ENC definition

AF

2024-07-22

##   
## Attaching package: 'dplyr'

## The following object is masked from 'package:MASS':  
##   
## select

## The following objects are masked from 'package:stats':  
##   
## filter, lag

## The following objects are masked from 'package:base':  
##   
## intersect, setdiff, setequal, union

##   
## Attaching package: 'tidyr'

## The following object is masked from 'package:reshape2':  
##   
## smiths

## Loading required package: viridisLite

### Motivation

Global change inflicts strong changes in the environment that species adapt to. Globally, species’ populations, distributions, and phenology is being highly altered. These changes can add, eliminate, or alter the strength and sign of interactions affecting community composition and structure, and therefore its functioning. Given the complexity of biodiversity and the challenge of its measurement, how can we anticipate changes of those properties?

Contemporary empirical investigations (ref) and theoretical advancements (ref) are highlighting the importance of evaluating the degree of shared responses to the environment across multiple species within ecological communities (commonly referred to as community synchrony; refs). This integrated approach underscores the interconnectedness of species dynamics and their collective impact on ecosystem stability (ref). Despite advancements in understanding the impacts of community synchrony, the incorporation of information on the interaction network structures remains lacking. Simultaneously, understanding how ecosystem functioning depends on network structure remains limited (Walther 2010, Montoya & Raffaelli 2010), as it necessitates information on species’ dynamics driven by more factors than interactions (Lavergne et al. 2010; Tylianakis & Morris 2017; Strydom et al. 2021; Purves et al. 2013; Harfoot et al. 2014). There is therefore a pressing need for theory and tools that comprehensively integrate the interconnected dynamics of species and their collective impact on ecosystem stability to anticipate changes in the structure and function of communities.

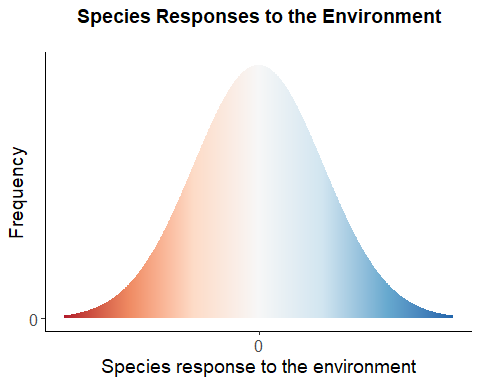
### Objective:

In this study, we introduce the concept of Ecological Network Coherence (ENC), which provides a conceptual framework for capturing the correlational structure of species’ interdependent responses to the environment in an ecological network. The concept is manifested in an empirical community-level pattern that results from the shared responses to environmental variables among interacting species. We propose that this empirical pattern could help predicting the potential for disruptions in ecosystem functioning and serve as a foundational element for developing a new indicator of the consequences of biodiversity changes at the community level. Our primary objectives encompass (1) defining ENC and its components, (2) exploring its quantitative application to ecosystem functioning, and (4) illustrating how to measure and explore it in empirical data.

### Concept definition

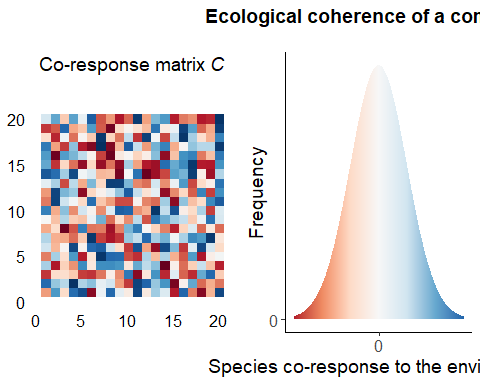
Species responses to environmental changes include alterations in population trends, spatial distribution shifts, and phenology changes (Bellard et al., 2012). Consequently, ecological communities are characterized by heterogeneous responses, which can be summarized as a distribution of species responses:

## Scale for x is already present.  
## Adding another scale for x, which will replace the existing scale.



The above example illustrates a hypothetical community where species responses are perfectly balanced, with most species would showing minimal response to environmental changes, while a minority exhibit strong positive or negative responses.

From this, we can assess the similarities and differences in species responses, or their co-responses. Species co-responses can be characterized based on their covariance: a positive covariance indicates that greater values of one variable typically correspond to greater values of the other, while a negative covariance suggests an inverse relationship. Correlation is a standardized version of covariance that adjusts the data to have a mean of 0 and a standard deviation of 1 (bounded between -1 and 1), providing a more straightforward interpretation of the relationship’s strength and direction. Species co-responses within a community can be summarized in a matrix,, which comprises pairwise correlations. The distribution of these co-responses forms a community pattern that we define as the Ecological Coherence of the community, reflecting how coherent species responses are to the environment.



