

# Temporal dynamics of seed-dispersal networks: Disentangling the role of direct and indirect biotic and climatic processes

Andrés F. Ramírez-Mejía, Corey E. Tarwater, J. Patrick Kelley, Jinelle H. Sperry,  
Jeffrey T. Foster, Donald R. Drake, and Jeferson Vizentin-Bugoni

Appendix S5. Journal: Ecology

## Table of contents

<b>S1 Introduction</b>	<b>4</b>
<b>S2 Loading the data</b>	<b>4</b>
<b>S3 Models</b>	<b>6</b>
S3.1 Mathematic notation . . . . .	6
S3.1.0.1 Nestedness (likelihood and priors) . . . . .	6
S3.1.0.2 $H'_2$ (likelihood and priors) . . . . .	7
S3.1.0.3 Modularity (likelihood and priors) . . . . .	8
S3.1.1 Overall structure . . . . .	8
S3.2 Nestedness . . . . .	10
S3.2.1 Stan code . . . . .	10
S3.2.2 Fitting the model . . . . .	21
S3.2.3 Sampling diagnostics . . . . .	21
S3.2.4 Extracting posterior draws . . . . .	26
S3.2.5 Plotting posterior distribution . . . . .	29
S3.3 Modularity . . . . .	30
S3.3.1 Stan code . . . . .	30
S3.3.2 Fitting the model . . . . .	41
S3.3.3 Sampling diagnostics . . . . .	41
S3.3.4 Extracting posterior draws . . . . .	46
S3.3.5 Plotting posterior distribution . . . . .	78
S3.4 $H'_2$ . . . . .	78
S3.4.1 Stan code . . . . .	78

S3.4.2 Fitting the model . . . . .	90
S3.4.3 Sampling diagnostics . . . . .	90
S3.4.4 Extracting posterior draws . . . . .	94
S3.4.5 Plotting posterior distribution . . . . .	97
<b>S4 Final results</b>	<b>98</b>
S4.1 Climatic predictors . . . . .	98
S4.2 Phenology . . . . .	100
S4.2.1 Statistical differences . . . . .	102
S4.2.1.1 Species richness . . . . .	102
S4.2.1.2 Fruit production . . . . .	103
S4.3 Interaction functional space . . . . .	105
S4.3.1 Statistical differences . . . . .	109
S4.3.1.1 Fruits' nutritional functional space . . . . .	109
S4.3.1.2 Fruits' morphological functional space . . . . .	110
S4.3.1.3 Fruits' functional space (total) . . . . .	111
S4.3.1.4 Birds' functional space . . . . .	112
S4.4 Network dynamics . . . . .	113
S4.4.1 Statistical differences . . . . .	115
S4.4.1.1 Nestedness . . . . .	115
S4.4.1.2 Modularity . . . . .	116
S4.4.1.3 $H'_2$ . . . . .	116
S4.5 Figure 2 . . . . .	117
S4.6 Temporal patterns: Expanded results . . . . .	120
S4.7 Effects among variables: expanded results . . . . .	121
S4.7.1 Effects of climate and fruiting phenology on IFS . . . . .	121
S4.7.2 Biotic and abiotic predictors explaining network structure . . . . .	122
S4.8 Paths of climatic effects . . . . .	123
S4.8.1 Nestedness . . . . .	124
S4.8.2 Modularity . . . . .	128
S4.8.3 $H'_2$ . . . . .	131
S4.9 Causal effects . . . . .	135
S4.9.1 Functional space . . . . .	139
S4.9.2 $X \rightarrow \text{Birds}' \text{ functional space}$ . . . . .	139
S4.9.2.1 <i>Fruit production</i> $\rightarrow$ <i>Birds'</i> functional space . . . . .	140
S4.9.2.2 <i>Rainfall</i> $\rightarrow$ <i>Birds'</i> functional space . . . . .	140
S4.9.2.3 <i>Temperature</i> $\rightarrow$ <i>Birds'</i> functional space . . . . .	141
S4.9.2.4 <i>Rainfall</i> $\rightarrow$ <i>fruitproduction</i> $\rightarrow$ <i>Birds'</i> functional space . . . . .	142
S4.9.2.5 Plot . . . . .	142
S4.9.3 $X \rightarrow \text{fruits}' \text{ nutritional functional space}$ . . . . .	144
S4.9.3.1 <i>Temperature</i> $\rightarrow$ <i>fruits'</i> nutritional functional space . . . . .	145
S4.9.3.2 <i>Fruit production</i> $\rightarrow$ <i>fruits'</i> nutritional functional space . . . . .	145
S4.9.3.3 Plot . . . . .	146

