Empirical Dynamic Modeling for Beginners

Electronic Supplementary Materials for Ecological Research

Chun-Wei Chang1,2,\*, Masayuki Ushio3,4,\*, and Chih-hao Hsieh1,5,6,7,8,\*

1Taiwan International Graduate Program (TIGP)–Earth System Science Program, Academia Sinica, Taipei 11529, Taiwan

2Taiwan International Graduate Program (TIGP)–Earth System Science Program, National Central University, Taoyuan 32001, Taiwan

3PRESTO, Japan Science and Technology Agency, Kawaguchi 332-0012, Japan

4Center for Ecological Research, Kyoto University, Otsu 520-2113, Japan

5Institute of Oceanography, National Taiwan University, Taipei 10617, Taiwan

6Research Center for Environmental Changes, Academia Sinica, Taipei 11529, Taiwan

7Institute of Ecology and Evolutionary Biology, Department of Life Science, National Taiwan University, Taipei 10617, Taiwan

8National Center for Theoretical Sciences, Taipei 10617, Taiwan

\* Corresponding author: Chun-Wei Chang ([clark.ms37@yahoo.com.tw](mailto:clark.ms37@yahoo.com.tw)), or Masayuki Ushio ([ong8181@gmail.com](mailto:ushio@ecology.kyoto-u.ac.jp)), or Chih-hao Hsieh ([chsieh@ntu.edu.tw](mailto:chsieh@ntu.edu.tw))

Supplementary information 1: Simplex projection and S-map analysis

Load package and time series

All the EDM analyses are carried out by the **rEDM** package. The red noise time series is generated by a difference equation specifying the value at next time step as proportional to the current value plus a random number drawn from the standard normal distribution. The logistic map time series is generated by a self-regulatory difference equation following Hsieh et al. (2005). Both models are simulated for 10000 time steps, but only the last 1000 data are used for further analyses in order to exclude the transient dynamics.

## loading R package: rEDM

**library**(rEDM)

Simplex projection (Sugihara & May 1990)

By simplex projection, we make one-step forward forecast (predict t+1 step) for the red noise or logistic map dynamics. Prior to simplex projection, the time series are normalized to zero mean and unit variance. The simplex projection is carried out by the function **rEDM::simplex()**. The object Red in the dataset is the normalized time series of red noise; the object Logi is the normalized time series of the logistic map. We divide the time series into two halves. The first half is used as the library set for manifold reconstruction. The second half is used as the target for out-of-sample prediction. The argument lib is the time index indicating the start (1) and the end (500) in the library set, respectively. Similarly, the argument pred indicates the time index in the prediction set. Then, we specify a sequence of testing embedding dimensions from 2 to 8 by the argument E=c(2:8)which executes the simplex projections using different embedding dimensions. Finally, we present the results showing the relationship between the predictive skill (*ρ*) and the embedding dimension (*E*) (Fig. 2c & d in the main text).

## Data loading

dat <- **read.csv**('ESM2\_Data\_noise.csv',header=T)

## Data normalization

Red <- ((dat[,"R"]-**mean**(dat[,"R"]))/**sd**(dat[,"R"]))

Logi <- ((dat[,"L"]-**mean**(dat[,"L"]))/**sd**(dat[,"L"]))

#######################################################################

## Simplex projection for red noise and logistic map

sim\_r <- **simplex**(Red,lib=c(1,500),pred=c(501,1000),E=**c**(2:8))

sim\_l <- **simplex**(Logi,lib=c(1,500),pred=c(501,1000),E=**c**(2:8))

## Plot predictive skill (rho) vs embedding dimension (E)

**par**(mfrow=c(2,1),mar=c(4,4,1,1))

**plot**(rho~E,data=sim\_r,type="l",xlab="Embedding dimension (E)",ylab=**expression**(rho),ylim=**c**(0.3,0.4),col=2,main="Red noise")

**plot**(rho~E,data=sim\_l,type="l",xlab="Embedding dimension (E)",ylab=**expression**(rho),ylim=c(0.95,1.02),col=4,main="Logistic map")