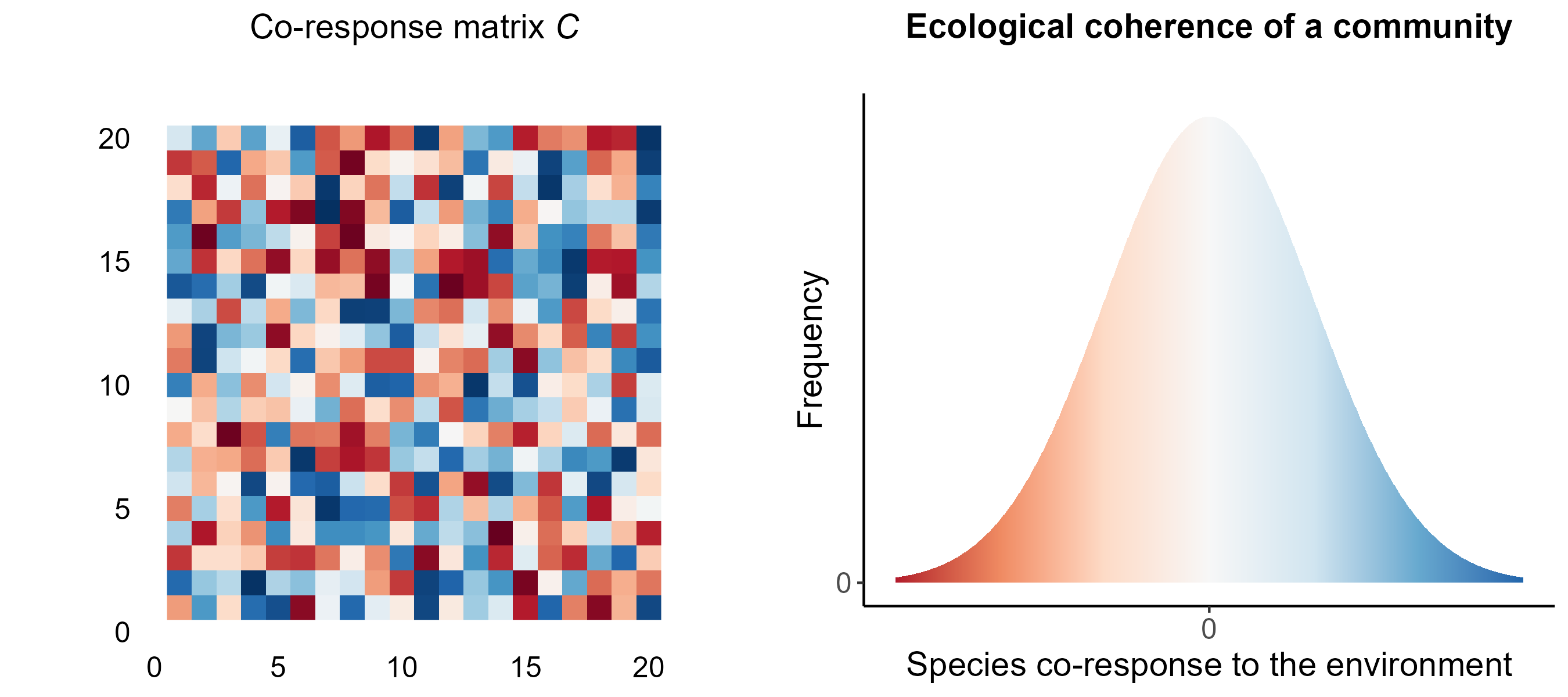


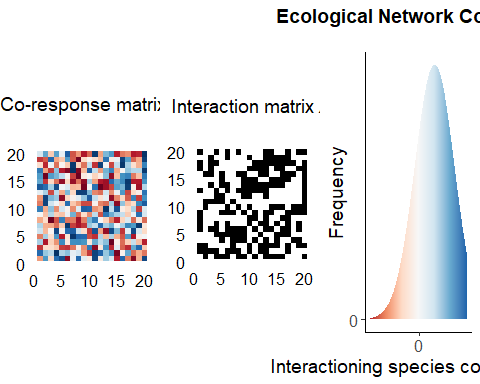
Fig. 2

We can incorporate information on community structure by considering the co-responses between pairs of interacting species: the matrix can be filtered out (multiplied) by the adjacency matrix of interactions , which denotes the presence or absence of interactions. The distribution of the resulting filtered co-response matrix is a pattern that shows the degree of co-responses to specific environmental variables between interacting species within a community at given location in space and time, which we define as Ecological Network Coherence.

# Add probabilities of black (presence) and white (absence) squares to the A matrix based on the values in C  
  
data\_matrix$color <- ifelse(data\_matrix$value > 0,   
 sample(c("black", "white"), size = nrow(data\_matrix), replace = TRUE, prob = c(0.7, 0.3)),  
 sample(c("black", "white"), size = nrow(data\_matrix), replace = TRUE, prob = c(0.3, 0.7)))  
  
# Plot   
plot\_matrix\_A <- ggplot(data\_matrix, aes(x = x, y = y, fill = color)) +  
 geom\_raster() +  
 scale\_fill\_manual(values = c("black" = "black", "white" = "white")) +  
 labs(x = NULL, y = NULL, title = expression("Interaction matrix"~italic("A"))) +  
 theme\_classic() +  
 theme(  
 legend.position = "none",  
 axis.text = element\_blank(),  
 axis.title = element\_blank(),  
 axis.ticks = element\_blank(), # Remove axis ticks  
 axis.line = element\_blank(), # Remove axis lines  
 plot.title = element\_text(hjust = 0.5)  
 ) +  
 coord\_fixed()+  
 my\_theme  
  
  
  
# Define parameters for the new distribution  
set.seed(42)   
mean\_new <- 1 # Shift the distribution to the right  
sd\_new <- 1.2 # Change the standard deviation  
  
# Generate new data  
data\_new <- data.frame(  
 response = seq(-3, 3, length.out = 1000),  
 frequency = dnorm(seq(-3, 3, length.out = 1000), mean = mean\_new, sd = sd\_new)  
)  
  
  
plot\_distrib\_resp <- ggplot(data\_new, aes(x = response, y = frequency, fill = response)) +  
 geom\_tile(aes(y = frequency / 2, height = frequency, width = 0.05)) + # Adjust width to better visualize the shape  
 scale\_fill\_distiller(palette = "RdBu", limits = c(-3, 3), direction = 1) +  
 labs(x = "Interactioning species co-response", y = "Frequency", title = "Ecological Network Coherence") +  
 theme\_classic() +  
 xlim(-3, 3) +  
 scale\_x\_continuous(breaks = 0) +  
 scale\_y\_continuous(breaks = c(0, 1)) +  
 theme(legend.position = "none",  
 plot.title = element\_text(hjust = 0.5))+  
 my\_theme

## Scale for x is already present.  
## Adding another scale for x, which will replace the existing scale.

plot\_ENC <- ggarrange(  
 plot\_matrix\_c,  
 plot\_matrix\_A,  
 plot\_distrib\_resp,  
   
 ncol = 3,   
 nrow = 1  
)  
  
plot\_ENC



#ggsave("fig3.png", height = 4, width = 11)

By summarizing the pattern of species co-responses and interactions in a community distribution, ENC brings the possibility to analyze the predictive consequences of different statistical modes on functioning. Summarizing the complex interdependence of species responses in a distribution also brings the advantage to bridge empirical and theoretical work: ENC is built from commonly measured data and used by empirisist and is manifested in a clear statistical object that can be explored.

