

Appendix A

Data

Data Sources

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).

References

1.

Baird, D. and Ulanowicz, R. (1989). The seasonal dynamics of the Chesapeake Bay ecosystem. *Ecol. Monogr.*, 59, 329â364.

2.

Christensen, V. and Pauly, D. (1992). ECOPATH IIâ software for balancing steady-state ecosystem models and calculating network characteristics. *Ecol. Modell.*, 61, 169â185.

3.

Christian, R.R. and Luczkovich, J.J. (1999). Organizing and understanding a winterâs seagrass foodweb network through effective trophic levels. *Ecol. Modell.*, 117, 99â124.

4.

Dunne, J., Williams, R. and Martinez, N. (2004). Network structure and robustness of marine food webs. *Mar. Ecol. Prog. Ser.*, 273, 291â302.

5.

Dunne, J.A., Williams, R.J. and Martinez, N.D. (2002). Network structure and biodiversity loss in food webs: robustness increases with connectance. *Ecol. Lett.*, 5, 558â567.

6.

Goldwasser, L. and Roughgarden, J. (1993). Construction and analysis of a large Caribbean food web. *Ecology*, 74, 1216â1233.

7.

Hall, S. and Raffaelli, D. (1991). Food-web patterns: lessons from a species-rich web. *J. Anim. Ecol.*, 60, 823â841.

8.

Havens, K. (1992). Scale and structure in natural food webs. *Science.*, 257, 1107â1109.

9.

Hechinger, R.F., Lafferty, K.D., McLaughlin, J.P., Fredensborg, B.L., Huspeni, T.C., Lorda, J., et al. (2011). Food webs including parasites, biomass, body sizes, and life stages for three California/Baja California estuaries. *Ecology*, 92, 791.

10.

Jaarsma, N.G., de Boer, S.M., Townsend, C.R., Thompson, R.M. and Edwards, E.D. (1998). Characterising foodâwebs in two New Zealand streams. *New Zeal. J. Mar. Freshw. Res.*, 32, 271â286.

11.

Link, J. (2002). Does food web theory work for marine ecosystems? *Mar. Ecol. Prog. Ser.*, 230, 1â9.

12.

Martinez, N. (1991). Artifacts or Attributes? Effects of Resolution on the Little Rock Lake Food Web. *Ecol. Monogr.*, 61, 367â392.

13.

Martinez, N., Hawkins, B., Dawah, H. and Feifarek, B. (1999). Effects of sampling effort on characterization of food-web structure. *Ecology*, 80, 1044â1055.

14.

Memmott, J., Martinez, N.D. and Cohen, J.E. (2000). Predators, parasitoids and pathogens: species richness, trophic generality and body sizes in a natural food web. *J. Anim. Ecol.*, 69, 1â15.

15.

Mouritsen, K.N., Poulin, R., McLaughlin, J.P. and Thieltges, D.W. (2011). Food web including metazoan parasites for an intertidal ecosystem in New Zealand. *Ecology*, 92, 2006â2006.

16.

Opitz, S. (1996). Trophic interactions in Caribbean coral reefs. ICALRM Tech, Makati City, Philippines.

17.

Polis, G. (1991). Complex trophic interactions in deserts: an empirical critique of food-web theory. *Am. Nat.*, 138, 123â155.

19.

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

20.

Roopnarine, P.D. and Hertog, R. (2012a). Data from: Detailed food web networks of three Greater Antillean coral reef systems: the Cayman Islands, Cuba, and Jamaica. Dryad Digital Repository. doi: 10.5061/dryad.c213h

21.

Roopnarine, P.D. and Hertog, R. (2012b). Detailed food web networks of three Greater Antillean coral reef systems: the Cayman Islands, Cuba, and Jamaica. Dataset Pap. Ecol., 2013, 857470. doi:10.7167/2013/857470

22.

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

23.

Thompson, R. and Edwards, E. (2001). Allocation of effort in stream food-web studies: the best compromise? *Mar. Freshw. Res.*, 52, 339â345.

24.

Thompson, R. and Townsend, C. (1999). The effect of seasonal variation on the community structure and food-web attributes of two streams: implications for food-web science. *Oikos*, 87, 75â88.

25.

Thompson, R. and Townsend, C. (2003). Impacts on stream food webs of native and exotic forest: an intercontinental comparison. *Ecology*, 84, 145â161.

26.

Thompson, R. and Townsend, C. (2005). Energy availability, spatial heterogeneity and ecosystem size predict food-web structure in streams. *Oikos*, 108, 137â148.

27.

Thompson, R.M. and Townsend, C.R. (2000). Is resolution the solution?: the effect of taxonomic resolution on the calculated properties of three stream food webs. *Freshw. Biol.*, 44, 413â422.

28.

Townsend, C.R., Thompson, R.M., McIntosh, A.R., Kilroy, C., Edwards, E. and Scarsbrook, M.R. (1998). Disturbance, resource supply, and food-web architecture in streams. *Ecol. Lett.*, 1, 200â209.

29.

Waide, R. and Reagan, W. (1996). *The food web of a tropical rainforest*. Univ. Chicago. University of Chicago Press, Chicago, Illinois, U.S.A.