## The optimal embedding dimension determined by maximizing rho

(E\_r <-sim\_r[**which.max**(sim\_r$rho),"E"][1])# The optimal E of red noise

(E\_l <-sim\_l[**which.max**(sim\_l$rho),"E"][1])# The optimal E of logistic map

S-map analysis

Using the function **rEDM::s\_map()**, we calculate the one-step forward forecast by S-map analysis. Again, we divide the time series into two halves, one for library, lib=c(1,500), and another for out-of-sample prediction, pred=c(501,1000). The embedding dimensions, E=E\_r for red noise and E=E\_l for the logistic map, have been already determined by simplex projection (Fig. 2c & d) in the previous section. Then, we specify a sequence of testing state-dependency parameters *θ* from 0 to 2 with an increment of 0.1 by the argument, theta=seq(0,2,0.1)which executes S-map by trial-and-error using different *θ*. Finally, we demonstrate the results showing the relationship between the predictive skills (*ρ*) and state-dependency parameters (*θ*)(Fig. 2e & f). Again, the criterion maximizing predictive skill is applied to determine the optimal *θ*.

# S map for Red Noise & logistic map

smap\_r <- **s\_map**(Red,E=E\_r,lib=c(1,500),pred=c(501,1000),theta=**seq**(0,2,0.1))

smap\_l <- **s\_map**(Logi,E=E\_l,lib=c(1,500),pred=c(501,1000),theta=**seq**(0,2,0.1))

## Plot predictive skill (rho) vs state-dependency parameter (theta)

**plot**(rho~theta,data=smap\_r,type="l",xlab=**expression**(theta),ylab=**expression**(rho),ylim=c(0.4,0.5),col=2,main="Red noise")

**plot**(rho~theta,data=smap\_l,type="l",xlab=**expression**(theta),ylab=**expression**(rho),ylim=c(0.6,1),col=4,main="Logistic map")

## The optimal theta determined by maximizing rho

(the\_r <- smap\_r[**which.max**(smap\_r$rho),"theta"][1])

(the\_l <- smap\_l[**which.max**(smap\_l$rho),"theta"][1])

References

Hsieh C.H, Glaser S.M., Lucas A.J. & Sugihara G. (2005). Distinguishing random environmental fluctuations from ecological catastrophes for the North Pacific Ocean. Nature, 435, 336-340.

Supplementary information 2: Determining causal variables by convergent cross mapping (CCM)

Load package and time series

We demonstrate the efficacy of CCM to correctly identify causation using the Moran effect model and the two-species competition model, following Sugihara et al. (2012). In the Moran effect model, we simulate adult-recruitment dynamics as commonly used in fisheries. In this model, two populations do not have any biological interaction, while both are driven by the same environmental factor. Despite that no interaction exists, the shared environmental driver leads to strong correlation between the two populations (Fig. 1b). In the two-species competition model, we find no lasting correlation between the two species, as the sign of correlation flips through time (Fig. 1d), which is an example of mirage correlation, a hallmark of nonlinear systems. Both models are simulated for 10000 time steps, but only the last 1000 data are kept for further analyses in order to exclude the transient dynamics.

# Loadi**n**g R packages

library(rEDM)

library(Kendall)

# Loading the time series for the Moran effect and mirage correlation models

dam <- read.csv('ESM3\_Data\_moran.csv',header=T) # Moran effect

dac <- read.csv('ESM4\_Data\_competition.csv',header=T) # Mirage correlation

# Data normalization

dac.n <- scale(dac[,-1], center = TRUE, scale = TRUE)

dam.n <- scale(dam[,-1], center = TRUE, scale = TRUE)

In the rEDM package, the function **rEDM::ccm()** is used for CCM analyses. Here, we illustrate how to implement the CCM causality test that examines whether *N2* causes *N1* in Moran effect model. To test this, we design a cross-mapping from *N1* to *N2* by the augument lib\_column="N1" and target\_column="N2". In CCM analysis, we firstly need to determine the best embedding dimension for the cross-mapping. At this step, we perform the cross-mapping with a fixed library size (lib\_sizes = 1000). Then, we use the time lags of *N*1 to predict the lagged one time step values of *N*2 by setting the augment tp=-1 and determine the optimal *E* based on the hindcast skill to avoid over-fitting (Deyle et al. 2016). Similarly, we can repeat the same process to determine the embedding dimension in the cross-mapping from *N*2 to *N*1. Next, we carried out the CCM causality test with varying library size.

