(motfreq, format = "pandoc")
```

The following code runs the null model analysis for the 50 food webs. First, each of the fifty webs are converted into binary adjacency matrices (web.matrices).

```
web.matrices <- lapply(web.graphs, get.adjacency, sparse = F)
```

The first null model, the Curveball algorithm can be applied to all 50 webs in parallel. This code starts by registering a cluster utilizing 1 less than the number of cores on the computer (for this paper it was 7 cores). It then creates set of 30000 matrices and returns a list (length equal to the number of webs, 50 in this case) of dataframes of motif frequences.

```
cl <- makeCluster(detectCores() - 1)
clusterExport(cl, c("web.adj", "motif_counter", "curve_ball", "curving"))
registerDoSNOW(cl)
randos <- parLapply(cl, web.adj, curving, n = 30000)
stopCluster(cl)
```

Once subgraph counts have been obtained, the mean and standard deviation for each subgraph are computed. Z-scores are then computed as described in the methods section:

$$z_i = \frac{x_i - \bar{x}}{\sigma}$$

The normalized profile was then computed (as desribed in the methods):

$$n_i = \frac{z_i}{\sqrt{\sum z_j^2}}$$

```
means <- t(sapply(randos, colMeans))
stdevs <- t(sapply(randos, function(x) {
   apply(x, 2, sd)
}))

motfreq <- motif_counter(web.graphs)

zscore <- (motfreq - means)/stdevs

# Normalized z-scores
zscore.norm <- t(apply(zscore, 1, function(x) {
   x/sqrt(sum(x^2, na.rm = T))
}))
```

Below is the code used to run the Curveball algorithm with the additional constraints described above. *Note this can take a very long time to run*

```
# Use the filepath.sink variable to set the path for storing the sink
# information
filepath.sink <- my.sink.path
# Use the filepath.data variable to set the location for storing the
# dataframes of motif frequencies for each web
filepath.data <- my.data.path

cl <- makeCluster(detectCores() - 1)
clusterExport(cl, c("web.adj", "motif_counter", "curve_ball", "nd", "aaply",
   "filepath"))
registerDoSNOW(cl)
```

```
system.time(randos.t3 <- foreach(i = 1:length(web.adj)) %dopar% {
  sink(file = paste(filepath.sink, names(web.adj[i]), ".txt", collapse = ""))
  motfreq <- dblcan.curve(web.adj[[i]], iter = 30000)
  print(motfreq)
  sink()
  write.csv(motfreq, file = paste(filepath.data, names(web.adj[i]), ".csv",
    collapse = ""))
  return(motfreq)
})
stopCluster(cl)
```

The normalized z-score profile can be calculated the same as above:

```
means.t <- t(sapply(randos.t3, colMeans))
stdevs.t <- t(sapply(randos.t3, function(x) {
  apply(x, 2, sd)
}))

motfreq <- motif_counter(web.graphs)

zscore.t <- (motfreq - means.t)/stdevs.t

# Normalized z-scores
zscore.N <- t(apply(zscore.t, 1, function(x) {
  x/sqrt(sum(x^2, na.rm = T))
}))
```

Figure 1 is then a boxplot of the above normalized z-scores, reordered according to decreasing quasi sign-stability (see below). When using the modified Curveball algorithm, those webs that have no double links produce NaN when computing the z-score for those motifs. These are ignored in Figure 1, and the z-scores presented are with the NaNs removed.

**Code for determining quasi sign-stability**

The first step to get quasi sign stability is to get the largest eigenvalues from a series of randomly parameterized sign matrices. In the following code I generate 10000 random parameterizations for each of the 13 subgraphs's sign matrices (mot.lst). The eig.analysis function will return a matrix where each column is a different subgraph and each row is the largest eigenvalue of a particular randomization.

```
set.seed(5)
```

```
n <- 10000
mot.stab <- eig.analysis(n, mot.lst)
colnames(mot.stab) <- names(mot.lst)
```

From that matrix, quasi sign-stability is calculated as the proportion of rows with a negative value. In other words, how many random parameterizations of the sign matrix were locally stable?

```
mot.qss <- apply(mot.stab, 2, function(x) {
  sum(x < 0)/n
})
sorted <- sort(mot.qss, decreasing = T)
sorted
```

```
    s1     s4     s5     s2     d3     d4     s3     d2     d1     d5
1.0000 1.0000 1.0000 0.5345 0.0891 0.0866 0.0561 0.0428 0.0370 0.0101
    d7     d6     d8
0.0021 0.0000 0.0000
```

### Code to determine the robustness to assumption of double link positives

I assumed that if there was a double link in the subgraph, that both the effect of the predator on the prey and the effect of the prey on the predator were positive, a (+/+) rather than a (+/-). Here I repeat the analysis described above, but instead assuming that the relative effects correspond to a (-/-).

```
mot.lst2 <- lapply(mot.lst, function(x) {
  for (i in 1:nrow(x)) {
    for (j in 1:ncol(x)) {
      if (x[i, j] == 1 && x[j, i] == 1) {
        x[i, j] <- x[i, j] * -1
        x[j, i] <- x[j, i] * -1
      } else {
        next
      }
    }
  }
  return(x)
})

set.seed(15)
```

```
n <- 10000
mot.stab2 <- eig.analysis(n, mot.lst2)
colnames(mot.stab2) <- names(mot.lst2)

mot.qss2 <- apply(mot.stab2, 2, function(x) {
    sum(x < 0)/n
})
sorted2 <- sort(mot.qss2, decreasing = T)
sorted2
```

```
    s1     s4     s5     d4     d3     s2     d2     d1     d7     d5 
1.0000 1.0000 1.0000 0.8924 0.8914 0.5339 0.5042 0.5038 0.4890 0.3946 
    d6     d8     s3 
0.2440 0.2163 0.0558 
```

**Code to determine robustness to assumption of matrix fill distributions**

I also tested how quasi sign-stability of the different subgraphs changed when varying the assumption of the relative impact of the predator on its prey and the prey on its predator.

I tested 7 additional distributions (all uniform) with different magnitudes.

```
params.u <- data.frame(pred1 = c(0, 0, 0, 0, 0, 0, 0, 0), pred2 = c(10, 10,
    10, 10, 10, 5, 3, 1), prey1 = c(-10, -5, -1, -0.1, -0.01, -1, -1, -1), prey2 = c(0,
    0, 0, 0, 0, 0, 0, 0))
parvals <- factor(paste(params.u[, 2], params.u[, 3], sep = "/"), levels = c("1/-1",
    "3/-1", "5/-1", "10/-1", "10/-0.01", "10/-0.1", "10/-1", "10/-5", "10/-10"))
```

Below is a function to take in the different parameters for the distributions and fill a matrix accordingly.

```
eigen_unif <- function(m, iter, params, self = -1) {
    # For when I want to use uniform distribution Params is dataframe of min and
    # max for relative impact of prey on pred and pred on prey
    ev <- c()
    for (i in 1:iter) {
        m1 <- apply(m, c(1, 2), function(x) {
            if (x == 1) {
                runif(1, params$pred1, params$pred2)
            } else if (x == -1) {
                runif(1, params$prey1, params$prey2)
            } else {
                0
            }
        })
        diag(m1) <- self
        ev[i] <- max(Re(eigen(m1)$values))
    }
    return(ev)
}
```

The following code loops through the dataframe of distributions and computes quasi sign-stability

```
test <- matrix(nrow = nrow(params.u), ncol = length(mot.lst))
for (i in 1:nrow(params.u)) {
    eigen.test <- lapply(mot.lst, eigen_unif, iter = 10000, params = params.u[i,
        ], self = runif(3, -1, 0))
    qss.test <- lapply(eigen.test, function(x) {
        sum(x < 0)/length(x)
    })
    test[i, ] <- unlist(qss.test)
}
```

### Code for Figure A2.1