30.

Warren, P. (1989). Spatial and temporal variation in the structure of a freshwater food web. *Oikos*, 55, 299â311.

31.

Yodzis, P. (1998). Local trophodynamics and the interaction of marine mammals and fisheries in the Benguela ecosystem. *J. Anim. Ecol.*, 67, 635â658.

32.

Yodzis, P. (2000). Diffuse Effects in Food Webs. *Ecology*, 81, 261â266.

33.

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

Code used in “Why are there so few trophic levels: selection against instability explains the pattern”

Loading required packages

```
require(RCurl)
require(igraph)
require(reshape2)
require(ggplot2)
require(grid)
require(devtools)
```

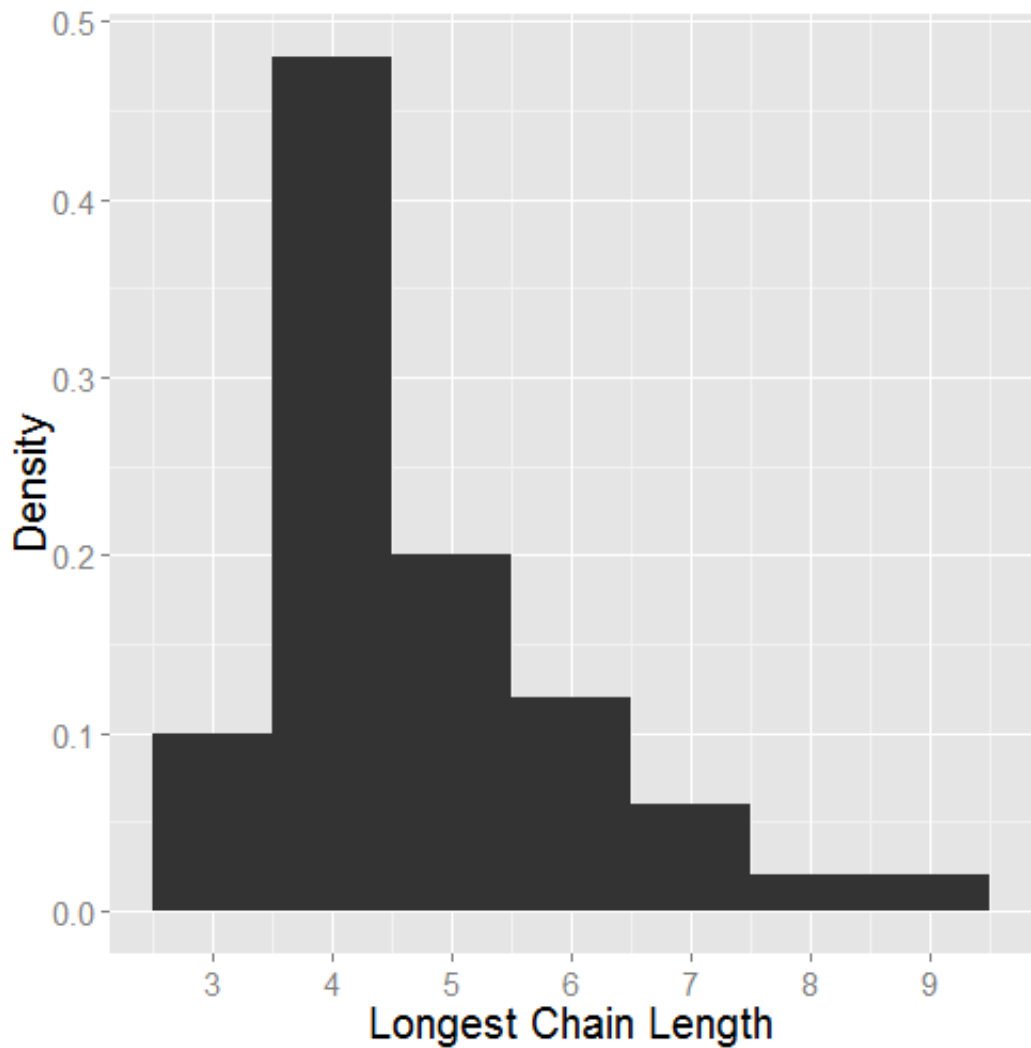
Figure S1

```
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, y
= ..density..))
diam.plot <- diam.plot + geom_histogram(breaks = seq(2.5,
9.5, 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")
diam.plot <- diam.plot + scale_x_continuous(name = "Longest
Chain Length", breaks = 0:9)
```

Figure A1: A histogram of the longest food chain in each of 50 food 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
```

```

$`2` sp`
      [,1] [,2]
[1,]    -1    1
[2,]    -1   -1

$`3` sp`
      [,1] [,2] [,3]
[1,]    -1    1    1
[2,]    -1   -1    1
[3,]    -1   -1   -1

$`4` sp`
      [,1] [,2] [,3] [,4]
[1,]    -1    1    1    1
[2,]    -1   -1    1    1
[3,]    -1   -1   -1    1
[4,]    -1   -1   -1   -1