libs <- c(seq(20,80,20),seq(100,1000,100))

In CCM analysis, we use the time lags of *N*1 to predict the current value of *N*2 (tp=0). To precisely estimate the predictive skill (*ρ*), we generate the 200 random samples with replacement (replace=T) for each library length *L* (num\_samples=200). As such, we obtain the sampling distribution of predictive skill *ρ*(*L*). A random seed is setup to make the results repeatable (RNGseed=2301). Finally, we offer a simple statistical test for the convergence of CCM by Mann-Kendall Tau trend test when the null time series is not easily accessible. This is a nonparametric test for the existence of monotonic increasing trend, using the function **Kendall::MannKendall()**. Practically, we can evaluate the significance of causations by examining whether all the quantiles of predictive skill demonstrate a significant increasing trend with increasing library size (*τ* statistics significantly > 0). Similarly, we can repeat all these procedures of CCM analysis to test the causality for the two species competition model with mirage correlation.

## CCM analysis of the Moran effect model, N1 and N2

# Design a sequence of library size

libs <- **c**(**seq**(20,80,20),**seq**(100,1000,100))

# Moran effect model: N1 cross-mapping N2 (i.e. testing N2 as a cause of N1)

# Determine the embedding dimension

E.test.n1=NULL

**for**(E.t in 2:8){

cmxy.t <- **ccm**(dam.n, E = E.t, lib\_column = "N1", target\_column = "N2",

lib\_sizes = 1000, num\_samples = 1, tp=-1,random\_libs = F)

E.test.n1=**rbind**(E.test.n1,cmxy.t)}

(E\_n1 <- E.test.n1$E[**which.max**(E.test.n1$rho)[1]]) # the optimal E

# CCM analysis with varying library size (L)

n1\_xmap\_n2 <- **ccm**(dam.n, E=E\_n1,lib\_column="N1", target\_column="N2",

lib\_sizes=libs, num\_samples=200, replace=T, RNGseed=2301)

*# Calculate the median, maximum, and 1st & 3rd quantile of rho for each L*

n12q=**as.matrix**(**aggregate**(n1\_xmap\_n2[,c('rho')],by = **list**(**as.factor**(n1\_xmap\_n2$lib\_size)), **quantile**)[,'x'])

**apply**(n12q[,2:5],2,**MannKendall**)

###########################################################

*# Moran effect model: N2 cross-mapping N1* (i.e. testing N1 as a cause of N2)

*# Determine the embedding dimension*

E.test.n2=NULL

**for**(E.t in 2:8){

cmxy.t <- **ccm**(dam.n, E = E.t, lib\_column = "N2", target\_column = "N1",

lib\_sizes = 1000, num\_samples = 1, tp=-1, random\_libs = F)

E.test.n2=**rbind**(E.test.n2,cmxy.t)}

(E\_n2 <- E.test.n2$E[**which.max**(E.test.n2$rho)[1]])

*# CCM analysis*

n2\_xmap\_n1 <- **ccm**(dam.n, E=E\_n2,lib\_column="N2", target\_column="N1",

lib\_sizes=libs, num\_samples=200, replace=T, RNGseed=2301)

*# Calculate the (25%,50%,75%,100%) quantile for predictive skills*

n21q=**as.matrix**(**aggregate**(n2\_xmap\_n1[,c('rho')],by = **list**(**as.factor**(n2\_xmap\_n1$lib\_size)), **quantile**)[,'x'])

**apply**(n21q[,2:5],2,**MannKendall**)

*# Plot forecast skill vs library size*

# Plot N1 cross-mapping N2

**plot**(n12q[,3]~libs,type="l",col="red",ylim=c(0,1),lwd=2,

main="Convergent cross mapping CCM",xlab="Library size",ylab=**expression**(rho)) # median predictive skill vs library size (or we can use mean predictive skill)