```
colnames(test) <- c("s1", "s2", "s3", "s4", "s5", "d1", "d2", "d3", "d4", "d5",
    "d6", "d7", "d8")
test <- data.frame(test, parvals)
dat1 <- melt(test[, c(names(sorted), "parvals")])

ggplot(dat1, aes(x = variable, y = value)) + geom_point() + facet_wrap(~parvals,
    ncol = 4) + theme_bw() + xlab("Subgraph") + ylab("Quasi Sign-Stability")
```

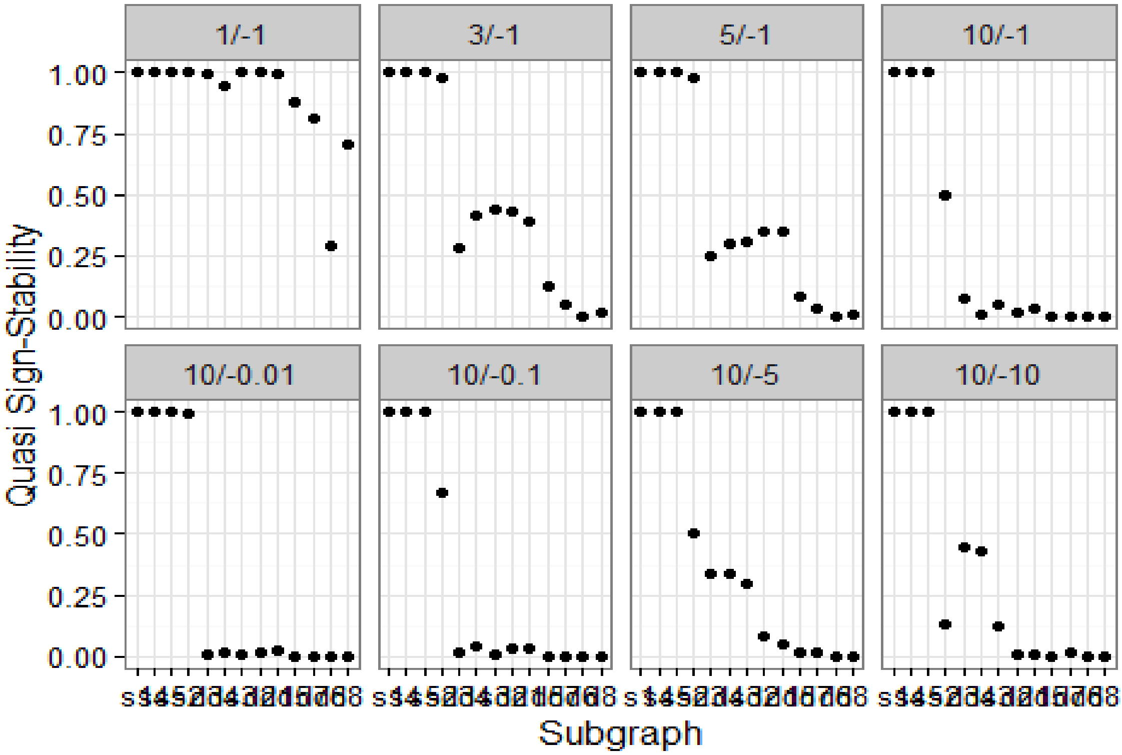


**Figure A2.1:** Robustness of quasi sign-stability to different distributions used to fill the random Jacobian matrix. Facet labels represent the maximum impact of the prey on the predator population (positive number) and maximum impact of the predator on the prey population (negative number)

**Code for the figures**

Code for Figure 1

```
par(mfrow = c(1, 13), mar = c(0.2, 0.2, 0.2, 0.2))
for (i in 1:13) {
  plot.igraph(graph.adjacency(mot.lst[[which(names(mot.lst) == names(sorted[i]))]]),
    layout = layout.circle, edge.arrow.size = 0.5, vertex.size = 30, vertex.color = "black",
    vertex.label = NA, frame = T, edge.width = 2, edge.color = "darkslategray4")
  text(-0.25, 0, names(sorted[i]), cex = 1.5)
}
```

Code for Figure 2

```
z1 <- cbind(Model = factor("Curveball"), melt(zscore.norm[, names(sorted)]))
z2 <- cbind(Model = factor("Double"), melt(zscore.N[, names(sorted)]))

z.both <- rbind(z1, z2)
ggplot(z.both, aes(x = Var2, y = value, fill = Model)) + geom_boxplot() + xlab("Subgraph") +
    ylab("Normalized Profile") + theme_bw()
```

```
Warning: Removed 308 rows containing non-finite values (stat_boxplot).
```

Code for Figure 3

```
sort.df <- melt(sorted)

qssplot <- ggplot(sort.df, aes(x = 1:13, y = value)) + geom_point(shape = 19,
    size = 3) + theme_bw()
qssplot + xlab("Subgraph") + ylab("Quasi Sign-Stability") + scale_x_discrete(limits =
names(sorted))
```

# Appendix 3: Annotated code for the *rend* package

The following is the code developed for the `rend` R package. To install and use this package simply run:

```
devtools::install_github("jjborrelli/rend")
```

This package is modular in nature, and centered around a single simulation function, `CRsimulator`. The only required user input for this function is the adjacency matrix (`Adj`), a binary representation of directed links, of the food web. All of the other parameters can remain as default, or be altered by the user. The `states` argument is an optional input (i.e., may be left null) of the initial biomasses for all species in the system. The `t` argument is the sequence of time steps. G is the function to be used for basal species growth. The `method` argument gives the function to be integrated with `deSolve::ode`. `FuncRes` is the functional response function, which in this package can be either a Holling (`Fij`) or Beddington-DeAngelis (`Fbd`; consumer interference). The `K`, `x.i`, `yij`, and `eij` arguments are all parameters of the bioenergetic model, they default to values set by Williams and Martinez (2005). The argument `xpar` is the tuning parameter for the functional response, either *q* for `Fij` or *c* for `Fbd`. `B.o` is the half saturation constant. The `r` argument, like `states`, is an optional argument of growth rates. If left null the function will assign a growth rate of 1 to all species that lack prey. The `ext` argument is the extinction function to be used to determine when species go extinct. The last argument defines whether a plot (using `matplot`) of biomasses against time should be produced following the simulation.

```
CRsimulator <- function(Adj, states = NULL, t = 1:200, G = Gi, method = CRmod, FuncRes = F
ij, K = 1, x.i = .5, yij = 6, eij = 1, xpar = .2, B.o =.5, r = NULL, ext = goExtinct, plot = FALSE){

 if(is.null(r)){grow <- getR(Adj)}else{grow <- r}

 par <- list(
  K = K,
  x.i = x.i,
  yij = yij,
  eij = 1,
  xpar = xpar,
  B.o = B.o,
  r.i = grow,
  A = Adj,
  G.i = G,
  FR = FuncRes
 )

 if(is.null(states)){states <- runif(nrow(Adj), .5, 1)}

 out <- deSolve::ode(y=states, times=t, func=method, parms=par, events = list(func = ext, time
= t))

  if(plot) print(matplot(out[,-1], typ = "l", lwd = 2, xlab = "Time", ylab = "Biomass"))

  return(out)
}
```

The function getR is applied when the r argument above is left NULL. It simply takes the adjacency matrix and outputs a vector of 1s and 0s, where a 1 indicates the growth rate of basal species. A species is considered basal if it has no prey, and so r should be supplied when there is a scenario including predators with no prey. If there are no basal species the function will warn the user, but will not stop the simulation.

```
getR <- function(amat){
  r.i <- c()
  r.i[colSums(amat) == 0] <- 1
  r.i[colSums(amat) != 0] <- 0

  if(sum(r.i) == 0){
    warning("No basal species in simulation")
  }
  return(r.i)
}
```

The basal growth function Gi is simply exponential growth with density dependence. Its arguments are the vector of growth rates (r), the vector of current biomasses (B), and the carrying capacity (K).

```
Gi <- function(r, B, K){return(r * B * (1 - (B/K)))}
```

There are two options for a functional response; one based on Holling's Type II and III (Fij), and another based on Beddington and DeAngelis' consumer interference functional response (Fbd). Both functions take the same set of parameters, the vector of current biomasses (B), the adjacency matrix (A), the half saturation constant (B.0), and the tuning parameter (xpar).

```
Fij <- function(B, A, B.0, xpar){
  sum.bk <- rowSums(sapply(1:nrow(A), function(x){B[x] * A[x,]}))^(1+xpar)
  denom <- sum.bk + B.0^(1+xpar)

  F1 <- sapply(1:nrow(A), function(x){(B[x] * A[x,])^(1+xpar)})/denom

  return(F1)
}

Fbd <- function(B, A, B.0, xpar){
  sum.bk <- rowSums(sapply(1:nrow(A), function(x){B[x] * A[x,]}))
  denom <- sum.bk + (1 + (xpar * B)) * B.0

  F1 <- sapply(1:nrow(A), function(x){(B[x] * A[x,])})/denom

  return(F1)
}
```

CRmod is the function describing the bioenergetic model, and is used as input to deSolve::ode. The inputs to the function are the sequence of time steps, the initial values for the state variables (in this case the biomasses of each species), and a list of named parameter values. To make this model more flexible, included in the parameter list are all of the specific functions used for growth and consumption, so that user defined functions may be used as well as those provided in the package.