# Linking ENC to ecosystem functioning

Let be the abundance of species in a Lotka-Volterra system, with intrinsic growth rate and interaction coefficient with species given by . We can write the dynamics of species as:

Assuming the existence of an equilibrium and that all species have non-zero abundances at equilibrium, we can write the equation for all species at equilibrium using vector notations (i.e. with arrows):

which, if the system is sufficiently not pathological, can be inverted:

If a push perturbation is applied to the system over a sufficiently long time, we thus expect the equilibrium to shift away from by a quantity , given by:

Now assuming that is random among species, i.e. that the effect of the perturbation on each and every species is drawn from an underlying distribution, we can call the mean push perturbation and the variance of push perturbation among species. For commodity, we note the vector comprising as many 1’s as the number of species in the system, and is used as the symbol for transposition.

In the same way, we can define the mean abundance change due to the perturbation and the variance of these abundance changes among species.

Hereafter, we use the notation in order to look at the effect of on changes in abundances at equilibrium. Getting back to equation and developing the sums, we get:

In this final expression, we recognize the average (*sensu* “among all species”) of the product of the and the column sums of . Hence, if we further decompose this using the abusive and notations (expectations are to be understood as averages over columns):

Adopting the perspective where the vector is a random vector that follows a multivariate distribution with mean vector and covariances between vector components, we can use the propagation of uncertainty to deduce that:

with (as above) the inverse of the LV interaction matrix.

### Predictions - effect of ENC on the predictability of community dynamics

We demonstrated mathematically how ENC links to changes in abundances of species in a community (equation ) and the predictability of this changes (equation ). We then explore whether different

As a first approach to investigating the consequences of ENC on ecosystem functioning, we demonstrate mathematically how it links to changes in abundance and its predictability (variance in abundance changes) in a Lotka Volterra system submitted to a push perturbation.

Equation (11) captures how the covariances of perturbation responses among all species in the network contribute to the variance in abundance changes for a focal species . The equation sums up the contributions of all pairs of species ( and ) to the variance of the abundance change of species . This means that the variance in abundance changes for a focal species is influenced by how all other species in the network respond to perturbations (as captured by the covariances and how these responses propagate through the network (as captured by the elements of the matrix ).

Upon a perturbation, we get for each species.

Therefore, the covariance matrix is:

### Breaking down the scenarios

The net effect on variance is the result of complex interactions between species’ responses to perturbations and their interaction coefficients, which can either amplify or reduce the variance depending on the specific configuration.

#### 1. and have the same sign

* is positive.
* If **Positive covariance** ( > 0), the term will be positive.
* If **Negative covariance** ( < 0), the term will be negative.

Positive covariance increases the overall variance , potentially making the system less predictable, whereas negative covariance has the contrary effect.

#### 2. and have opposite signs

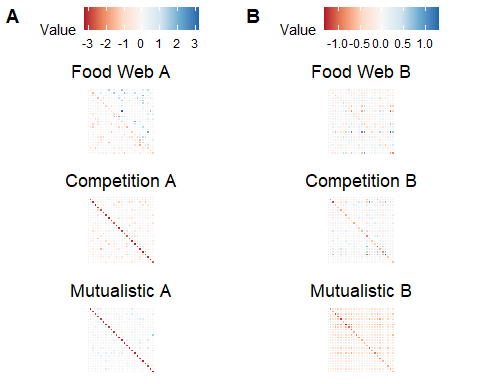
* is negative.
* If **Positive covariance** ( > 0), the term will be negative.
* If **Negative covariance** ( < 0), the term will be positive.

Positive covariance decreases the overall variance , potentially making the system more predictable, whereas negative covariance has the contrary effect.

#### 3. Heterogeneity of interaction effects

The scenarios discussed apply to communities where most interactions between species are either predominantly positive or negative. However, in empirical ecological communities, the net effects of interactions are often heterogeneous, with interaction effects (positive or negative) varying widely.

# Function to simulate quantitative networks  
sim\_quantitative\_network <- function(Net\_type, S, C, aij\_params, rho = 0, diag\_type) {  
 A <- matrix(0, S, S)  
 n\_pairs <- S \* (S - 1) / 2  
 B <- runif(n\_pairs) <= C  
 if (Net\_type == "random") {  
 A[upper.tri(A)] <- B \* rnorm(n\_pairs, aij\_params[1], aij\_params[2])  
 A <- t(A)  
 A[upper.tri(A)] <- B \* rnorm(n\_pairs, aij\_params[1], aij\_params[2])  
 } else if (Net\_type == "predator-prey") {  
 aij <- -abs(rnorm(n\_pairs, aij\_params[1], aij\_params[2]))  
 A[upper.tri(A)] <- B \* aij  
 A <- t(A)  
 aij <- abs(rnorm(n\_pairs, aij\_params[1], aij\_params[2]))  
 A[upper.tri(A)] <- B \* aij  
 } else if (Net\_type == "competition") {  
 aij <- -abs(rnorm(n\_pairs \* 2, aij\_params[1], aij\_params[2]))  
 A[upper.tri(A)] <- B \* aij[1:n\_pairs]  
 A <- t(A)  
 A[upper.tri(A)] <- B \* aij[(n\_pairs + 1):length(aij)]  
 } else if (Net\_type == "mutualistic") {  
 aij <- abs(rnorm(n\_pairs \* 2, aij\_params[1], aij\_params[2]))  
 A[upper.tri(A)] <- B \* aij[1:n\_pairs]  
 A <- t(A)  
 A[upper.tri(A)] <- B \* aij[(n\_pairs + 1):length(aij)]  
 } else {  
 stop("Incorrect network type")  
 }  
   
 if(diag\_type == "zero"){  
 diag(A) <- 0  
 } else {  
 diag(A) <- -(max(Re(eigen(A)$values)) + runif(S, 0.1))  
 while (max(Re(eigen(A)$values)) > 0) {  
 diag(A) <- -(max(Re(eigen(A)$values)) + runif(S, 0.1))  
 }  
 }  
   
 return(A)  
}  
  
  
# Parameters  
S <- 25 # Number of species  
C <- 0.3 # Connectivity  
aij\_params <- c(0, 1) # Mean and SD for interaction strengths  
  