**lines**(n12q[,2]~libs,col="red",lwd=1,lty=2) # 1st quantile

**lines**(n12q[,4]~libs,col="red",lwd=1,lty=2) # 3rd quantile

# Plot N2 cross-mapping N1

**lines**(n21q[,3]~libs,col="blue",lwd=1,lty=1) # median

**lines**(n21q[,2]~libs,col="blue",lwd=1,lty=2) # 1st quantile

**lines**(n21q[,4]~libs,col="blue",lwd=1,lty=2) # 3rd quantile

**legend**(600,1,c("N1 xmap N2","N2 xmap N1"),lty=c(1,1),col=c("red","blue"))

**abline**(h=**cor**(dam[,'N1'],dam[,'N2']),lty=3)

Following the same procedure, we apply CCM to test the mutual causation between the two competitors exhibiting mirage correlations.

#########################################################################

#########################################################################

## CCM analysis of the two species competition model with mirage correlation

# Design a sequence of library size

libs <- **c**(**seq**(20,80,20),**seq**(100,1000,100))

# Mirage correlation model: M1 cross-mapping M2 (i.e. testing M2 as a cause of M1)

# Determine the embedding dimension

E.test.x=NULL

**for**(E.t in 2:8){

cmxy.t <- **ccm**(dac.n, E = E.t, lib\_column = "M1", target\_column = "M2",

lib\_sizes = 1000, num\_samples = 1, tp=-1,random\_libs = F)

E.test.x=**rbind**(E.test.x,cmxy.t)}

(E\_x <- E.test.x$E[**which.max**(E.test.x$rho)[1]])

# CCM analysis: varying library size

x\_xmap\_y <- **ccm**(dac.n, E=E\_x,lib\_column="M1", target\_column="M2",

lib\_sizes=libs, num\_samples=200, replace=T, RNGseed=2301)

*# Calculate the median, maximum, and 1st & 3rd quantiles of rho*

xyq=**as.matrix**(**aggregate**(x\_xmap\_y[,c('rho')],by = **list**(**as.factor**(x\_xmap\_y$lib\_size)), **quantile**)[,'x'])

**apply**(xyq[,2:5],2,**MannKendall**)

###########################################################

*# Mirage correlation model: M2 cross-mapping M1* (i.e. testing M1 as a cause of M2)

*# Determine the embedding dimension*

E.test.y=NULL

**for**(E.t in 2:8){

cmxy.t <- **ccm**(dac.n, E = E.t, lib\_column = "M2", target\_column = "M1",

lib\_sizes = 1000, num\_samples = 1,tp=-1,random\_libs = F)

E.test.y=**rbind**(E.test.y,cmxy.t)}

(E\_y <- E.test.y$E[**which.max**(E.test.y$rho)[1]])

*# CCM analysis*

y\_xmap\_x <- **ccm**(dac.n, E=E\_y,lib\_column="M2", target\_column="M1",

lib\_sizes=libs, num\_samples=200, replace=T, RNGseed=2301)

*# Calculate the (25%,50%,75%,100%) quantile for predictive skills*

yxq=**as.matrix**(**aggregate**(y\_xmap\_x[,c('rho')],by = **list**(**as.factor**(y\_xmap\_x$lib\_size)), **quantile**)[,'x'])

**apply**(yxq[,2:5],2,**MannKendall**)

*# Plot forecast skill vs library size*

# Plot X cross-mapping Y

**plot**(xyq[,3]~libs,type="l",col="red",ylim=c(0,1),lwd=2,

main="Convergent cross mapping CCM",xlab="Library size",ylab=**expression**(rho)) # median predictive skill vs library size (or we can use mean predictive skill)