```
CRmod <- function(t,states,par){

  with(as.list(c(states, par)), {
    dB <- G.i(r = r.i, B = states, K = K) -                              # growth
      x.i*states +                                                       # death
      rowSums((x.i * yij * FR(states, A, B.o, xpar = xpar) * states)) -       # consumption
      rowSums((x.i * yij * t(FR(states, A, B.o, xpar = xpar)* states))/eij)   # death by predation

    list(c(dB))
  })

}
```

The goExtinct function checks each biomass to see if it is below the threshold of $10^{-10}$. If it is below the threshold, then that species' biomass is reset to 0, simulating extinction. This function is also supplied as an event function argument to deSolve::ode. The times argument allows the function to be evaluated at each time step.

```
goExtinct <- function(times, states, parms){
  with(as.list(states), {
    for(i in 1:length(states)){
      if(states[i] < 10^-10){states[i] <- 0}else{states[i]}
    }
    return(c(states))
  })
}
```

The rend package also has a function for the visualization of food web dynamics. With the initial adjacency matrix (mat), and the output of the CRsimulator function (dyn) this function computes both the species that remain extant at each time step, and the strength of their interactions.

Interaction strength is computed by the functional response at each time step, which requires the additional arguments of FR for the functional repsonse and xpar.

```
netHTML <- function(mat, dyn, FR = Fij, xpar = .2, path1 = getwd()){
  if(!requireNamespace("animation", quietly = TRUE)){stop("This function requires the 'animation' package to be installed and loaded", call. = FALSE)}

  lay <- matrix(c(layout.sphere(graph.adjacency(mat))[,1], TrophInd(mat)$TL), ncol = 2)
  s <- matrix(0, nrow = nrow(dyn), ncol = ncol(mat))

  ani.options(interval = .25)
  saveHTML(
    {
      for(i in 1:50){
        fr <- FR(dyn[i,-1], mat, .5, xpar)
        strength <- melt(fr)[,3][melt(fr)[,3] > 0]
        fr[fr > 0 ] <- 1

        g.new <- graph.adjacency(t(fr))
        E(g.new)$weight <- strength/max(strength)*10
        s[i,c(which(dyn[i,-1] > 0))] <-log(dyn[i, c(which(dyn[i,] > 0)[-1])])+abs(min(log(dyn[i, c(which(dyn[i,] > 0)[-1])])))

        plot.igraph(g.new, vertex.size = s[i,], edge.width = E(g.new)$weight, layout = lay)
      }
    },
    img.name = paste(path1, "fwdyn", sep = ""), htmlfile = paste(path1, "fwdyn.html", sep = ""),
    interval = .25, nmax =500, ani.width = 500, ani.height = 500, outdir = path1
  )
}
```

There are three functions for computing the change in food webs over time during the simulation: WEBind, motifCounter3, and trophicChange. All three functions take in the initial food web adjacency matrix, web, and the output of the CRsimulator function (dyn) as arguments. Each function then generates a list of adjacency matrices, one for each time step, and then computes the respective network properties.

WEBind returns, for each time step, number of species, number of links, link density, connectance, diameter, average path length, clustering coefficient, modularity, and number of modules. The motifCounter3 returns the frequency of thirteen three-species modules for each time step. Finally, trophicChangereturns the computed trophic position of each species at every time step.

```
WEBind <- function(dyn, web){
  if(!requireNamespace("rnetcarto", quietly = TRUE)){
    stop("This function requires the 'rnetcarto' package to be installed and loaded", call. = FALSE
)
    }

  adj.list <- lapply(1:nrow(dyn), function(x){web[dyn[x, -1] > 0, dyn[x, -1] > 0]})
  g.list <- lapply(adj.list, graph.adjacency)

  # Number of species
  N <- sapply(adj.list, nrow)
  # Number of links
  Ltot <- sapply(adj.list, sum)
  # Link Density
  LD <- sapply(adj.list, function(x) sum(x)/nrow(x))
  # Connectance: Links / (N * (N - 1))
  C <- sapply(adj.list, function(x){sum(x)/(nrow(x) * (nrow(x) - 1))})
  # Web diameter
  D <- sapply(g.list, diameter)
  # Average path length
  APL <- sapply(g.list, average.path.length)
  # Clustering coefficient
  CC <- sapply(g.list, transitivity)
  # Modularity
  mod <- lapply(lapply(adj.list, conversion), function(x){
    if(nrow(x) > 2 && sum(x) > 2){rnetcarto::netcarto(x)}else{list(data.frame(module = c(0,0)
),0)}
  })
  M <- sapply(mod, "[[", 2)
  nMod <- sapply(lapply(mod, "[[", 1), function(x) max(x$module) + 1)

  indices <- matrix(c(N, Ltot, LD, C, D, APL, CC, M, nMod), nrow = nrow(dyn))
  colnames(indices) <- c("N", "Ltot", "LD", "C", "D", "APL", "CC", "M", "nMod")
  return(indices)
}

motifCounter3 <- function(dyn, web){
  adj.list <- lapply(1:nrow(dyn), function(x){web[dyn[x, -1] > 0, dyn[x, -1] > 0]})
  g.list <- lapply(adj.list, graph.adjacency)

  triad.count <- lapply(g.list, triad.census)
  triad.matrix <- matrix(unlist(triad.count), nrow = length(g.list), ncol = 16, byrow = T)
  colnames(triad.matrix) <- c("empty", "single", "mutual", "s5", "s4", "s1", "d4",
                    "d3", "s2", "s3","d8", "d2", "d1", "d5", "d7", "d6")

  triad.df <- as.data.frame(triad.matrix)
  motif.data.frame <- data.frame(s1 = triad.df$s1, s2 = triad.df$s2, s3 = triad.df$s3, s4 = triad.df
$s4,
                                 s5 = triad.df$s5, d1 = triad.df$d1, d2 = triad.df$d2, d3 = triad.df$d3, d4 = tri
ad.df$d4,
                                 d5 = triad.df$d5, d6 = triad.df$d6, d7 = triad.df$d7, d8 = triad.df$d8)

  return(motif.data.frame)
}

trophicChange <- function(dyn, web){
  if(!requireNamespace("NetIndices", quietly = TRUE)){
    stop("This function requires the 'NetIndices' package to be installed and loaded", call. = FALS
E)
    }

  adj.list <- lapply(1:nrow(dyn), function(x){web[dyn[x, -1] > 0, dyn[x, -1] > 0]})
  til <- lapply(adj.list, NetIndices::TrophInd)

  m <- matrix(0, ncol = (ncol(dyn) - 1), nrow = nrow(dyn))
  for(x in 1:nrow(dyn)){
    m[x, dyn[x, -1] > 0] <- til[[x]]$TL
  }
  return(m)
}
```

# Appendix 4: Annotated code for Chapter 5 simulations

The code for all simulations can be found at https://github.com/jjborrelli/fw_dyn. The simulations are defined by four components (four R scripts); the base functions used, the higher level wrapper functions, the simulations themselves, and the analysis.

The required libraries for the following code:

```
library(igraph)
library(NetIndices)
library(rend)
library(data.table)
library(ggplot2)
library(parallel)
library(doSNOW)
library(dplyr)
library(lme4)
library(reshape2)
```

## Base Functions

The first function is the niche model for food webs, with an additional condition that requires the model web to be connected (i.e., no species that neither consume nor is consumed). Species (the number of which is defined by S) are arranged along a single dimensional axis. Each species has a feeding center and range, and consumes all species within its range. The feeding center is randomly sampled to be less than the value the species holds on the niche axis, and the range is sampled according to a beta distribution.

```
niche.model<-function(S,C){
  connected = FALSE
  while(!connected){
    new.mat<-matrix(0,nrow=S,ncol=S)
    ci<-vector()
    niche<-runif(S,0,1)
    r<-rbeta(S,1,((1/(2*C))-1))*niche

    for(i in 1:S){
      ci[i]<-runif(1,r[i]/2,niche[i])
    }

    r[which(niche==min(niche))]<-.00000001

    for(i in 1:S){

      for(j in 1:S){
        if(niche[j]>(ci[i]-(.5*r[i])) && niche[j]<(ci[i]+.5*r[i])){
          new.mat[j,i]<-1
        }
      }
    }

    new.mat<-new.mat[order(apply(new.mat,2,sum)), order(apply(new.mat,2,sum))]

    connected <- is.connected(graph.adjacency(new.mat))
  }
  return(new.mat)
}
```

The `initialNiche` function is designed to sample local food webs of size `S.local` from a pre-defined regional web (`niche`). An additional constraint is the number of basal species in the local web (`nbasal`).