# Generate matrices  
A\_foodweb <- sim\_quantitative\_network("predator-prey", S, C, aij\_params, diag\_type = "nonzero")  
A\_competition <- sim\_quantitative\_network("competition", S, C, aij\_params, diag\_type = "nonzero")  
A\_mutualistic <- sim\_quantitative\_network("mutualistic", S, C, aij\_params, diag\_type = "nonzero")  
  
# Invert matrices  
B\_foodweb <- solve(A\_foodweb)  
B\_competition <- solve(A\_competition)  
B\_mutualistic <- solve(A\_mutualistic)  
  
# Function to plot matrices  
plot\_matrix <- function(mat, title) {  
 max\_val <- max(abs(mat), na.rm = TRUE)  
 mat\_melt <- as.data.frame(as.table(mat))  
 mat\_melt$Var1 <- factor(mat\_melt$Var1, levels = rev(levels(mat\_melt$Var1))) # Reverse the y-axis order  
 ggplot(mat\_melt, aes(Var2, Var1, fill = Freq)) +  
 geom\_tile(color = "white") +  
 scale\_fill\_distiller(palette = "RdBu", direction = 1, na.value = "white", limits = c(-max\_val, max\_val)) +  
 labs(x = NULL, y = NULL, fill = "Value", title = title) +  
 theme\_minimal() +  
 theme(axis.text = element\_blank(), axis.ticks = element\_blank(), plot.title = element\_text(hjust = 0.5)) +  
 coord\_fixed(ratio = 1) # Ensure the matrix is square  
}  
  
# Create plots  
p1 <- plot\_matrix(A\_foodweb, "Food Web A")  
p2 <- plot\_matrix(B\_foodweb, "Food Web B")  
p3 <- plot\_matrix(A\_competition, "Competition A")  
p4 <- plot\_matrix(B\_competition, "Competition B")  
p5 <- plot\_matrix(A\_mutualistic, "Mutualistic A")  
p6 <- plot\_matrix(B\_mutualistic, "Mutualistic B")  
  
pA <- ggarrange(p1,  
 p3,  
 p5,  
 ncol = 1,  
 nrow = 3,  
 common.legend = TRUE)  
  
pB <- ggarrange(p2,  
 p4,  
 p6,  
 ncol = 1,  
 nrow = 3,  
 common.legend = TRUE)  
  
# Arrange plots in a grid  
ggarrange(pA, pB, ncol = 2, nrow = 1,  
 labels = LETTERS[1:2])



#ggsave("fig4.1.png", height = 8, width = 5)

This heterogeneity in interaction coefficients\* means that any covariance structure will result in both increases and decreases in abundance variance. The strength of coherence will modulate the magnitude of these changes in variance: small covariances are likely to decrease changes in variance, while large covariances will increase them. This leads to two general predictions: first, the heterogeneous nature of real-world interactions makes the overall impact of coherence on stability complex and challenging to predict. Second, ENC patterns with strong coherence will reduce the predictability of communities and increase the potential for significant disruptions in their structure and function.

\*Seems that this is not the case for mutualistic networks, whose B matrix does not contain positive values:

# Function to test if a matrix contains any negative values  
contains\_positive\_values <- function(mat) {  
 any(mat > 0)  
}  
  
# Simulate 100 mutualistic networks and test for negative values  
  
positive\_values\_found <- numeric(length = 1000)  
  
for (i in 1:1000) {  
 A <- sim\_quantitative\_network("mutualistic", S, C, aij\_params, diag\_type = "nonzero")  
 B <- solve(A)  
 positive\_values\_found[i] <- contains\_positive\_values(B)  
}  
  
# Check the results  
sum(positive\_values\_found) # Number of B matrices containing negative values

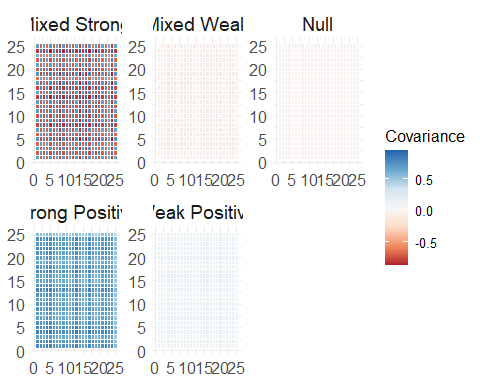
## [1] 0

## Simulations to test predictions

We simulate 100 food web structures with random interaction strengths and compute the abundance change and variance in abundance change according to formulas and . We test different covariance structures: strong negative, weak negative, null (zero covariance), weak positive, strong positive, weak mixed (positive and negative), and strong mixed. Based on our mathematical intuition, we expect to find that the variance in abundance change is higher for the scenarios with strong positive, strong negative, and strong mixed covariances, and lower for weak positive, weak negative, and null covariances.