$`5` sp`
      [,1] [,2] [,3] [,4] [,5]
[1,]    -1    1    1    1    1
[2,]    -1   -1    1    1    1
[3,]    -1   -1   -1    1    1
[4,]    -1   -1   -1   -1    1
[5,]    -1   -1   -1   -1   -1

$`6` sp`
      [,1] [,2] [,3] [,4] [,5] [,6]
[1,]    -1    1    1    1    1    1
[2,]    -1   -1    1    1    1    1
[3,]    -1   -1   -1    1    1    1
[4,]    -1   -1   -1   -1    1    1
[5,]    -1   -1   -1   -1   -1    1
[6,]    -1   -1   -1   -1   -1   -1

```

Figure A2

These 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)
text<-c("a","b","c","d","e")

```

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 =
"darkslategrey"
}

```

Create the plot

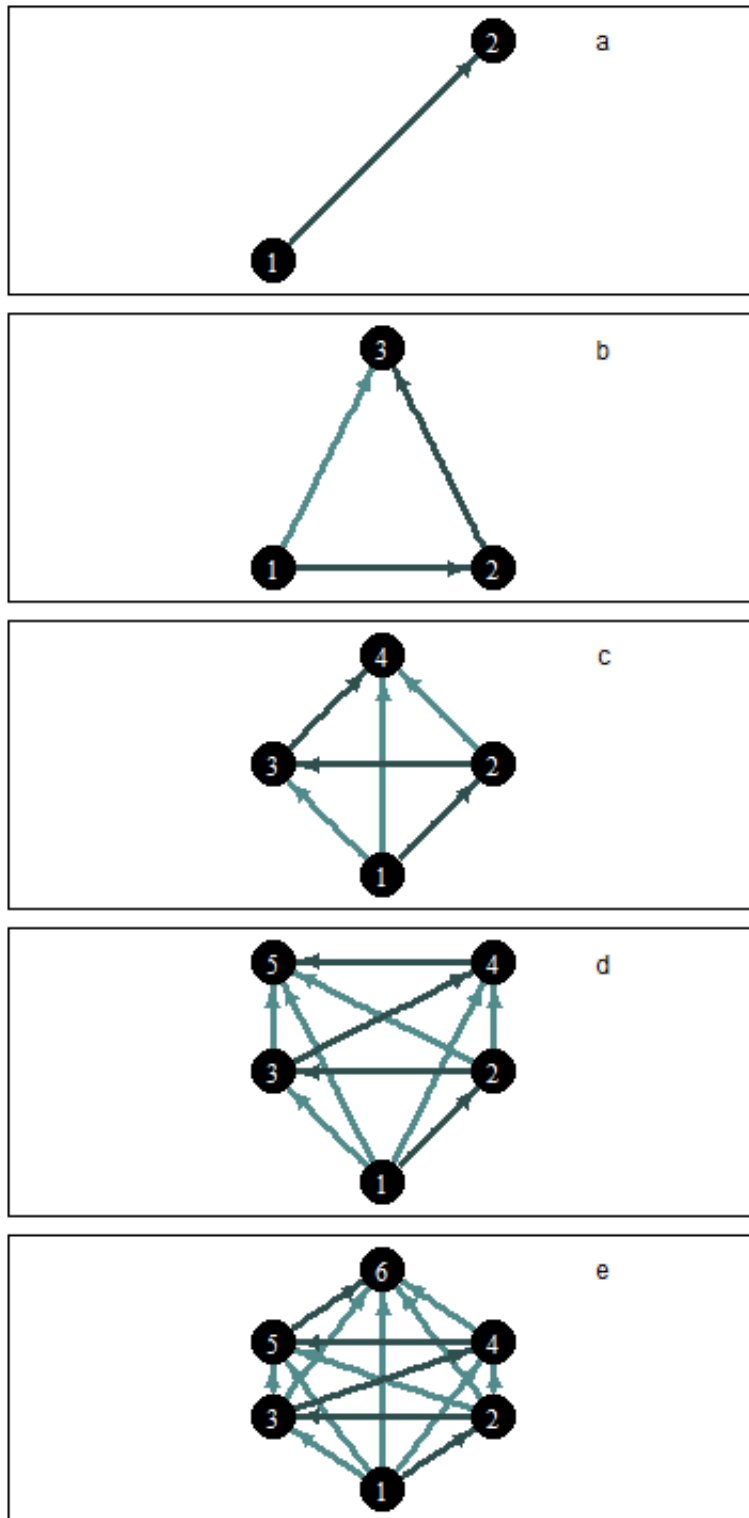
```

par(mfrow=c(5, 1),mar=c(.5, .5, .5, .5))
for(i in 1:5){
  plot.igraph(graph.chains[[i]], layout = layouts[[i]],
              vertex.size = 40,
              vertex.color = "black",
              vertex.label.color = "white",
              vertex.label.cex = 1.5,
              edge.width = 3,
              edge.arrow.size = .75,
              frame = T)
  text(2, 1, label = text[i], cex = 1.5)
}

```

Figure A2: Plot of each of the 5 food webs analyzed in the paper. The longest chain is highlighted in each case

```
par(mfrow=c(5, 1),mar=c(.5, .5, .5, .5))
for(i in 1:5){
  plot.igraph(graph.chains[[i]], layout = layouts[[i]],
              vertex.size = 40,
              vertex.color = "black",
              vertex.label.color = "white",
              vertex.label.cex = 1.5,
              edge.width = 3,
              edge.arrow.size = .75,
              frame = T)
  text(2, 1, label = text[i], cex = 1.5)
}
```



Defining the function to calculate stability

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.