```
initialNiche <- function(niche, S.local, nbasal){
  sppNAMES <- as.character(1:nrow(niche))
  colnames(niche) <- sppNAMES
  rownames(niche) <- sppNAMES
  cond <- FALSE
  #get a connected initial network
  while(!cond){
    #initial basal spp
    inBAS <- sample(which(colSums(niche) == 0), nbasal)
    #initial consumer spp
    inCON <- sample((1:1000)[-which(colSums(niche) == 0)], S.local)
    #all spp
    inSPP <- c(inBAS, inCON)
    #intial matrix
    inN <- niche[inSPP, inSPP]
    cond <- sum(colSums(inN)[names(colSums(inN)) %in% as.character(inCON)] == 0) == 0
  }

  return(inN)
}
```

The `initialize` function takes as arguments the desired number of species and connectance (`Sregion`, `Cregion`) for the regional food web, the number of local food webs to sample (`n.initial`), the number of non-basal species to have in the local web (`Slocal`), the number of basal species in the local webs (`Sbasal`), the number of time steps to simulate the dynamics of the local webs to get equilibrium webs (`times`), and the functional response and tuning parameter for the dynamics (`fr` and `frtune`). This function generates a regional food web and then samples `n.initial` local webs. The dynamics of each local web are simulated (using `rend::CRsimulator`) for `times` time steps with the desired combination of functional response type and tuning parameter. The function outputs a list that includes the regional web adjacency matrix, a list of the local web adjacency matrices, and the matrices of biomass dynamics for each local web.

```
initialize <- function(Sregion, Cregion, n.initial, Slocal = 45, Sbasal = 5, times = 1000, fr = Fij, fr
tune = 0.2){
  n.regional <- niche.model(Sregion, Cregion)

  in.n <- lapply(1:n.initial, function(x) initialNiche(n.regional, S.local = Slocal, nbasal = Sbasal)
)

  cl <- makeCluster(detectCores() - 1)
  registerDoSNOW(cl)
  in.dyn <- parLapply(cl, in.n, CRsimulator, t = 1:times, FuncRes = fr, xpar = frtune)
  stopCluster(cl)

  return(list(REGIONAL = n.regional, INITIAL = in.n, DYNAMICS = in.dyn))
}
```

The motif_counter function takes a list of graphs and returns the frequency of three-species modules.

```
motif_counter <- function(graph.lists){
  require(igraph)

  if(!is.list(graph.lists)){
    stop("The input should be a list of graph objects")
  }

  triad.count <- lapply(graph.lists, triad.census)
  triad.matrix <- matrix(unlist(triad.count), nrow = length(graph.lists), ncol = 16, byrow = T)
  colnames(triad.matrix) <- c("empty", "single", "mutual", "s5", "s4", "s1", "d4",
                             "d3", "s2", "s3","d8", "d2", "d1", "d5", "d7", "d6")

  triad.df <- as.data.frame(triad.matrix)

  motif.data.frame <- data.frame(s1 = triad.df$s1, s2 = triad.df$s2, s3 = triad.df$s3, s4 = triad.df
$s4,
                                s5 = triad.df$s5, d1 = triad.df$d1, d2 = triad.df$d2, d3 = triad.df$d3, d4 = tri
ad.df$d4,
                                d5 = triad.df$d5, d6 = triad.df$d6, d7 = triad.df$d7, d8 = triad.df$d8)

  return(motif.data.frame)
}
```

The wprop function takes in an adjacency matrix and computes a number of food web properties: number of species, number of links, connectance, mean generality, mean vulnerability, standard deviation of generality and vulnerability, mean and standard deviation of trophic position, mean and standard deviation of omnivory index, average path length, diameter, and the frequencies of the thirteen three-species modules.

```
wprop <- function(m){

  N <- nrow(m)
  L <- sum(m)
  C <- L/(N*(N-1))

  meanGen <- mean(colSums(m))
  meanVul <- mean(rowSums(m))
  sdGen <- sd(colSums(m))
  sdVul <- sd(rowSums(m))
  ti <- TrophInd(m)
  meanTP <- mean(ti$TL)
  sdTP <- sd(ti$TL)
  meanOI <- mean(ti$OI)
  sdOI <- sd(ti$OI)

  g <- graph.adjacency(m)
  APL <- average.path.length(g)
  D <- diameter(g)

  mc <- motif_counter(list(g))

  return(data.frame(N, L, C, meanGen, meanVul, sdGen, sdVul, meanTP, sdTP, meanOI, sdOI, APL, D, mc))
}
```

The netdiff function compares two adjacency matrices and computes the difference in their web properties as defined by wprop.

```
netdiff <- function(mat1, mat2){
  w.in <- wprop(mat1)
  w.fi <- wprop(mat2)

  return(w.fi - w.in)
}
```

Two functions are defined to compute the properties of individual species. One, iprop is defined for species newly introduced to a food web, and delpropis defined for species removed from a web. Both take in the adjacency matrix and the identity of the introduced or deleted species. Both return the same properties of the focal species.

```
iprop <- function(m, invader){
  invGen <- sum(m[,ncol(m)])
  invVul <- sum(m[nrow(m),])
  ti <- TrophInd(m)
  invTP <- tail(ti$TL, 1)
  invOI <- tail(ti$OI, 1)

  return(data.frame(invID = invader, invGen, invVul, invTP, invOI))
}

delprop <- function(m, d){
  delGen <- sum(m[,d])
  delVul <- sum(m[d,])
  ti <- TrophInd(m)
  delTP <- ti$TL[d]
  delOI <- ti$OI[d]

  return(data.frame(delID = d, delGen, delVul, delTP, delOI))
}
```

The invading2 function describes the introduction of a single species. It takes as arguments the biomass dynamics of the local food web and its adjacency matrix, the regional web adjacency matrix, and the function response with its tuning parameter. The function defines the equilibrium local web, and then samples a new species from the regional web. The new species is constrained to be either basal, or have at least one resource in the extant community. Following the introduction of the new species the change in web properties is calculated and biomass dynamics are simulated for 500 time steps. The web at 500 time steps is then used to determine whether the new species has established or not, and if not how long before it went extinct. The number of secondary extinctions is also computed.

```
invading2 <- function(dyn1, initial, regional, FR = Fij, frtune = 0.2){
  extant <- which(tail(dyn1, 1)[,-1] > 0)
  extantSPP <- as.numeric(colnames(initial)[extant])
  eqABUND <- tail(dyn1, 1)[,-1][extant]

  eq.n <- initial[extant, extant]
  intro <- sample((1:nrow(regional))[-extantSPP], 1)
  invaded <- c(extantSPP, intro)

  newN <- regional[invaded, invaded]


  if(sum(regional[,intro]) != 0){
    while(sum(newN[,nrow(newN)]) == 0){
      intro <- sample((1:nrow(regional))[-extantSPP], 1)
      invaded <- c(extantSPP, intro)
      newN <- regional[invaded, invaded]
    }
  }
  colnames(newN) <- as.character(invaded)
  rownames(newN) <- as.character(invaded)

  nd1 <- netdiff(eq.n, newN)

  r.in <- as.numeric(colnames(newN) %in% which(colSums(regional) == 0))
  inABUND <- c(as.vector(eqABUND), 0.5)

  dyn <- CRsimulator(newN, states = inABUND, r = r.in, t = 1:500, FuncRes = FR, xpar = frtun
e)

  ext1 <- which(tail(dyn, 1)[,-1] > 0)
  eq.n2 <- newN[ext1, ext1]

  in.success <- intro %in% colnames(eq.n2)
  sp.change <- nrow(eq.n2) - nrow(newN)

  invdyn <- dyn[,(nrow(newN) + 1)]
  if(invdyn[500] != 0){inv.tte <- 500}else{inv.tte <- min(which(invdyn == 0))}

  inv.prop <- iprop(newN, intro)
  inv.prop$dN <- sp.change
  inv.prop$tte <- inv.tte
  inv.prop$I <- in.success

  return(list(nd1, inv.prop))
}
```

The deleting2 function describes the removal of a single species. It takes as arguments the biomass dynamics of the local food web and its adjacency matrix, the regional web adjacency matrix, and the function response with its tuning parameter. The function defines the equilibrium local web, and then removes each species one at a time. Following removal, the change in web properties is computed and the dynamics of the resulting web are simulated for 500 time steps. The simulated dynamics are then examined to determine how many species have gone extinct.