Covariance matrices:

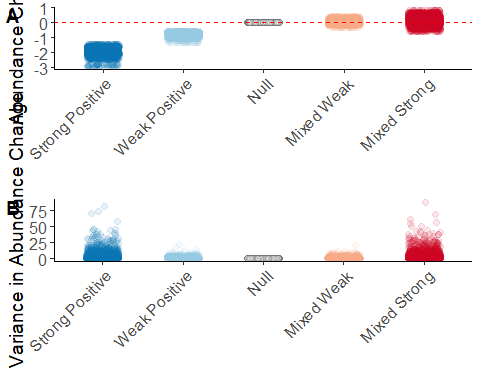
# Number of species  
S <- 25  
  
# Simulate initial r values (ri)  
set.seed(123)  
ri <- runif(S, -1, 1)  
  
# Function to generate rf values with positive covariance  
generate\_rf\_positive <- function(ri, mean\_change, sd\_variation) {  
 rf <- ri + rnorm(length(ri), mean\_change, sd\_variation)  
 return(rf)  
}  
  
# Mixed (positive and negative)  
generate\_rf\_mixed <- function(ri, mean\_change\_positive, mean\_change\_negative, sd\_variation) {  
 rf <- ri  
 for (i in 1:length(ri)) {  
 if (i %% 2 == 0) {  
 rf[i] <- ri[i] + rnorm(1, mean\_change\_positive, sd\_variation)  
 } else {  
 rf[i] <- ri[i] + rnorm(1, mean\_change\_negative, sd\_variation)  
 }  
 }  
 return(rf)  
}  
  
# Compute the covariance matrix using delta\_r vectors  
compute\_cov\_matrix <- function(delta\_r\_vectors) {  
 S <- nrow(delta\_r\_vectors)  
 cov\_matrix <- matrix(NA, nrow = S, ncol = S)  
 for (i in 1:S) {  
 for (j in 1:S) {  
 cov\_matrix[i, j] <- cov(delta\_r\_vectors[i, ], delta\_r\_vectors[j, ])  
 }  
 }  
 return(cov\_matrix)  
}  
  
# Generate rf values for different scenarios  
mean\_change\_strong <- 1.2  
mean\_change\_weak <- 0.5  
sd\_variation <- 0.1  
  
# Strong positive  
rf\_strong\_positive <- generate\_rf\_positive(ri, mean\_change\_strong, sd\_variation)  
delta\_r\_strong\_positive <- rf\_strong\_positive - ri  
  
# Weak positive  
rf\_weak\_positive <- generate\_rf\_positive(ri, mean\_change\_weak, sd\_variation)  
delta\_r\_weak\_positive <- rf\_weak\_positive - ri  
  
# Mixed weak  
mean\_change\_positive <- 0.5  
mean\_change\_negative <- -0.5  
rf\_mixed\_weak <- generate\_rf\_mixed(ri, mean\_change\_positive, mean\_change\_negative, sd\_variation)  
delta\_r\_mixed\_weak <- rf\_mixed\_weak - ri  
  
# Mixed strong  
mean\_change\_positive\_strong <- 1.2  
mean\_change\_negative\_strong <- -1.2  
rf\_mixed\_strong <- generate\_rf\_mixed(ri, mean\_change\_positive\_strong, mean\_change\_negative\_strong, sd\_variation)  
delta\_r\_mixed\_strong <- rf\_mixed\_strong - ri  
  
# Null (no change)  
delta\_r\_null <- rep(0, S)  
delta\_r\_vectors\_null <- cbind(ri, ri) # No change, so rf = ri  
  
# Create vectors delta\_r for each species (each row corresponds to a species)  
delta\_r\_vectors\_strong\_positive <- cbind(ri, rf\_strong\_positive)  
delta\_r\_vectors\_weak\_positive <- cbind(ri, rf\_weak\_positive)  
delta\_r\_vectors\_mixed\_weak <- cbind(ri, rf\_mixed\_weak)  
delta\_r\_vectors\_mixed\_strong <- cbind(ri, rf\_mixed\_strong)  
  
# Compute C matrices  
cov\_matrix\_strong\_positive <- compute\_cov\_matrix(delta\_r\_vectors\_strong\_positive)  
cov\_matrix\_weak\_positive <- compute\_cov\_matrix(delta\_r\_vectors\_weak\_positive)  
cov\_matrix\_mixed\_weak <- compute\_cov\_matrix(delta\_r\_vectors\_mixed\_weak)  
cov\_matrix\_mixed\_strong <- compute\_cov\_matrix(delta\_r\_vectors\_mixed\_strong)  
cov\_matrix\_null <- matrix(0, nrow = S, ncol = S) # Null scenario has no change  
  
# Combine all covariance matrices into a list  
cov\_matrices <- list(  
 "Strong Positive" = cov\_matrix\_strong\_positive,  
 "Weak Positive" = cov\_matrix\_weak\_positive,  
 "Mixed Weak" = cov\_matrix\_mixed\_weak,  
 "Mixed Strong" = cov\_matrix\_mixed\_strong,  
 "Null" = cov\_matrix\_null  
)  
  
# Prepare the plot data for the example C matrices  
prepare\_plot\_data <- function(cov\_matrices) {  
 plot\_data <- lapply(names(cov\_matrices), function(name) {  
 matrix <- cov\_matrices[[name]]  
 df <- melt(matrix)  
 df$Scenario <- name  
 return(df)  
 })  
 combined\_plot\_data <- do.call(rbind, plot\_data)  
 return(combined\_plot\_data)  
}  
  
# Generate the plot data  
plot\_data <- prepare\_plot\_data(cov\_matrices)  
  
# Generate the plot  
p <- ggplot(plot\_data, aes(x = Var1, y = Var2, fill = value)) +  
 geom\_tile(color = "white") +  
 scale\_fill\_distiller(palette = "RdBu", direction = 1, na.value = "white") +  
 labs(  
 x = NULL,  
 y = NULL,  
 fill = "Covariance"  
 ) +  
 theme\_minimal() +  
 theme(strip.text = element\_text(size = 14)) +  
 facet\_wrap(~ Scenario, ncol = 3, scales = "free") +  
 my\_theme  
  