```
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<-max(Re(eigen(m)$values))
  return(ev)
}
```

Simulation

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(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
```

```
$`2 sp`
[1] 1

$`3 sp`
[1] 0.8574

$`4 sp`
[1] 0.414

$`5 sp`
[1] 0.1017

$`6 sp`
[1] 0.0115
```

The results can then be plotted with `qss` as a function of the number of levels:

```
plot(unlist(qss)~c(2,3,4,5,6),  
     ylim=c(0,1), typ="o", lty=2, pch=21, bg="black",  
     xlab="Number of Trophic Levels",  
     ylab="QSS")
```

Sensitivity Analysis

Functions

For more indepth analysis of the impact of chain length, asymmetry of interaction strength, connectance, and density dependence on stability we used a slightly modified version of the functions used above.

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

The `eig.analysis` function takes an input 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){
  cols <- length(matrices)
  rows <- n
  eigenMATRIX <- matrix(nrow = rows, ncol = cols)
  for(i in 1:n){
    ranmat <- lapply(matrices, ran.unif, pred = params[,1],
                     prey = params[,2], random = T)
    eigs <- sapply(ranmat, maxRE)
    eigenMATRIX[i,] <- eigs
  }
  return(eigenMATRIX)
}
```

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 randomQSS function generates numweb random webs with total number of interactions. The chain argument determines the length of the initial chain the web is seeded with. Distributions for the relative impacts of predator and prey on each other are determined by the values in the params argument. For each random web max, mean, and median trophic position are computed along with quasi sign-stability. Quasi sign-stability is calculated by randomly parameterizing each matrix 1000 times and determining the proportion that are stable.

```

randomQSS <- function(numweb = 200, chain = 9, total = 14,
params){
  require(NetIndices)
  mywebs <- list()
  for(j in 1:numweb){
    check <- 1
    while(!check == 0){
      myweb <- matrix(0, nrow = 10, ncol = 10)
      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}
    }
  }

  mywebs1 <- lapply(mywebs, conversion)
  myweb.tl <- lapply(mywebs, TrophInd)
  emat <- eig.analysis(1000, mywebs1, params, mode =
"unif")
  qss <- apply(emat, 2, function(x){sum(x<0)/1000})
  mtl <- sapply(myweb.tl, max)

  return(list.webs = mywebs, trophics = myweb.tl, eigs =
emat, qss = qss, maxTL = mtl))
}

```

The testLENGTH function iterates through each possible seed chain length.

```

testLENGTH <- function(webiter = 200, maxchain = 9,
totalINT = 14, params){
  qssLIST <- list()
  mtlLIST <- list()
  troLIST <- list()
  for(i in 1:maxchain){
    cat(i, "\n")
    test <- randomQSS(numweb = webiter, chain = i, total =
totalINT, params = params)
    qssLIST[[i]] <- test$qss
    mtlLIST[[i]] <- test$maxTL
    troLIST[[i]] <- lapply(test$trophics, function(x)
{x$TL})
  }
  quas <- unlist(qssLIST)
  maxtl <- unlist(mtlLIST)
  meantl <- rapply(troLIST, mean)
  medtl <- rapply(troLIST, median)

  data <- data.frame(QSS = quas, MaxTL = maxtl, MeanTL =
meantl, MedTL = medtl)

  return(data)
}

```

Simulation

```

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)

```



```

totalDATA <- list()
for(i in 1:nrow(pars)){
  chainDATA <- data.frame(QSS = c(), MaxTL = c(), MeanTL =
c(), MedTL = c(), ints = factor())
  for(j in 1:length(ints)){
    chain <- testLENGTH(webiter = 100, maxchain = 9,
totalINT = ints[j], params = pars[i,])
    chain <- cbind(chain, ints = factor(rep(ints[j], 900)))
    chainDATA <- rbind(chainDATA, chain)
    cat("\n", ints[j], "is done", "\n")
  }
  totalDATA[[i]] <- chainDATA
  cat("\n", i, "th par done", "\n")
  # each item of the list corresponds to the row of
parameters
}

totalDATA2 <- list()
for(i in 1:9){
  totalDATA2[[i]] <- cbind(totalDATA[[i]], scenario =
paste(pars[i,], collapse = "/"))
}

head(totalDATA2[[1]])

totalDAT <- do.call(rbind, totalDATA2)

```

Figure S3

Figure S3: Plot of quasi sign-stability against mean trophic position. Each panel represents different degrees of asymmetry in the relative impacts of predators on their prey and vice versa.