```
deleting2 <- function(dyn1, initial, regional, FR = Fij, frtune = 0.2){
  extant <- which(tail(dyn1, 1)[,-1] > 0)
  extantSPP <- as.numeric(colnames(initial)[extant])
  eqABUND <- tail(dyn1, 1)[,-1][extant]
  eq.n <- initial[extant, extant]

  d.ch <- list()
  nd1 <- list()
  persist <- c()
  s.initial <- c()
  s.final <- c()
  for(i in 1:nrow(eq.n)){
    d.ch[[i]] <- delprop(eq.n, i)
    newN <- eq.n[-i,-i]
    nd1[[i]] <- netdiff(eq.n, newN)
    r.in <- as.numeric(colnames(newN) %in% which(colSums(regional) == 0))
    inABUND <- eqABUND[-i]

    dyn <- CRsimulator(newN, states = inABUND, r = r.in, t = 1:500, FuncRes = FR, xpar = frtu
ne)

    ext1 <- which(tail(dyn, 1)[,-1] > 0)
    persist[i] <- length(ext1)/nrow(newN)
    s.initial[i] <- nrow(newN)
    s.final[i] <- length(ext1)
  }

  nd2 <- data.frame(persist = persist, do.call(rbind, nd1), do.call(rbind, d.ch))
  nd2$s.in <- s.initial
  nd2$s.fi <- s.final
  return(nd2)
}
```

## Wrapper functions

The two wrapper functions apply the `invading2` and `deleting2` in parallel. For each initial network that comes from a regional network `sp.addition` applies the `invading2` function 300 times, and `sp.deletion` applies the `deleting2` function.

```
sp.addition <- function(in.dyn, in.n, n.regional, funcres = Fij, x = 0.2, init){
  cl <- makeCluster(detectCores() - 1)
  registerDoSNOW(cl)

  INVASION <- foreach(i = 1:length(in.n)) %dopar% {
    source("../../../Dropbox/dis-assem.R")
    sink(file = paste0(filepath.sink, "invasion/", init, "/", "invading-web", i,".txt", collapse = ""))

    # SIMULATION
    numinvs = 300
    invaded <- lapply(1:numinvs, function(x) invading2(in.dyn[[i]], in.n[[i]], n.regional, FR = fu
ncres, frtune = x))

    webi.props <- do.call(rbind, lapply(invaded, "[[", 1))
    inv.props <- do.call(rbind, lapply(invaded, "[[", 2))

    webi.props <- data.frame(webi.props, inum = 1:numinvs, locweb = i)
    inv.props <- data.frame(inv.props, inum = 1:numinvs, locweb = i)


    print(list(webi.props, inv.props))

    sink()

    return(list(webi.props, inv.props))
  }

  stopCluster(cl)

  wpdf <- rbindlist(lapply(INVASION, "[[", 1))
  ipdf <- rbindlist(lapply(INVASION, "[[", 2))

  return(list(wpdf = wpdf, ipdf = ipdf))
}


sp.deletion <- function(in.dyn, in.n, n.regional, funcres = Fij, x = 0.2, init){
  cl <- makeCluster(detectCores() - 1)
  registerDoSNOW(cl)

  DELETION <- foreach(i = 1:length(in.n)) %dopar% {
    source("../../../Dropbox/dis-assem.R")
    sink(file = paste0(filepath.sink, "deletion/", init, "/", "deleting-web", i,".txt", collapse = ""))

    # SIMULATION

    deleted <- deleting2(in.dyn[[i]], in.n[[i]], n.regional, FR = funcres, frtune = x)
    del.props <- data.frame(deleted, locweb = i)

    print(del.props)

    sink()

    return(del.props)
```

```
 }

 stopCluster(cl)


 dels <- rbindlist(DELETION)

 return(dels)
}
```

Simulations

Six scenarios were simulated, defined by the functional response type and the tuning parameter. For each scenario there was a regional web from which 200 initial webs were sampled.

```
#########################
## Get starter webs

init1 <- initialize(Sregion = 1000, Cregion = 0.15, n.initial = 200, Slocal = 45, Sbasal = 5, times
= 1000, fr = Fij, frtune = 0)
in1.eqS <- sapply(init1[["DYNAMICS"]], function(x){sum(x[1000, -1] > 0)})
init2 <- initialize(Sregion = 1000, Cregion = 0.15, n.initial = 200, Slocal = 45, Sbasal = 5, times
= 1000, fr = Fij, frtune = 0.2)
in2.eqS <- sapply(init2[["DYNAMICS"]], function(x){sum(x[1000, -1] > 0)})
init3 <- initialize(Sregion = 1000, Cregion = 0.15, n.initial = 200, Slocal = 45, Sbasal = 5, times
= 1000, fr = Fij, frtune = 1)
in3.eqS <- sapply(init3[["DYNAMICS"]], function(x){sum(x[1000, -1] > 0)})

init4 <- initialize(Sregion = 1000, Cregion = 0.15, n.initial = 200, Slocal = 45, Sbasal = 5, times
= 1000, fr = Fbd, frtune = 0)
in4.eqS <- sapply(init4[["DYNAMICS"]], function(x){sum(x[1000, -1] > 0)})
init5 <- initialize(Sregion = 1000, Cregion = 0.15, n.initial = 200, Slocal = 45, Sbasal = 5, times
= 1000, fr = Fbd, frtune = 0.2)
in5.eqS <- sapply(init5[["DYNAMICS"]], function(x){sum(x[1000, -1] > 0)})
init6 <- initialize(Sregion = 1000, Cregion = 0.15, n.initial = 200, Slocal = 45, Sbasal = 5, times
= 1000, fr = Fbd, frtune = 1)
in6.eqS <- sapply(init6[["DYNAMICS"]], function(x){sum(x[1000, -1] > 0)})

#########################
## Deletion simulations
allstrt <- Sys.time()

spdel1 <- sp.deletion(in.dyn = init1[["DYNAMICS"]][in1.eqS> 5], in.n = init1[["INITIAL"]][in
1.eqS > 5], n.regional = init1[["REGIONAL"]], funcres = Fij, x = 0, init = "ini1")
spdel2 <- sp.deletion(in.dyn = init2[["DYNAMICS"]][in2.eqS> 5], in.n = init2[["INITIAL"]][in
2.eqS> 5], n.regional = init2[["REGIONAL"]], funcres = Fij, x = 0.2, init = "ini2")
```

```
spdel3 <- sp.deletion(in.dyn = init3[["DYNAMICS"]][in3.eqS> 5], in.n = init3[["INITIAL"]][in
3.eqS> 5], n.regional = init3[["REGIONAL"]], funcres = Fij, x = 1, init = "ini3")
spdel4 <- sp.deletion(in.dyn = init4[["DYNAMICS"]][in4.eqS> 5], in.n = init4[["INITIAL"]][in
4.eqS> 5], n.regional = init4[["REGIONAL"]], funcres = Fbd, x = 0, init = "ini4")
spdel5 <- sp.deletion(in.dyn = init5[["DYNAMICS"]][in5.eqS> 5], in.n = init5[["INITIAL"]][in
5.eqS> 5], n.regional = init5[["REGIONAL"]], funcres = Fbd, x = 0.2, init = "ini5")
spdel6 <- sp.deletion(in.dyn = init6[["DYNAMICS"]][in6.eqS> 5], in.n = init6[["INITIAL"]][in
6.eqS> 5], n.regional = init6[["REGIONAL"]], funcres = Fbd, x = 1, init = "ini6")

allend <- Sys.time()

allend - allstrt

########################
## Invasion simulations
allstrt1 <- Sys.time()

spadd1 <- sp.addition(in.dyn = init1[["DYNAMICS"]][in1.eqS> 5], in.n = init1[["INITIAL"]][i
n1.eqS> 5], n.regional = init1[["REGIONAL"]], funcres = Fij, x = 0, init = "ini1")
end1 <- Sys.time()
end1 - allstrt1