# Display the plot  
print(p)



#ggsave("fig5.png", height = 7, width = 9)

Simulations (similar results for mutualistic, competition, and predator-prey interactions):

# Number of species  
S <- 25  
  
# Simulate initial r values (ri)  
set.seed(123)  
ri <- runif(S, -1, 1)  
  
# Function to generate rf values with positive covariance  
generate\_rf\_positive <- function(ri, mean\_change, sd\_variation) {  
 rf <- ri + rnorm(length(ri), mean\_change, sd\_variation)  
 return(rf)  
}  
  
# Function to generate rf values with Mixed covariance (positive and negative)  
generate\_rf\_mixed <- function(ri, mean\_change\_positive, mean\_change\_negative, sd\_variation) {  
 rf <- ri  
 for (i in 1:length(ri)) {  
 if (i %% 2 == 0) {  
 rf[i] <- ri[i] + rnorm(1, mean\_change\_positive, sd\_variation)  
 } else {  
 rf[i] <- ri[i] + rnorm(1, mean\_change\_negative, sd\_variation)  
 }  
 }  
 return(rf)  
}  
  
# Compute the covariance matrix using delta\_r vectors  
compute\_cov\_matrix <- function(delta\_r\_vectors) {  
 S <- nrow(delta\_r\_vectors)  
 cov\_matrix <- matrix(NA, nrow = S, ncol = S)  
 for (i in 1:S) {  
 for (j in 1:S) {  
 cov\_matrix[i, j] <- cov(delta\_r\_vectors[i, ], delta\_r\_vectors[j, ])  
 }  
 }  
 return(cov\_matrix)  
}  
  
# Simulating scenarios  
mean\_change\_strong <- 1.2  
mean\_change\_weak <- 0.5  
sd\_variation <- 0.1  
  
# Function to simulate quantitative networks  
sim\_quantitative\_network <- function(Net\_type, S, C, aij\_params, rho = 0, diag\_type) {  
 A <- matrix(0, S, S)  
 n\_pairs <- S \* (S - 1) / 2  
 B <- runif(n\_pairs) <= C  
 if (Net\_type == "random") {  
 A[upper.tri(A)] <- B \* rnorm(n\_pairs, aij\_params[1], aij\_params[2])  
 A <- t(A)  
 A[upper.tri(A)] <- B \* rnorm(n\_pairs, aij\_params[1], aij\_params[2])  
 } else if (Net\_type == "predator-prey") {  
 aij <- -abs(rnorm(n\_pairs, aij\_params[1], aij\_params[2]))  
 A[upper.tri(A)] <- B \* aij  
 A <- t(A)  
 aij <- abs(rnorm(n\_pairs, aij\_params[1], aij\_params[2]))  
 A[upper.tri(A)] <- B \* aij  
 } else if (Net\_type == "competition") {  
 aij <- -abs(rnorm(n\_pairs \* 2, aij\_params[1], aij\_params[2]))  
 A[upper.tri(A)] <- B \* aij[1:n\_pairs]  
 A <- t(A)  
 A[upper.tri(A)] <- B \* aij[(n\_pairs + 1):length(aij)]  
 } else if (Net\_type == "mutualistic") {  
 aij <- abs(rnorm(n\_pairs \* 2, aij\_params[1], aij\_params[2]))  
 A[upper.tri(A)] <- B \* aij[1:n\_pairs]  
 A <- t(A)  
 A[upper.tri(A)] <- B \* aij[(n\_pairs + 1):length(aij)]  
 } else {  
 stop("Incorrect network type")  
 }  
   
 if(diag\_type == "zero"){  
 diag(A) <- 0  
 } else {  
 diag(A) <- -(max(Re(eigen(A)$values)) + runif(S, 0.1))  
 while (max(Re(eigen(A)$values)) > 0) {  
 diag(A) <- -(max(Re(eigen(A)$values)) + runif(S, 0.1))  
 }  
 }  
   
 return(A)  
}  
  
# Function to calculate metrics  
calculate\_metrics <- function(B, cov\_delta\_r, delta\_r) {  
 n <- nrow(B)  
   
 # Variance of Delta X  
 var\_delta\_X <- numeric(n)  
 for (i in 1:n) {  
 var\_sum <- 0  
 for (k in 1:n) {  
 for (l in 1:n) {  
 var\_sum <- var\_sum + B[i, k] \* B[i, l] \* cov\_delta\_r[k, l]  
 }  
 }  
 var\_delta\_X[i] <- var\_sum  
 }  
   
 # Compute column sums of B  
 col\_sums\_B <- colSums(B)  
   
 # Average of the sum of each column of B  
 mean\_sum\_B <- mean(col\_sums\_B)  
   
 # Mean of delta\_r (assuming it is an average of a random distribution)  
 mean\_delta\_r <- mean(delta\_r)  
   
 # Covariance between sum of each column of B and delta\_r  
 cov\_sum\_B\_delta\_r <- cov(col\_sums\_B, delta\_r)  
   
 # Calculate Abundance Change  
 abundance\_change <- mean\_sum\_B \* mean\_delta\_r + cov\_sum\_B\_delta\_r  
   
 return(list(Variance = var\_delta\_X, Abundance\_Change = abundance\_change))  
}  
  