```

ggplot(totalDAT, aes(x = MeanTL, y = QSS)) +
  geom_point(aes(col = ints)) +
  geom_smooth(aes(lty = ints), method = "glm") +
  facet_wrap(~scenario)

```

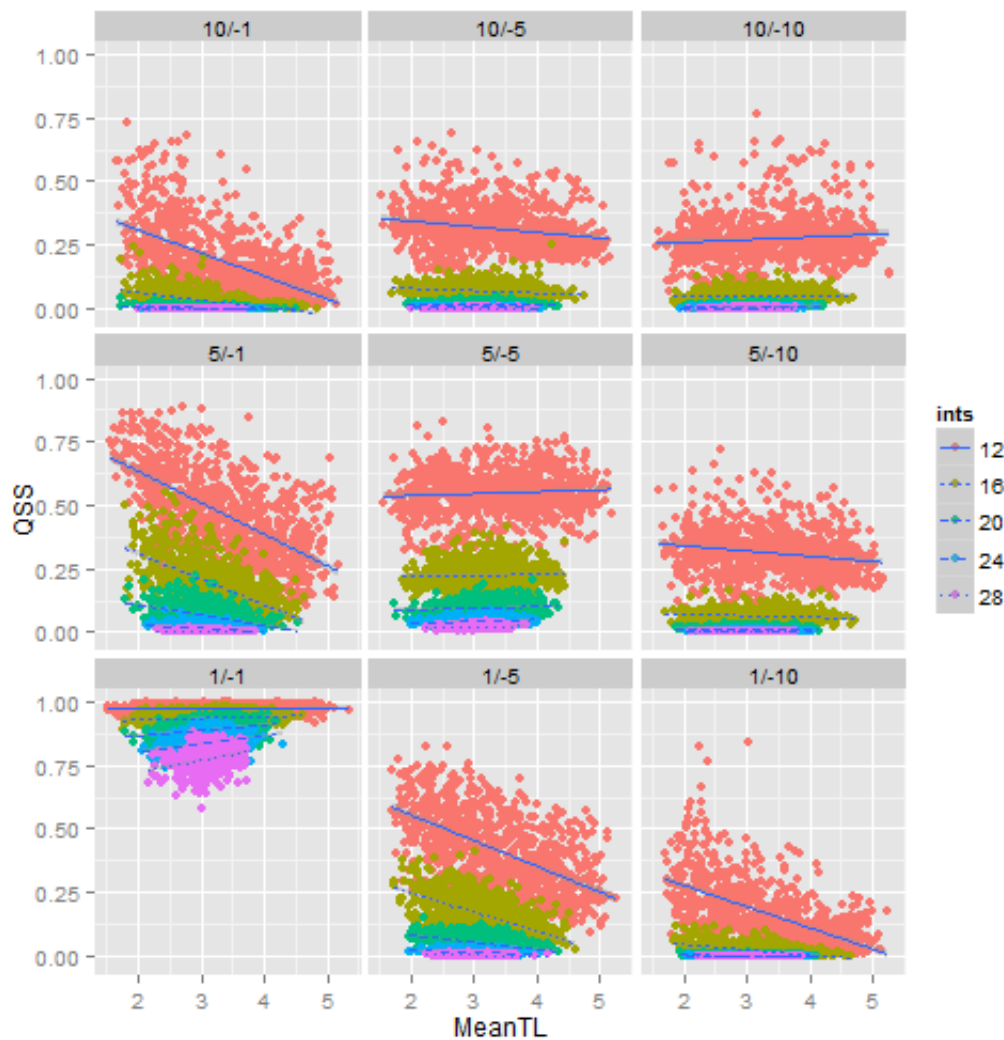


Figure S4

Figure S4: Plot of quasi sign-stability against median trophic position. Each panel represents different degrees of asymmetry in the relative impacts of predators on their prey and vice versa.