spadd2 <- sp.addition(in.dyn = init2[["DYNAMICS"]][in2.eqS> 5], in.n = init2[["INITIAL"]][i
n2.eqS> 5], n.regional = init2[["REGIONAL"]], funcres = Fij, x = 0.2, init = "ini2")
end2 <- Sys.time()
end2 - end1
spadd3 <- sp.addition(in.dyn = init3[["DYNAMICS"]][in3.eqS> 5], in.n = init3[["INITIAL"]][i
n3.eqS > 5], n.regional = init3[["REGIONAL"]], funcres = Fij, x = 1, init = "ini3")
end3 <- Sys.time()
end3 - end2
spadd4 <- sp.addition(in.dyn = init4[["DYNAMICS"]][in4.eqS> 5], in.n = init4[["INITIAL"]][i
n4.eqS> 5], n.regional = init4[["REGIONAL"]], funcres = Fbd, x = 0, init = "ini4")
end4 <- Sys.time()
end4 - end3
spadd5 <- sp.addition(in.dyn = init5[["DYNAMICS"]][in5.eqS> 5], in.n = init5[["INITIAL"]][i
n5.eqS> 5], n.regional = init5[["REGIONAL"]], funcres = Fbd, x = 0.2, init = "ini5")
end5 <- Sys.time()
end5 - end4
spadd6 <- sp.addition(in.dyn = init6[["DYNAMICS"]][in6.eqS> 5], in.n = init6[["INITIAL"]][i
n6.eqS> 5], n.regional = init6[["REGIONAL"]], funcres = Fbd, x = 1, init = "ini6")

allend1 <- Sys.time()

allend1 - allstrt1
```

Analysis

```
# How many webs in each

nweb1 <- sum(in1.eqS > 5)
nweb2 <- sum(in2.eqS > 5)
nweb3 <- sum(in3.eqS > 5)
nweb4 <- sum(in4.eqS > 5)
nweb5 <- sum(in5.eqS > 5)
nweb6 <- sum(in6.eqS > 5)

n.eqwebs <- c(nweb1, nweb2, nweb3, nweb4, nweb5, nweb6)

# How many species were deleted from each web

ndel1 <- sum(in1.eqS[in1.eqS > 5])
ndel2 <- sum(in2.eqS[in2.eqS > 5])
ndel3 <- sum(in3.eqS[in3.eqS > 5])
ndel4 <- sum(in4.eqS[in4.eqS > 5])
ndel5 <- sum(in5.eqS[in5.eqS > 5])
ndel6 <- sum(in6.eqS[in6.eqS > 5])

n.deletions <- c(ndel1, ndel2, ndel3, ndel4, ndel5, ndel6)

# How many invasions events for each set

n.invasions <- c(nweb1, nweb2, nweb3, nweb4, nweb5, nweb6)*300
n.invasions


# Is the web connected

iscon1 <- sapply(1:200, function(x){is.connected(graph.adjacency(init1[["INITIAL"]][[x]][init1[["DYNAMICS"]][[x]][1000,-1] > 0,init1[["DYNAMICS"]][[x]][1000,-1] > 0]))})
iscon2 <- sapply(1:200, function(x){is.connected(graph.adjacency(init2[["INITIAL"]][[x]][init2[["DYNAMICS"]][[x]][1000,-1] > 0,init2[["DYNAMICS"]][[x]][1000,-1] > 0]))})
iscon3 <- sapply(1:200, function(x){is.connected(graph.adjacency(init3[["INITIAL"]][[x]][init3[["DYNAMICS"]][[x]][1000,-1] > 0,init3[["DYNAMICS"]][[x]][1000,-1] > 0]))})
iscon4 <- sapply(1:200, function(x){is.connected(graph.adjacency(init4[["INITIAL"]][[x]][init4[["DYNAMICS"]][[x]][1000,-1] > 0,init4[["DYNAMICS"]][[x]][1000,-1] > 0]))})
iscon5 <- sapply(1:200, function(x){is.connected(graph.adjacency(init5[["INITIAL"]][[x]][init5[["DYNAMICS"]][[x]][1000,-1] > 0,init5[["DYNAMICS"]][[x]][1000,-1] > 0]))})
iscon6 <- sapply(1:200, function(x){is.connected(graph.adjacency(init6[["INITIAL"]][[x]][init6[["DYNAMICS"]][[x]][1000,-1] > 0,init6[["DYNAMICS"]][[x]][1000,-1] > 0]))})
```

```
# How many species in each web set

l1 <- list(in1.eqS[in1.eqS > 5 & iscon1], in2.eqS[in2.eqS > 5 & iscon2], in3.eqS[in3.eqS > 5 & i
scon3], in4.eqS[in4.eqS > 5 & iscon4], in5.eqS[in5.eqS > 5 & iscon5], in6.eqS[in6.eqS > 5 & isc
on6])
ggplot(melt(l1), aes(x = factor(L1), y = value)) + geom_boxplot()


#########################################################
#### DATA SELECTION

# get deletion data from each connected web

spd1 <- spdel1[spdel1$locweb %in% which(iscon1[in1.eqS > 5]),]
spd2 <- spdel2[spdel2$locweb %in% which(iscon2[in2.eqS > 5]),]
spd3 <- spdel3[spdel3$locweb %in% which(iscon3[in3.eqS > 5]),]
spd4 <- spdel4[spdel4$locweb %in% which(iscon4[in4.eqS > 5]),]
spd5 <- spdel5[spdel5$locweb %in% which(iscon5[in5.eqS > 5]),]
spd6 <- spdel6[spdel6$locweb %in% which(iscon6[in6.eqS > 5]),]

# get invasion data from each connectd web

spa1 <- lapply(spadd1, function(x) x[x$locweb %in% which(iscon1[in1.eqS > 5]),])
spa2 <- lapply(spadd2, function(x) x[x$locweb %in% which(iscon2[in1.eqS > 5]),])
spa3 <- lapply(spadd3, function(x) x[x$locweb %in% which(iscon3[in1.eqS > 5]),])
spa4 <- lapply(spadd4, function(x) x[x$locweb %in% which(iscon4[in1.eqS > 5]),])
spa5 <- lapply(spadd5, function(x) x[x$locweb %in% which(iscon5[in1.eqS > 5]),])
spa6 <- lapply(spadd6, function(x) x[x$locweb %in% which(iscon6[in1.eqS > 5]),])
```

Plotting functions for the subsets of the data.

```
wplot.del <- function(spdlist){
  dyn <- expand.grid(c("Fij", "Fbd"), c("0", "0.2", "1"))
  df <- list()
  for(i in 1:length(spdlist)){
    spd <- spdlist[[i]]
    spdP <- spd$persist == 1
    spdNP <- spd$persist != 1
    wp.spd1 <- dplyr::select(spd, L:D, -C)
    t1 <- melt(wp.spd1[spdNP,], measure.vars = names(wp.spd1[spdNP,]))
    df1 <- data.frame(values = t1$value, metric = factor(t1$variable, levels = names(wp.spd1)),
                       L1 = i, FR = dyn[i,1], par = dyn[i,2], stability = "not persistent")
    t2 <- melt(wp.spd1[spdP,], measure.vars = names(wp.spd1[spdP,]))
    df2 <- data.frame(values = t2$value, metric = factor(t2$variable, levels = names(wp.spd1)),
                  L1 = i, FR = dyn[i,1], par = dyn[i,2], stability = "persistent")
    df[[i]] <- rbind(df1, df2)
  }

  return(rbindlist(df))
}

mplot.del <- function(spdlist){
  dyn <- expand.grid(c("Fij", "Fbd"), c("0", "0.2", "1"))
  df <- list()
  for(i in 1:length(spdlist)){
    spd <- spdlist[[i]]
    spdP <- spd$persist == 1
    spdNP <- spd$persist != 1
    mo.spd1 <- dplyr::select(spd, s1:d8)
    t1 <- melt(mo.spd1[spdNP,], measure.vars = names(mo.spd1[spdNP,]))
    df1 <- data.frame(values = t1$value, metric = factor(t1$variable, levels = names(mo.spd1)),
                      L1 = i, FR = dyn[i,1], par = dyn[i,2], stability = "not persistent")
    t2 <- melt(mo.spd1[spdP,], measure.vars = names(mo.spd1[spdP,]))
    df2 <- data.frame(values = t2$value, metric = factor(t2$variable, levels = names(mo.spd1)),
                      L1 = i, FR = dyn[i,1], par = dyn[i,2], stability = "persistent")
    df[[i]] <- rbind(df1, df2)
  }