# Run simulations  
run\_simulations\_with\_metrics <- function(num\_simulations, Net\_type, S, C, aij\_params, scenario, mean\_change\_strong, mean\_change\_weak, sd\_variation) {  
 results <- replicate(num\_simulations, {  
 # Generate new C matrix for each iteration  
 if (scenario == "Strong Positive") {  
 rf <- generate\_rf\_positive(ri, mean\_change\_strong, sd\_variation)  
 } else if (scenario == "Weak Positive") {  
 rf <- generate\_rf\_positive(ri, mean\_change\_weak, sd\_variation)  
 } else if (scenario == "Null") {  
 rf <- ri  
 } else if (scenario == "Mixed Weak") {  
 rf <- generate\_rf\_mixed(ri, mean\_change\_weak, -mean\_change\_weak, sd\_variation)  
 } else if (scenario == "Mixed Strong") {  
 rf <- generate\_rf\_mixed(ri, mean\_change\_strong, -mean\_change\_strong, sd\_variation)  
 }  
   
 delta\_r <- rf - ri  
 delta\_r\_vectors <- cbind(ri, rf)  
 cov\_matrix <- compute\_cov\_matrix(delta\_r\_vectors) # compute C matrix from delta\_r vectors  
   
 interaction\_matrix <- sim\_quantitative\_network(Net\_type, S, C, aij\_params, diag\_type = "nonzero")  
 inverse\_matrix <- solve(interaction\_matrix)  
 if (is.null(inverse\_matrix)) {  
 return(list(Variance = rep(NA, S), Abundance\_Change = NA))  
 }  
   
 metrics <- calculate\_metrics(inverse\_matrix, cov\_matrix, delta\_r)  
 list(metrics = metrics, A\_matrix = interaction\_matrix, B\_matrix = inverse\_matrix, C\_matrix = cov\_matrix)  
 }, simplify = FALSE)  
   
 list(results = lapply(results, function(res) res$metrics),   
 A\_matrices = lapply(results, function(res) res$A\_matrix),   
 B\_matrices = lapply(results, function(res) res$B\_matrix),   
 C\_matrices = lapply(results, function(res) res$C\_matrix))  
}  
  
# Generate results for each scenario  
Net\_type <- "predator-prey"  
C <- 0.2  
aij\_params <- c(0, 0.1)  
num\_simulations <- 100  
  
scenarios <- c("Strong Positive", "Weak Positive", "Null", "Mixed Weak", "Mixed Strong")  
  
simulation\_results\_with\_metrics <- lapply(scenarios, function(scenario) {  
 run\_simulations\_with\_metrics(num\_simulations, Net\_type, S, C, aij\_params, scenario, mean\_change\_strong, mean\_change\_weak, sd\_variation)  
})  
  
  
  
# Prepare the matrices data  
prepare\_matrices\_data <- function(simulation\_results) {  
 df\_list\_matrices <- lapply(names(simulation\_results), function(scenario) {  
 A\_matrices <- simulation\_results[[scenario]]$A\_matrices  
 B\_matrices <- simulation\_results[[scenario]]$B\_matrices  
 C\_matrices <- simulation\_results[[scenario]]$C\_matrices  
 df <- data.frame(  
 Scenario = rep(scenario, length(A\_matrices)),  
 A\_matrix = I(A\_matrices),  
 B\_matrix = I(B\_matrices),  
 C\_matrix = I(C\_matrices)  
 )  
 return(df)  
 })  
 combined\_df\_matrices <- bind\_rows(df\_list\_matrices)  
 return(combined\_df\_matrices)  
}  
  
# Prepare the matrices data  
combined\_matrices\_data <- prepare\_matrices\_data(setNames(simulation\_results\_with\_metrics, scenarios))  
  
  
  
###################################################  
  
# Prepare the data for plotting  
prepare\_combined\_plot\_data <- function(simulation\_results) {  
 df\_list\_var <- lapply(names(simulation\_results), function(scenario) {  
 data <- simulation\_results[[scenario]]$results  
 df <- data.frame(  
 Scenario = scenario,  
 Variance = unlist(lapply(data, function(res) res$Variance)),  
 Abundance\_Change = unlist(lapply(data, function(res) res$Abundance\_Change))  
 )  
 return(df)  
 })  
 combined\_df\_var <- bind\_rows(df\_list\_var)  
 combined\_df\_var <- combined\_df\_var %>% filter(!is.na(Variance) & !is.na(Abundance\_Change))  
   
 return(combined\_df\_var)  
}  
  
# Prepare the data  
combined\_plot\_data <- prepare\_combined\_plot\_data(setNames(simulation\_results\_with\_metrics, scenarios))  
  
# Plot results  
scenario\_order <- c(  
 "Strong Positive",  
 "Weak Positive",  
 "Null",  
 "Mixed Weak",  
 "Mixed Strong"  
)  
  
combined\_plot\_data$Scenario <- factor(combined\_plot\_data$Scenario, levels = scenario\_order)  
  
# Define color palette based on RdBu  
color\_palette <- scales::brewer\_pal(palette = "RdBu", direction = -1)(5)  
mixed\_weak\_color <- "violet"  
mixed\_strong\_color <- "magenta4"  
  
  
# Define scenario colors  
scenario\_colors <- setNames(c(color\_palette, mixed\_weak\_color, mixed\_strong\_color), scenario\_order)  
  
# Plot abundance change  
p1 <- ggplot(combined\_plot\_data, aes(x = Scenario, y = Abundance\_Change, fill = Scenario)) +  
 geom\_jitter(width = 0.2, alpha = 0.1, shape = 21, size = 2, aes(color = Scenario)) +  
 scale\_fill\_manual(values = scenario\_colors) +  
 scale\_color\_manual(values = c(setNames(ifelse(scenario\_order == "Null", "black", scenario\_colors), scenario\_order), "black")) +  
 geom\_hline(yintercept = 0, linetype = "dashed", color = "red") +  
 labs(x = " ", y = "Abundance Change") +  
 theme\_classic() +  
 theme(axis.text.x = element\_text(angle = 45, hjust = 1), legend.position = "none") +  
 my\_theme  
  