```
ggplot(totalDAT, aes(x = MedTL, y = QSS)) +
  geom_point(aes(col = ints)) +
  geom_smooth(aes(col = ints), method = "glm") +
  facet_wrap(~scenario)
```

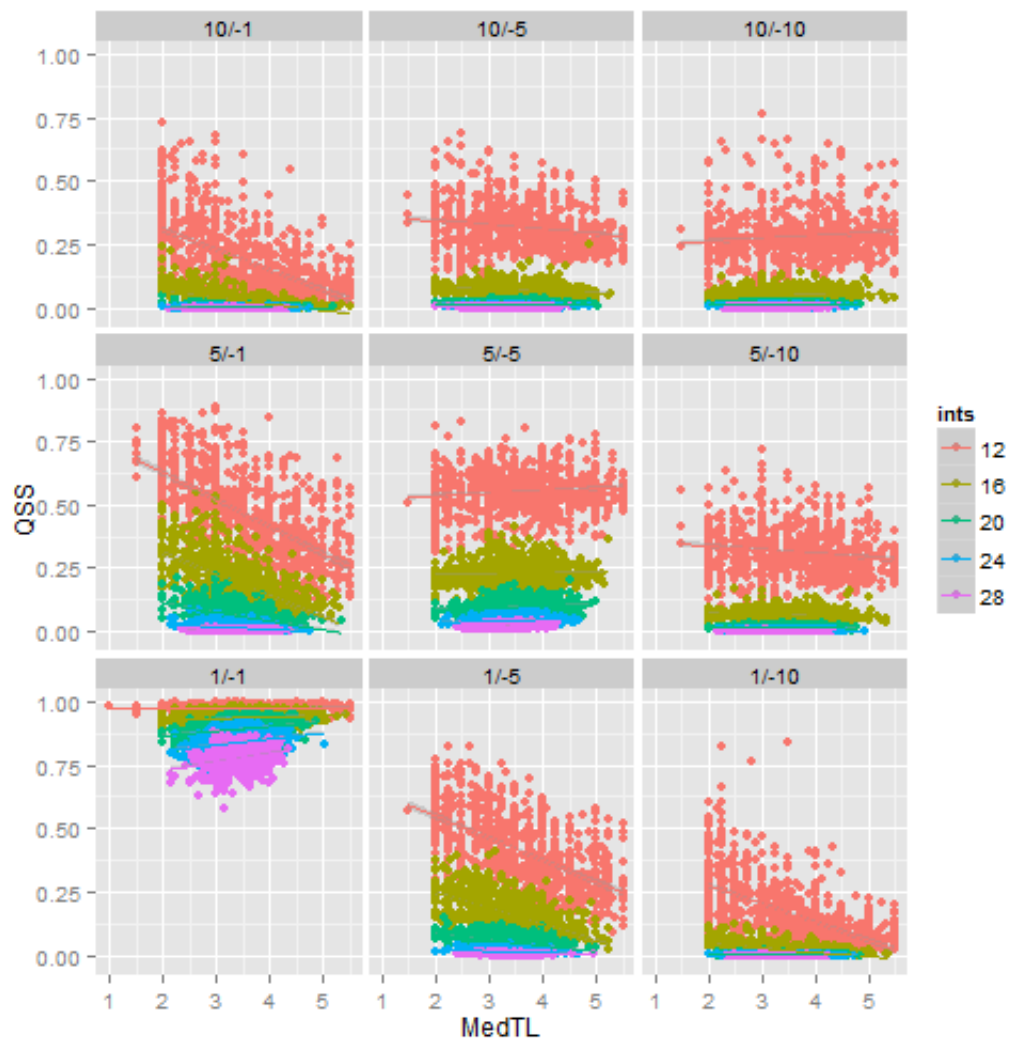
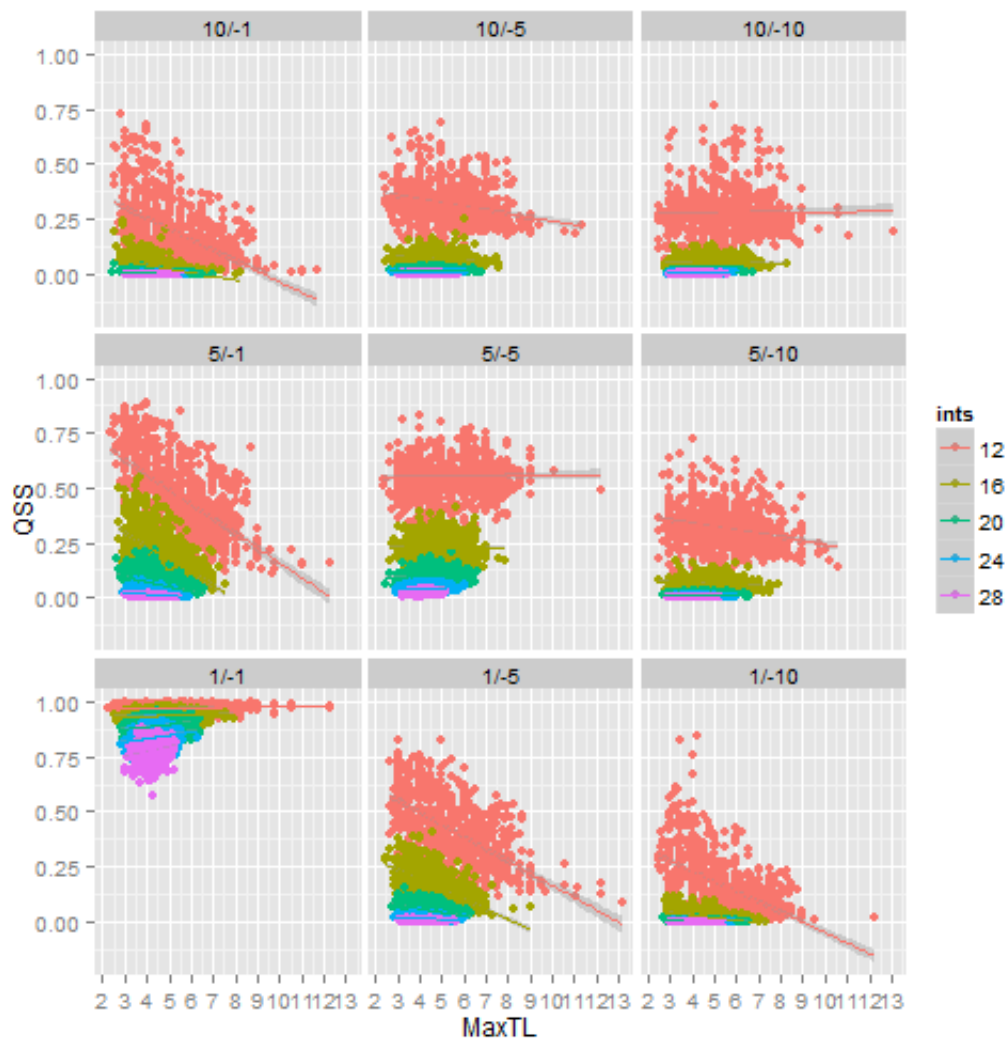


Figure S5

Figure S5: Plot of quasi sign-stability against maximum trophic position. Each panel represents different degrees of asymmetry in the relative impacts of predators on their prey and vice versa.