**lines**(xyq[,2]~libs,col="red",lwd=1,lty=2) # 1st quantile

**lines**(xyq[,4]~libs,col="red",lwd=1,lty=2) # 3rd quantile

# Plot Y cross-mapping X

**lines**(yxq[,3]~libs,col="blue",lwd=1,lty=1) # median

**lines**(yxq[,2]~libs,col="blue",lwd=1,lty=2) # 1st quantile

**lines**(yxq[,4]~libs,col="blue",lwd=1,lty=2) # 3rd quantile

**legend**(600,0.4,c("M1 xmap M2","M2 xmap M1"),lty=c(1,1),col=c("red","blue"))

**abline**(h=**cor**(dac[,'M1'],dac[,'M2']),lty=3)

References

Deyle ER, Maher MC, Hernandez RD, Basu S, Sugihara G (2016) Global environmental drivers of influenza. Proc Natl Acad Sci USA 113: 13081-13086. DOI: 10.1073/pnas.1607747113

Sugihara G, May R, Ye H, Hsieh CH, Deyle E, Fogarty M, Munch S (2012) Detecting causality in complex ecosystems. Science 338: 496-500. DOI: 10.1126/science.1227079

Supplementary information 3: Univariate, multivariate and multi-view embbedings

Load package and time series

Time series are generated according to a 5-species model (Resource, Consumer1, Consumer2, Predator1, Predator2) following Deyle et al. (2016). Note that the model is started with a burn-in period to ensure the dynamics relax to attractor manifold (see Deyle et al. 2016).

## Load package and data  
library(rEDM)  
d <- read.csv("ESM5\_Data\_5spModel.csv")

Univariate embedding (Sugihara & May 1990)

In this demonstration, we want to forecast dynamics of Consumer1 (*C*1). Information (history) encoded in the dynamics of *C*1 is used to forecast the dynamics of *C*1.

# Please reduce the number of data points if the calculation needs long time  
data\_used <- 1:1000  
  
# Specify the length of time series to be used to reconstruct state space (Library length)  
lib\_point <- c(1,floor(max(data\_used)/2))  
  
# Specify which points will be predicted based on the reconstructed state space  
pred\_point <- c(floor(max(data\_used)/2)+1, max(data\_used))  
  
# Time series of C1 is normalized  
C1 <- as.numeric(scale(d[data\_used,'C1']))  
  
# Estimate the best embedding dimension  
simp\_C1\_tmp <- simplex(C1, E=1:10, silent = T)  
plot(simp\_C1\_tmp$E, simp\_C1\_tmp$mae, type="l", xlab="E", ylab="MAE")  
  
# Best E = 3  
bestE\_C1 <- simp\_C1\_tmp[which.min(simp\_C1\_tmp$mae),"E"]  
  
# Perform univariate simplex projection  
# We need to specify time series (C1), embedding dimension (E), library length (lib), predictee (pred) and which output we need (stats\_only). If you do not want to see warning message, "silent" option should be set as "T".  
simp\_C1 <- simplex(C1, E=bestE\_C1, lib=lib\_point, pred=pred\_point, stats\_only = F, silent = T)  
C1\_pred\_uni <- simp\_C1[[1]]$model\_output[“pred”]  
C1\_obs\_uni <- simp\_C1[[1]]$model\_output$obs  
plot(C1\_obs\_uni, C1\_pred\_uni, xlab="Observed", ylab="Predicted")  
abline(0,1) # add 1:1 line

The forecast skill (*ρ*) is 0.970 when we use univariate embedding (Fig. 6a in the main text).

Multivariate embedding (Deyle & Sugihara 2011, Deyle et al. 2013)

In this demonstration, we want to forecast the dynamics of *C*1. In the 5-species model, *P1* and *R* are directly related to *C*1. Thus, we used *C*1, *R* and *P1* for multivariate embedding.

# Make multivariate embedding  
Embedding <- c("C1", "R", "P1")  
block <- d[,Embedding]

# Normalize data  
block <- as.data.frame(apply(block, 2, function(x) (x-mean(x))/sd(x)))  
  
# Do multivariate simplex projection using block\_lnlp() function  
# We need to specify time series, method (simplex or s-map), library length (lib), predictee (pred) and which output we need (stats\_only).  
mult\_simp\_C1 <- block\_lnlp(block[data\_used,], method = "simplex", lib = lib\_point, pred = pred\_point,  
 stats\_only = F, silent = T)  
C1\_pred\_mult <- mult\_simp\_C1[[1]]$model\_output$pred  
C1\_obs\_mult <- mult\_simp\_C1[[1]]$model\_output$obs  
plot(C1\_obs\_mult, C1\_pred\_mult, xlab="Observed", ylab="Predicted")  
abline(0,1) # add 1:1 line

The forecast skill (*ρ*) is 0.987 when we use maltivariate embedding (Fig. 6b).