  return(rbindlist(df))
}

dplot.del <- function(spdlist){
  dyn <- expand.grid(c("Fij", "Fbd"), c("0", "0.2", "1"))
  df <- list()
  for(i in 1:length(spdlist)){
    spd <- spdlist[[i]]
    spdP <- spd$persist == 1
    spdNP <- spd$persist != 1
    dp.spd1 <- dplyr::select(spd, delGen:delOI)
    t1 <- melt(dp.spd1[spdNP,], measure.vars = names(dp.spd1[spdNP,]))
    df1 <- data.frame(values = t1$value, metric = factor(t1$variable, levels = names(dp.spd1)),
                      L1 = i, FR = dyn[i,1], par = dyn[i,2], stability = "not persistent")
    t2 <- melt(dp.spd1[spdP,], measure.vars = names(dp.spd1[spdP,]))
    df2 <- data.frame(values = t2$value, metric = factor(t2$variable, levels = names(dp.spd1)),
                      L1 = i, FR = dyn[i,1], par = dyn[i,2], stability = "persistent")
    df[[i]] <- rbind(df1, df2)
  }

  return(rbindlist(df))
}
```

```
wplot.inv <- function(spalist){
  dyn <- expand.grid(c("Fij", "Fbd"), c("0", "0.2", "1"))
  df <- list()
  for(i in 1:length(spalist)){
    spd <- spalist[[i]]
    spdP <- spd[[2]]$I == TRUE
    spdNP <- spd[[2]]$I == FALSE
    wp.spd1 <- dplyr::select(spd[[1]], L:D, -C)
    t1 <- melt(wp.spd1[spdNP,], measure.vars = names(wp.spd1[spdNP,]))
    df1 <- data.frame(values = t1$value, metric = factor(t1$variable, levels = names(wp.spd1)),
                      L1 = i, FR = dyn[i,1], par = dyn[i,2], stability = "not invaded")
    t2 <- melt(wp.spd1[spdP,], measure.vars = names(wp.spd1[spdP,]))
    df2 <- data.frame(values = t2$value, metric = factor(t2$variable, levels = names(wp.spd1)),
                      L1 = i, FR = dyn[i,1], par = dyn[i,2], stability = "invaded")
    df[[i]] <- rbind(df1, df2)
  }

  return(rbindlist(df))
}

mplot.inv <- function(spalist){
  dyn <- expand.grid(c("Fij", "Fbd"), c("0", "0.2", "1"))
  df <- list()
  for(i in 1:length(spalist)){
    spd <- spalist[[i]]
    spdP <- spd[[2]]$I == TRUE
    spdNP <- spd[[2]]$I == FALSE
    mo.spd1 <- dplyr::select(spd[[1]], s1:d8)
    t1 <- melt(mo.spd1[spdNP,], measure.vars = names(mo.spd1[spdNP,]))
    df1 <- data.frame(values = t1$value, metric = factor(t1$variable, levels = names(mo.spd1)),
                      L1 = i, FR = dyn[i,1], par = dyn[i,2], stability = "not invaded")
    t2 <- melt(mo.spd1[spdP,], measure.vars = names(mo.spd1[spdP,]))
    df2 <- data.frame(values = t2$value, metric = factor(t2$variable, levels = names(mo.spd1)),
                      L1 = i, FR = dyn[i,1], par = dyn[i,2], stability = "invaded")
    df[[i]] <- rbind(df1, df2)
  }

  return(rbindlist(df))
}

dplot.inv <- function(spdlist){
  dyn <- expand.grid(c("Fij", "Fbd"), c("0", "0.2", "1"))
  df <- list()
  for(i in 1:length(spdlist)){
    spd <- spdlist[[i]]
    spdP <- spd[[2]]$I == TRUE
    spdNP <- spd[[2]]$I == FALSE
    dp.spd1 <- dplyr::select(spd[[2]], invGen:dN)
    dp.spd1$dN <- dp.spd1$dN * -1
    t1 <- melt(dp.spd1[spdNP,], measure.vars = names(dp.spd1[spdNP,]))
    df1 <- data.frame(values = t1$value, metric = factor(t1$variable, levels = names(dp.spd1)),
                      L1 = i, FR = dyn[i,1], par = dyn[i,2], stability = "not invaded")
    t2 <- melt(dp.spd1[spdP,], measure.vars = names(dp.spd1[spdP,]))
    df2 <- data.frame(values = t2$value, metric = factor(t2$variable, levels = names(dp.spd1)),
                      L1 = i, FR = dyn[i,1], par = dyn[i,2], stability = "invaded")
    df[[i]] <- rbind(df1, df2)
  }

  return(rbindlist(df))
}

max.se <- function(x){mean(x) + 1.96*(sd(x)/sqrt(length(x)))}
min.se <- function(x){mean(x) - 1.96*(sd(x)/sqrt(length(x)))}
```

Plotting the figures.

```
spdl <- list(spd1, spd2, spd3, spd4, spd5, spd6)

ggplot(wplot.del(spdl), aes(x = par, y = values, fill = stability)) +
  geom_bar(stat = "summary", position = "dodge") +
  stat_summary(fun.y = "mean", fun.ymin = min.se, fun.ymax = max.se, geom = "errorbar", pos
ition = "dodge") +
  facet_grid(metric~FR, scales = "free_y") +
  theme_bw() + #theme(axis.text.x = element_text(angle = -90, hjust = 0, vjust = 0)) +
  xlab("Model Parameter") + ylab("Change in Value Following Species Removal")
#ggsave("./Figs/wplotDEL.png")

ggplot(mplot.del(spdl), aes(x = par, y = values, fill = stability))  +
  geom_bar(stat = "summary", position = "dodge") +
  stat_summary(fun.y = "mean", fun.ymin = min.se, fun.ymax = max.se, geom = "errorbar", pos
ition = "dodge") +
  facet_grid(metric~FR, scales = "free_y") +
  theme_bw() +
  xlab("Model Parameter") + ylab("Change in Frequency Following Species Removal")
#ggsave("./Figs/mplotDEL.png")

ggplot(dplot.del(spdl), aes(x = par, y = values, fill = stability))  +
  geom_bar(stat = "summary", position = "dodge") +
  stat_summary(fun.y = "mean", fun.ymin = min.se, fun.ymax = max.se, geom = "errorbar", pos
ition = "dodge") +
  facet_grid(metric~FR, scales = "free_y") +
  theme_bw() +
  xlab("Model Parameter") + ylab("Value For Removed Species")
#ggsave("./Figs/dplotDEL.png")


spal <- list(spa1, spa2, spa3, spa4, spa5, spa6)

ggplot(wplot.inv(spal), aes(x = par, y = values, fill = stability)) +
  geom_bar(stat = "summary", position = "dodge") +
  stat_summary(fun.y = "mean", fun.ymin = min.se, fun.ymax = max.se, geom = "errorbar", pos
ition = "dodge") +
  facet_grid(metric~FR, scales = "free_y") +
  theme_bw() + #theme(axis.text.x = element_text(angle = -90, hjust = 0, vjust = 0)) +
  xlab("Model Parameter") + ylab("Change in Value Following Species Introduction")
#ggsave("./Figs/wplotINV.png")

ggplot(mplot.inv(spal), aes(x = par, y = values, fill = stability)) +
  geom_bar(stat = "summary", position = "dodge") +
  stat_summary(fun.y = "mean", fun.ymin = min.se, fun.ymax = max.se, geom = "errorbar", pos
ition = "dodge") +
  facet_grid(metric~FR, scales = "free_y") +
  theme_bw() +
  xlab("Model Parameter") + ylab("Change in Frequency Following Species Introduction")
#ggsave("./Figs/mplotINV.png")

ggplot(dplot.inv(spal), aes(x = par, y = values, fill = stability)) +
  geom_bar(stat = "summary", position = "dodge") +
  stat_summary(fun.y = "mean", fun.ymin = min.se, fun.ymax = max.se, geom = "errorbar", pos
ition = "dodge") +
  facet_grid(metric~FR, scales = "free_y") +
  theme_bw() +
  xlab("Model Parameter") + ylab("Value for Introduced Species")
#ggsave("./Figs/dplotINV.png")
```

Principal components regression with linear mixed effects models for species deletions