```
ggplot(totalDAT, aes(x = MaxTL, y = QSS)) +
  geom_point(aes(col = ints)) +
  geom_smooth(aes(col = ints), method = "glm") +
  facet_wrap(~scenario) +
  scale_x_continuous(breaks = 2:15)
```



Code for Figures in the Manuscript

Figure 1

Figure 1a

```
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, y = ..density..)) + theme_bw()  
  
tc.plot <- tc.plot + geom_histogram(binwidth = .8) +  
xlab("Trophic Position") + ylab("Density")  
tc.plot <- tc.plot + theme(axis.title.x = element_text(size  
= 25))  
tc.plot <- tc.plot + theme(axis.title.y = element_text(size  
= 25))  
tc.plot <- tc.plot + theme(axis.text.x = element_text(size  
= 18))  
tc.plot <- tc.plot + theme(axis.text.y = element_text(size  
= 18))  
tc.plot + scale_x_continuous(breaks = 2:6, name = "Trophic  
Position") + scale_y_continuous(name = "Density")
```

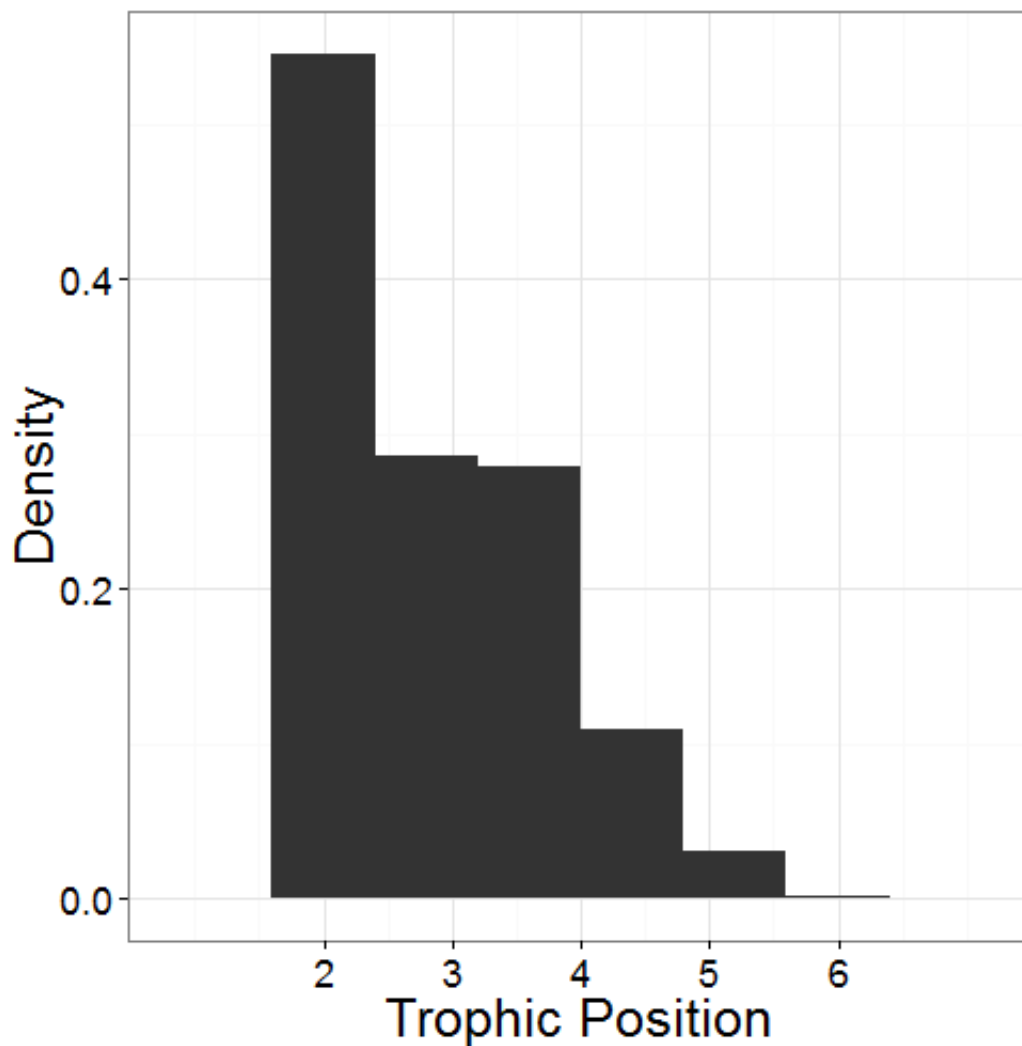


Figure 1b