Multi-view embedding (Ye & Sugihara 2016)

Multi-view embedding combines multiple embeddings and leverage information of many embeddings.

# Do multiview forecasting using multiview() function  
# We need to specify time series, library length (lib), predictee (pred) and which output we need (stats\_only).  
multiview\_C1 <- multiview(block[data\_used,], lib = lib\_point, pred = pred\_point, stats\_only = F, silent = T)  
C1\_pred\_multv <- multiview\_C1[[1]]$model\_output$pred  
C1\_obs\_multv <- multiview\_C1[[1]]$model\_output$obs  
plot(C1\_obs\_multv, C1\_pred\_multv, xlab="Observed", ylab="Predicted")  
abline(0,1) # add 1:1 line

The forecast skill (*ρ*) is 0.989 when we use multiview embedding (Fig, 6c).

Compare univariate, multivariate, and multiview embeddings

Forecasting skill improves if we use multivariate or multiview embedding compared with univariate embedding (Fig. 6d).

rhos <- c(simp\_C1[[1]]$stats$rho, mult\_simp\_C1[[1]]$stats$rho, multiview\_C1[[1]]$pred\_stats$rho)  
names(rhos) <- c("Univariate", "Multivariate", "Multiview")

# Plot result  
barplot(rhos, xlab="Methods", xpd = F, ylab=expression(paste("Forecasting skill (", rho, ")")), ylim=c(0.96, 1)); box()

References

Deyle E, Sugihara G (2011) Generalized theorems for nonlinear state space reconstruction. PLoS ONE 6: e18295.

Deyle ER, Fogarty M, Hsieh CH, Kaufman L, MacCall AD, Munch SB, Perretti CT, Ye H, Sugihara G (2013) Predicting climate effects on Pacific sardine. Proc Natl Acad Sci USA 110: 6430-6435. DOI: 10.1073/pnas.1215506110

Sugihara G, May R, Ye H, Hsieh CH, Deyle E, Fogarty M, Munch S (2012) Detecting causality in complex ecosystems. Science 338: 496-500. DOI: 10.1126/science.1227079

Ye H, Sugihara G (2016) Information leverage in interconnected ecosystems: Overcoming the curse of dimensionality. Science 353: 922.

Supplementary information 4: Tracking interaction strength using S-map

Load package and time series

In this demonstration, we use the same time series generated from the 5-species model (Resource, Consumer1, Consumer2, Predator1, Predator2) following Deyle et al. (2016).

## Load package and data  
library(rEDM)   
d <- read.csv("ESM5\_Data\_5spModel.csv")

Set parameters and do S-map

In this demonstration, we focus on the effects on Consumer1 (*C*1). As shown in Deyle et al. (2016), the effects of Predator2 (*P*2) on *C*1 are negligible, and thus we ignore *P*2 in the embedding. We use fully multivariate embedding (Deyle et al. 2016, Deyle et al. 2011) in order to investigate effects of *R*, *C2*, and *P*1 on *C*1.

# Make multivariate embedding  
Embedding <- c("R","C1","C2","P1")  
block <- d[,Embedding]

# Normalize data  
block <- as.data.frame(apply(block, 2, function(x) (x-mean(x))/sd(x)))  
  
# Define the target column (C1 = column 2)  
targ\_col <- 2  
  
# Please reduce the number of data points if the calculation takes long  
data\_used <- 1:2000  
  
# Explore the best weighting parameter (nonlinear parameter = theta)  
# Best theta is selected based on mean absolute error (MAE)  
test\_theta <- block\_lnlp(block[data\_used,],  
 method = "s-map",  
 num\_neighbors = 0, # We have to use any value < 1 for s-map  
 theta = c(0, 1e-04, 3e-04, 0.001,  
 0.003, 0.01, 0.03, 0.1,  
 0.3, 0.5, 0.75, 1, 1.5,  
 2, 3, 4, 6, 8), # We try many thetas to find the best parameter  
 target\_column = targ\_col, # Specify the target column  
 silent = T)  
  