# Plot variance in abundance change  
p2 <- ggplot(combined\_plot\_data, aes(x = Scenario, y = Variance, fill = Scenario)) +  
 geom\_jitter(width = 0.2, alpha = 0.1, shape = 21, size = 2, aes(color = Scenario)) +  
 scale\_fill\_manual(values = scenario\_colors) +  
 scale\_color\_manual(values = c(setNames(ifelse(scenario\_order == "Null", "black", scenario\_colors), scenario\_order), "black")) +  
 labs(x = " ", y = "Variance in Abundance Change") +  
 theme\_classic() +  
 theme(axis.text.x = element\_text(angle = 45, hjust = 1), legend.position = "none") +  
 my\_theme  
  
# Display plots  
ggarrange(p1, p2, nrow = 2, ncol = 1, labels = LETTERS[1:2])

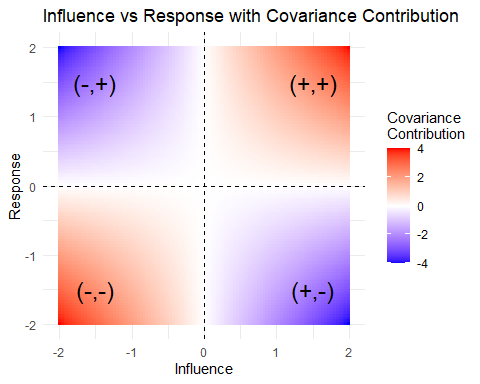


#ggsave("fig6.png", height = 10, width = 6)

The simulations confirm the expectation that the predictability of a community of interacting species increases when coherence is low (i.e. as values around 0 increase in the ENC pattern), whereas it decreases as coherence increases (i.e. as values approach the extremes -1 and 1 in the ENC pattern).

The covariance contribution in Formula 10, , suggests that species with higher centrality or influence within the interaction network (as captured by the matrix) and their corresponding responses () significantly impact the overall change in abundance. If species that are central in the network exhibit major responses to perturbations, this will likely lead to substantial changes in community abundance.

# Define a grid for influence and response  
influence <- seq(-2, 2, length.out = 100)  
response <- seq(-2, 2, length.out = 100)  
  
# Create a dataframe with all combinations of influence and response  
grid <- expand.grid(influence = influence, response = response)  
  
# Compute the covariance contribution for each combination  
grid$covariance <- grid$influence \* grid$response  
  
# Create the plot  
ggplot(grid, aes(x = influence, y = response, fill = covariance)) +  
 geom\_tile() +  
 scale\_fill\_gradient2(low = "blue", high = "red", mid = "white", midpoint = 0,   
 limit = c(-4, 4), space = "Lab",   
 name = "Covariance\nContribution") +  
 geom\_vline(xintercept = 0, linetype = "dashed", color = "black") +  
 geom\_hline(yintercept = 0, linetype = "dashed", color = "black") +  
 annotate("text", x = 1.5, y = 1.5, label = "(+,+)", size = 6, color = "black") +  
 annotate("text", x = -1.5, y = 1.5, label = "(-,+)", size = 6, color = "black") +  
 annotate("text", x = -1.5, y = -1.5, label = "(-,-)", size = 6, color = "black") +  
 annotate("text", x = 1.5, y = -1.5, label = "(+,-)", size = 6, color = "black") +  
 labs(x = "Influence", y = "Response", title = "Influence vs Response with Covariance Contribution") +  
 theme\_minimal()



# Parameters  
S <- 20 # Number of species  
C <- 0.3 # Connectivity  
aij\_params <- c(0, 1) # Mean and SD for interaction strengths  
net\_type <- "random" # Type of network  
  
# Simulate the interaction matrix A  
A <- sim\_quantitative\_network(net\_type, S, C, aij\_params, diag\_type = "nonzero")  
  
# Compute the inverse matrix B  
B <- solve(A)  
  
# Generate a random response vector Delta\_r  
Delta\_r <- rnorm(S, 0, 1)  
  
# Compute the influence and response for plotting  
influence <- rowSums(B) # Sum of rows in B  
response <- Delta\_r # Response vector  
  
# Scale the influence and response to fall within [-2, 2]  
scale\_to\_range <- function(x, new\_min = -2, new\_max = 2) {  
 old\_min <- min(x)  
 old\_max <- max(x)  
 (x - old\_min) / (old\_max - old\_min) \* (new\_max - new\_min) + new\_min  
}  
  
influence <- scale\_to\_range(influence)  
response <- scale\_to\_range(response)  
  
# Generate a grid for the underlying color field  
x <- seq(-2, 2, length.out = 100)  
y <- seq(-2, 2, length.out = 100)  
grid <- expand.grid(influence = x, response = y)  
grid$covariance <- grid$influence \* grid$response # Covariance contribution  
  
# Convert real network data to a data frame  
real\_data <- data.frame(influence = influence, response = response)  
  
# Create the plot  
ggplot(grid, aes(x = influence, y = response, fill = covariance)) +  
 geom\_tile() +  
 scale\_fill\_gradient2(low = "blue", high = "red", midpoint = 0,   
 limit = c(-4, 4), space = "Lab",   
 name = expression(paste("Covariance\nContribution\nto", bar(Delta), "X"))) +  
 geom\_vline(xintercept = 0, linetype = "dashed", color = "black") +  
 geom\_hline(yintercept = 0, linetype = "dashed", color = "black") +  
 annotate("text", x = 1.5, y = 1.5, label = "(+,+)", size = 6, color = "black") +  
 annotate("text", x = -1.5, y = 1.5, label = "(-,+)", size = 6, color = "black") +  
 annotate("text", x = -1.5, y = -1.5, label = "(-,-)", size = 6, color = "black") +  
 geom\_point(data = real\_data, aes(x = influence, y = response), color = "black", size = 2, inherit.aes = FALSE) +  
 annotate("text", x = 1.5, y = -1.5, label = "(+,-)", size = 6, color = "black") +  
 labs(x = "Influence", y = "Response", title = "Influence vs Response with Covariance Contribution") +  
 theme\_minimal() +  
 theme(plot.title = element\_text(hjust = 0.5))