```
qss.plot <- qplot(2:6, unlist(qss), xlab = "Number of  
Trophic Levels", ylab = "Quasi Sign-  
Stability", margin = T)  
qss.plot <- qss.plot + geom_point(size = 4)  
qss.plot <- qss.plot + geom_line() + theme_bw()  
  
qss.plot <- qss.plot + theme(axis.title.x =  
element_text(size=22))  
qss.plot <- qss.plot + theme(axis.title.y =  
element_text(size=22))  
qss.plot <- qss.plot + theme(axis.text.x =  
element_text(size=18))  
qss.plot <- qss.plot + theme(axis.text.y =  
element_text(size=18))
```

```
qss.plot
```

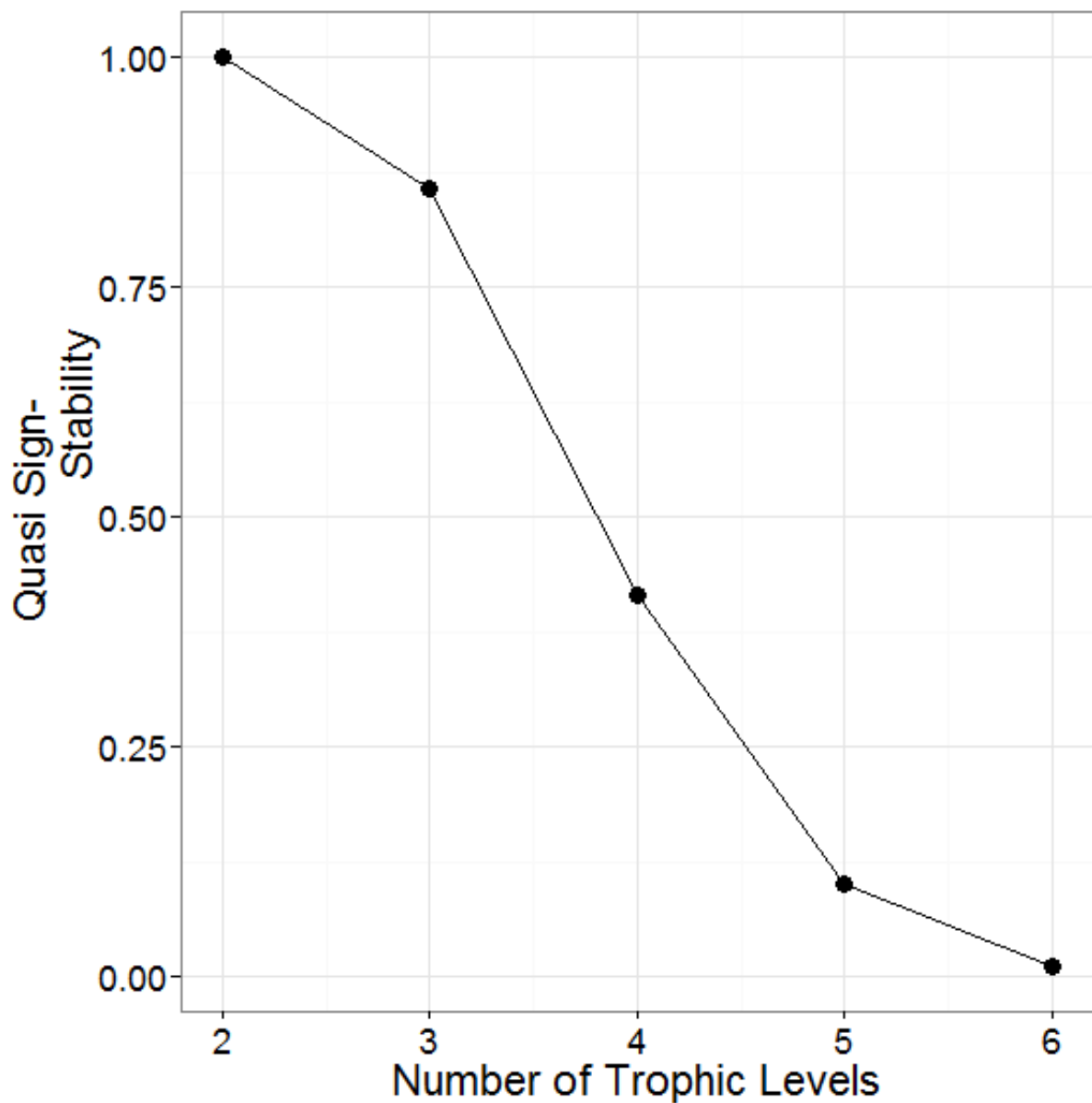


Figure 2

```
1 <- levels(totalDAT$ints)[c(1,2,3)]
ggplot(totalDAT[totalDAT$ints == 1,], aes(x = MeanTL, y =
QSS)) +
  geom_point(aes(shape = ints), size = 1.5, alpha = .75) +
  geom_smooth(aes(lty = ints), lwd = 1, method = "glm") +
  theme_bw() + xlab("Mean Trophic Postion") + ylab("Quasi
Sign-Stability") +
  facet_wrap(~scenario)
```