# Check MAE and theta  
plot(test\_theta$mae~test\_theta$theta, type="l", xlab="Theta", ylab="MAE")  
  
# Best theta = 8 in this case  
best\_theta <- test\_theta[which.min(test\_theta$mae),"theta"]  
  
# Do S-map analysis with the best theta  
smap\_res <- block\_lnlp(block[data\_used,],  
 method = "s-map",  
 num\_neighbors = 0, # we have to use any value < 1 for s-map  
 theta = best\_theta,  
 target\_column = targ\_col,  
 silent = T,  
 save\_smap\_coefficients = T) # save S-map coefficients

#### Visualize results  
## Observed v.s. Predicted  
smap\_out <- as.data.frame(smap\_res[[1]]$model\_output)  
plot(smap\_out$obs, smap\_out$pred, xlab="Observed", ylab="Predicted")

## Time series of fluctuating interaction strength  
smap\_coef <- as.data.frame(smap\_res[[1]]$smap\_coefficients)  
colnames(smap\_coef) <- c("dC1dR","dC1dC1","dC1dC2","dC1dP1","Intercept")

# Plot all partial derivatives  
trange <- 1:200  
quartz(width=6, height=4)  
plot(smap\_coef[trange,"dC1dR"],  
 type="l", col="royalblue", lwd=2, xlab="time",  
 ylab="Interction strength", ylim = c(-1.0, 2.5),  
 main = "Fluctuating interaction strength")  
lines(smap\_coef[trange,"dC1dC2"], lwd=2, col="red3")  
lines(smap\_coef[trange,"dC1dP1"], lwd=2, col="springgreen3")  
abline(a=0 ,b=0 , lty="dashed", col="black", lwd=.5)

As can be seen in Figure 7 in the main text, interaction strengths fluctuate with time.

References

Deyle E, Sugihara G (2011) Generalized theorems for nonlinear state space reconstruction. PLoS ONE 6: e18295.

Deyle ER, May RM, Munch SB, Sugihara G (2016) Tracking and forecasting ecosystem interactions in real time. Proc R Soc Lond B Biol Sci 283: 20152258.

Supplementary information 5: Scenario exploration

Load package and time series

In this demonstration, we use the same time series generated from the 5-species model (Resource, Consumer1, Consumer2, Predator1, Predator2) following Deyle et al. (2016).

## Load package and data  
library(rEDM)   
d <- read.csv("ESM5\_Data\_5spModel.csv")

Set parameters

In this demonstration, we focus on the dynamics of Consumer1 (*C*1). We investigate how changes in Resource (*R*) induce changes in *C*1 using scenario exploration (Deyle et al. 2013). First, we determine the best embedding dimension of *C*1 using simplex projection

# Normalize data  
std\_C1 <- as.numeric(scale(d[,"C1"]))  
  
# Determine the best embedding dimension by simplex projection  
E\_tested <- simplex(std\_C1, E=1:10, silent=T)  
  
 # The best E is determined based on MAE  
E\_C1 <- E\_tested[which.min(E\_tested$mae),"E"]

The, we generate data frame for scenario exploration by adding one dimension (i.e., R(t)) to the univariate embedding. The multivariate embedding follows:

# Normalize data  
std\_R <- as.numeric(scale(d[,"R"]))  
  
# Make time-delayed embedding  
embed\_C1 <- embed(std\_C1, dimension = E\_C1)  
  
# Add R as an external force  
block0 <- cbind(embed\_C1, std\_R[E\_C1:length(std\_R)])

We prepare a new "R" column that includes simulated changes in Resource (0.5\*sd(R) increase or decrease in Resource). The magnitude of change is arbitrarily defined here. The users can decide the magnitude of change, depending on their own research questions.