```
get_pc <- function(spdeldata){
  subg1 <- select(spdeldata, s1:d8)
  webp1 <- select(spdeldata, L:D, -C)
  webp1$APL[is.na(webp1$APL)] <- 0
  delp1 <- select(spdeldata, delGen:delOI)

  pcsub <- princomp(subg1)
  pcweb <- princomp(webp1)
  pcdel <- princomp(delp1)

  return(list(pcsub, pcweb, pcdel))
}


get_df_glms <- function(spdeldata, pcomplist){
  pcsub <- pcomplist[[1]]
  pcweb <- pcomplist[[2]]
  pcdel <- pcomplist[[3]]

  persisted <- select(spdeldata, persist)
  persisted.bin <- matrix(c(spdeldata$s.fi, (spdeldata$s.in - spdeldata$s.fi)), ncol = 2)


  df1 <- data.frame(pc1 = pcsub$scores[,1], pc2 = pcsub$scores[,2], pc3 = pcsub$scores[,3], dID = spdeldata$delID, lweb = spdeldata$locweb)
  df2 <- data.frame(pc1 = pcweb$scores[,1], pc2 = pcweb$scores[,2], pc3 = pcweb$scores[,3], dID = spdeldata$delID, lweb = spdeldata$locweb)
  df3 <- data.frame(pc1 = pcdel$scores[,1], pc2 = pcdel$scores[,2], pc3 = pcdel$scores[,3], dID = spdeldata$delID, lweb = spdeldata$locweb)

  return(list(df1, df2, df3))
}

SS <- function(x, y){
  ssr <- sum((fitted(x) - y)^2)
  sst <- sum((y- mean(y))^2)
  return(c(r2 = ssr/sst, SSR = ssr))
}


spd1pc <- get_pc(spd1)
spd2pc <- get_pc(spd2)
spd3pc <- get_pc(spd3)
spd4pc <- get_pc(spd4)
spd5pc <- get_pc(spd5)
spd6pc <- get_pc(spd6)

load1sD <- lapply(list(spd1pc, spd2pc, spd3pc, spd4pc, spd5pc, spd6pc), function(q) lapply(q, function(x) loadings(x)[,1]))
subloadingD <- round(sapply(load1sD, "[[",1), 3)
webloadingD <- round(sapply(load1sD, "[[",2), 3)
delloadingD <- round(sapply(load1sD, "[[",3), 3)
#write.csv(subloadingD, "D:/jjborrelli/AssemblyDATA/invdel/subloadD.csv")
```

```
#write.csv(webloadingD, "D:/jjborrelli/AssemblyDATA/invdel/webloadD.csv")
#write.csv(delloadingD, "D:/jjborrelli/AssemblyDATA/invdel/delloadD.csv")


spd1gm <- get_df_glms(spd1, spd1pc)
spd2gm <- get_df_glms(spd2, spd2pc)
spd3gm <- get_df_glms(spd3, spd3pc)
spd4gm <- get_df_glms(spd4, spd4pc)
spd5gm <- get_df_glms(spd5, spd5pc)
spd6gm <- get_df_glms(spd6, spd6pc)

allspD <- list(spd1, spd2, spd3, spd4, spd5, spd6)
allgmdf <- list(spd1gm, spd2gm, spd3gm, spd4gm, spd5gm, spd6gm)

allgmtab <- list()
for(i in 1:6){
  # glm web prop
  persisted.bin <- cbind(allspD[[i]]$s.fi, (allspD[[i]]$s.in - allspD[[i]]$s.fi))
  gm1 <- glmer(persisted.bin ~ pc1 + (1 | lweb), data = allgmdf[[i]][[1]], family = "binomial")
  ss <- getME(gm1,c("theta","fixef"))
  gm1.1 <- update(gm1,start=ss,control=glmerControl(optimizer="bobyqa", optCtrl=list(maxfu
n=2e5)))
  gm.sum <- summary(gm1.1)
  gm.ss <- SS(gm1, allspD[[i]]$persist)
  subg.gm <- c(gm.sum$coefficients[2,], gm.sum$AICtab, gm.ss)

  gm2 <- glmer(persisted.bin ~ pc1 + (1 | lweb), data = allgmdf[[i]][[2]], family = "binomial")
  gm.sum <- summary(gm2)
  gm.ss <- SS(gm2, allspD[[i]]$persist)
  webp.gm <- c(gm.sum$coefficients[2,], gm.sum$AICtab, gm.ss)

  gm3 <- glmer(persisted.bin ~ pc1 + (1 | lweb), data = allgmdf[[i]][[3]], family = "binomial")
  gm.sum <- summary(gm3)
  gm.ss <- SS(gm3, allspD[[i]]$persist)
  delp.gm <- c(gm.sum$coefficients[2,], gm.sum$AICtab, gm.ss)

  allgmtab[[i]] <- rbind(subg.gm, webp.gm, delp.gm)
  print(i)
}

allgmtab
```

Principal components regression with linear mixed effects models for species invasions.

```
get_pc_inv <- function(spinvdata){
  subg1 <- select(spinvdata[[1]], s1:d8)
```

```r
  webp1 <- select(spinvdata[[1]], L:D, -C)
  webp1$APL[is.na(webp1$APL)] <- 0
  invp1 <- select(spinvdata[[2]], invGen:invOI)

  pcsub <- princomp(subg1)
  pcweb <- princomp(webp1)
  pcinv <- princomp(invp1)

  return(list(pcsub, pcweb, pcinv))
}


get_df_glmsI <- function(spdeldata, pcomplist){
  pcsub <- pcomplist[[1]]
  pcweb <- pcomplist[[2]]
  pcdel <- pcomplist[[3]]


  df1 <- data.frame(pc1 = pcsub$scores[,1], pc2 = pcsub$scores[,2], pc3 = pcsub$scores[,3], lwe
b = spdeldata[[1]]$locweb)
  df2 <- data.frame(pc1 = pcweb$scores[,1], pc2 = pcweb$scores[,2], pc3 = pcweb$scores[,3], l
web = spdeldata[[1]]$locweb)
  df3 <- data.frame(pc1 = pcdel$scores[,1], pc2 = pcdel$scores[,2], pc3 = pcdel$scores[,3], lweb
= spdeldata[[1]]$locweb)

  return(list(df1, df2, df3))
}



spa1pc <- get_pc_inv(spa1)
spa2pc <- get_pc_inv(spa2)
spa3pc <- get_pc_inv(spa3)
spa4pc <- get_pc_inv(spa4)
spa5pc <- get_pc_inv(spa5)
spa6pc <- get_pc_inv(spa6)

load1s <- lapply(list(spa1pc, spa2pc, spa3pc, spa4pc, spa5pc, spa6pc), function(q) lapply(q, fun
ction(x) loadings(x)[,1]))
subloadingI <- round(sapply(load1s, "[[",1), 3)
webloadingI <- round(sapply(load1s, "[[",2), 3)
invloadingI <- round(sapply(load1s, "[[",3), 3)
#write.csv(subloadingI, "D:/jjborrelli/AssemblyDATA/invdel/subloadI.csv")
#write.csv(webloadingI, "D:/jjborrelli/AssemblyDATA/invdel/webloadI.csv")
#write.csv(invloadingI, "D:/jjborrelli/AssemblyDATA/invdel/invloadI.csv")
```

```
spa1gm <- get_df_glmsI(spa1, spa1pc)
spa2gm <- get_df_glmsI(spa2, spa2pc)
spa3gm <- get_df_glmsI(spa3, spa3pc)
spa4gm <- get_df_glmsI(spa4, spa4pc)
spa5gm <- get_df_glmsI(spa5, spa5pc)
spa6gm <- get_df_glmsI(spa6, spa6pc)

allspI <- list(spa1[[2]], spa2[[2]], spa3[[2]], spa4[[2]], spa5[[2]], spa6[[2]])
allgmdfI <- list(spa1gm, spa2gm, spa3gm, spa4gm, spa5gm, spa6gm)

allgmtabI <- list()
for(i in 1:6){
  # glm web prop
  invading <- allspI[[i]]$I
  gm1 <- glmer(invading ~ pc1 + (1 | lweb), data = allgmdfI[[i]][[1]], family = "binomial")
  gm1.1 <- update(gm1,start=ss,control=glmerControl(optimizer="bobyqa", optCtrl=list(maxfun=2e5)))
  gm.sum <- summary(gm1.1)
  gm.ss <- SS(gm1, invading)
  subg.gm <- c(gm.sum$coefficients[2,], gm.sum$AICtab, gm.ss)

  gm2 <- glmer(invading ~ pc1 + (1 | lweb), data = allgmdfI[[i]][[2]], family = "binomial")
  gm.sum <- summary(gm2)
  gm.ss <- SS(gm2, invading)
  webp.gm <- c(gm.sum$coefficients[2,], gm.sum$AICtab, gm.ss)

  gm3 <- glmer(invading ~ pc1 + (1 | lweb), data = allgmdfI[[i]][[3]], family = "binomial")
  gm.sum <- summary(gm3)
  gm.ss <- SS(gm3, invading)
  invp.gm <- c(gm.sum$coefficients[2,], gm.sum$AICtab, gm.ss)

  allgmtabI[[i]] <- rbind(subg.gm, webp.gm, invp.gm)
  print(i)
}

allgmtabI
```