# Make new R (increased/decreased situation)  
std\_R\_increase <- std\_R + 0.5  
std\_R\_decrease <- std\_R - 0.5  
  
# Make new embeddings (increased/decreased situation)  
block\_inc0 <- cbind(embed\_C1, std\_R\_increase[E\_C1:length(std\_R)])  
block\_dec0 <- cbind(embed\_C1, std\_R\_decrease[E\_C1:length(std\_R)])  
  
# Combine the original and simulated time series  
block\_inc <- rbind(block0, rep(NaN,4), block\_inc0) # NaNs were added to separate two data  
block\_dec <- rbind(block0, rep(NaN,4), block\_dec0) # NaNs were added to separate two data  
colnames(block\_inc) <- colnames(block\_dec) <- c("C1\_t","C1\_t\_1","C1\_t\_2","R\_t")

Do scenario exploration using simplex projection

In this demonstration, we use simplex projection to forecast the effects of changing resource (*R*).

# Normal multivariate simplex projection  
scenario\_res <- block\_lnlp(block0,  
 method = "simplex",  
 columns = 1:4, # columns used to reconstruct the state space  
 target\_column = 1,  
 stats\_only = F,  
 silent = T)  
  
# "Resource increased" situation  
# Attractor is reconstructed using the original time series, but the dynamics is forecasted based on the resource-increased situation  
scenario\_inc <- block\_lnlp(block\_inc,  
 lib = c(1,nrow(block0)),  
 pred = c((nrow(block0)+2),nrow(block\_inc)),  
 method = "simplex",  
 columns = 1:4, # columns used to reconstruct the state space.  
 target\_column = 1,  
 stats\_only = F,  
 silent = T)  
  
## "Resource decreased" situation  
# Attractor is reconstructed using the original time series, but the dynamics is forecasted based on the resource-decreased situation  
scenario\_dec <- block\_lnlp(block\_dec,  
 lib = c(1,nrow(block0)),  
 pred = c((nrow(block0)+2),nrow(block\_dec)),  
 method = "simplex",  
 columns = 1:4, # columns used to reconstruct the state space.  
 target\_column = 1,  
 stats\_only = F,  
 silent = T)  
  
# Extract model outputs  
pred\_nochange <- scenario\_res[[1]]$model\_output  
pred\_increase <- scenario\_inc[[1]]$model\_output[2000:nrow(scenario\_inc[[1]]$model\_output),]  
pred\_decrease <- scenario\_dec[[1]]$model\_output[2000:nrow(scenario\_dec[[1]]$model\_output),]  
  
  
#### Visualize results  
# Extract time window (for figure)  
trange <- 1:50  
plot(pred\_nochange[trange,"obs"],  
 type="l", col="black", lwd=2, xlab="time",  
 ylab="Normalized C1 values", main="Scenario exploration")

# Add predicted values using noraml multivariate simplex projection  
points(pred\_nochange[trange,"pred"],bg="black",pch=21)  
# Add predicted values under R increased situation  
points(pred\_increase[trange,"pred"],bg="red",pch=24)  
# Add predicted values under R increased situation  
points(pred\_decrease[trange,"pred"],bg="blue",pch=25)  
# Add legend  
legend("topleft",c("predicted","R increased","R decreased"),  
 pt.bg=c(1,2,4), pch=c(21,24,25), bty="n", cex = 0.8)

As can be seen in Figure 8 in the main text, increase in *R* does not always result in increased *C1*. This is because the effect of *R* on *C1* depends additionally on the condition of other species, a phenomenon known as state-dependent behavior in dynamical systems.

References

Deyle ER, Fogarty M, Hsieh CH, Kaufman L, MacCall AD, Munch SB, Perretti CT, Ye H, Sugihara G (2013) Predicting climate effects on Pacific sardine. Proc Natl Acad Sci USA 110: 6430-6435. DOI: 10.1073/pnas.1215506110

Deyle ER, May RM, Munch SB, Sugihara G (2016) Tracking and forecasting ecosystem interactions in real time. Proc R Soc Lond B Biol Sci 283: 20152258.