S4.9.4	$X \rightarrow \text{fruits' morphological functional space}$	148
S4.9.4.1	$\text{Fruit production} \rightarrow \text{fruits' morphological functional space}$	148
S4.9.4.2	$\text{Rainfall} \rightarrow \text{Fruit production} \rightarrow \text{fruits' morphological functional space}$	149
S4.9.4.3	Plot	149
S4.9.5	$X \rightarrow \text{fruits' functional space}$	151
S4.9.5.1	$\text{Rainfall} \rightarrow \text{fruits' functional space}$	153
S4.9.5.2	$\text{Fruit production} \rightarrow \text{fruits' functional space}$	153
S4.9.5.3	$\text{Rainfall} \rightarrow \text{Fruit production} \rightarrow \text{fruits' functional space}$	154
S4.9.5.4	Plot	155
S4.9.6	Network metrics	156
S4.9.7	$X \rightarrow \text{Nestedness}$	156
S4.9.7.1	$\text{Temperature} \rightarrow \text{Nestedness}$	157
S4.9.7.2	$\text{Rainfall} \rightarrow \text{Nestedness}$	158
S4.9.7.3	$\text{Fruit production} \rightarrow \text{Nestedness}$	158
S4.9.7.4	$\text{Fruit's morphological functional space} \rightarrow \text{Nestedness}$	159
S4.9.7.5	$\text{Fruit's functional space} \rightarrow \text{Nestedness}$	160
S4.9.7.6	$\text{Birds's functional space} \rightarrow \text{Nestedness}$	160
S4.9.7.7	Plot direct effects	161
S4.9.7.8	Bottom-up effects (temperature and rainfall)	163
S4.9.7.9	Plot bottom-up effects	167
S4.9.7.10	Contrast between direct and bottom-up effect	168
S4.9.7.11	Variance decomposition	170
S4.9.8	$X \rightarrow \text{Modularity}$	171
S4.9.8.1	$\text{Rainfall} \rightarrow \text{modularity}$	172
S4.9.8.2	$\text{Fruits' nutritiona functional space} \rightarrow \text{modularity}$	173
S4.9.8.3	$\text{Fruits' functional space} \rightarrow \text{modularity}$	173
S4.9.8.4	Plot direct effects	174
S4.9.8.5	Bottom-up effects (temperature and rainfall)	176
S4.9.8.6	Plot bottom-up effects	180
S4.9.8.7	Contrast between direct and bottom-up effect	181
S4.9.8.8	Variance decomposition	183
S4.9.9	$X \rightarrow H'_2$	184
S4.9.9.1	$\text{Rainfall} \rightarrow H'_2$	185
S4.9.9.2	$\text{Temperature} \rightarrow H'_2$	185
S4.9.9.3	Plot direct effect	186
S4.9.9.4	Variance decomposition	187

## **S5 Computational environment** 188

## S1 Introduction

This document holds the code for fitting generative models from Ramírez-Mejía et al. ‘**Temporal dynamics of seed-dispersal networks: Disentangling the role of direct and indirect biotic and climatic processes**’. The code includes mathematic notation, sampling diagnostics, plotting of posterior distributions and figures in the main manuscript.

```
sapply(c('dplyr', 'ggplot2', 'lubridate', 'forcats',
        'magrittr', 'cmdstanr', 'rethinking', 'cowplot',
        'bipartite', 'igraph', 'iNEXT', 'hypervolume',
        'readxl', 'patchwork', 'ggfortify', 'tidyf', 'gtools',
        'ggsankey', 'ggridges'),
      library, character.only = T)

extrafont::loadfonts(device = 'win')

source('functions_mod_diagnostics.r')
```

## S2 Loading the data

```
phenology <- readRDS('phenology_data.rds')

alien_native <- readRDS('alien_native_prop.rds')

colnames(alien_native)[1:2] <- c('prop_alien', 'site2')

dist_mat <- readRDS('mat_distance_dites.rds')

d <- readRDS('data_for_models.rds')

d <- full_join(d, alien_native, by = c('site2', 'month'))

d <- d[, -8]

colnames(d)[9] <- 'site'

phenology <- lapply(phenology,
                     function(x) {
                       colnames(x) <- tolower(colnames(x))
                       x
                     })

d <-
  left_join(d,
            phenology$abun_phenology[, c('site', 'month', 'abu_mu')],
            by = c('site', 'month'))

d <-
  left_join(d,
            phenology$S_phenology[, c('site', 'month', 's_mu')],
            by = c('site', 'month'))

colnames(d)[grep('mu', colnames(d))] <- c('fruit_abun', 'fruit_S')

dat <- lapply(d, function(x) x)
```

```

dat$site <- as.numeric(as.factor(dat$site))

dat$HV_network <- as.vector(scale(sqrt(dat$HV_network)))
dat$size <- as.vector(scale(log(dat$size)))
dat$fruit_S <- as.vector(scale(dat$fruit_S))
dat$HV_bird <- as.vector(scale(dat$HV_bird))
dat$HV_plants_nut <- as.vector(scale(dat$HV_plants_nut))
dat$HV_plants_morfo <- as.vector(scale(sqrt(dat$HV_plants_morfo)))
dat$HV_plant <- as.vector(scale(dat$HV_plant))
dat$nestedness <- as.vector(scale(dat$nestedness))

dat$H2[which(dat$H2 >= 0.99)] <- 0.9

names(dat)[1] <- 'network_size'

dat$N <- length(dat$network_size)
dat$N_sites <- max(dat$site)
dat$N_month <- max(dat$month)
dat$dist_sites <- dist_mat

codes_sites <-
  unique(tibble(cod = dat$site,
    label = d$site))

```

## S3 Models

### S3.1 Mathematic notation

The generative model combines a set of time series Gaussian process hierarchical models to assess the causal effect of climatic, phenological, and interaction functional space variables to explain temporal dynamics of seed dispersal networks. Therefore, generative models only varies on the final response variable (i.e., the network metric), its likelihood function and prior probabilities:

#### S3.1.0.1 Nestedness (likelihood and priors)

$$\begin{aligned}
Nestedness_i &\sim \text{Student-T}(\nu = 7, \mu_i, \sigma) \\
\mu_i &= \alpha_{site_i} + f_{month_i} + \beta_1 \cdot temperature_i + \\
&\quad \beta_2 \cdot rainfall_i + \beta_3 \cdot fruit\ production_i + \\
&\quad \beta_4 \cdot fruit\ S_i + \beta_5 \cdot nut.FS_i + \\
&\quad \beta_6 \cdot morph.FS_i + \beta_7 \cdot nut.FS_i + \\
&\quad \beta_8 \cdot total\ fruit\ FS_i + \beta_9 \cdot bird.FS_i \\
K1[i, j] &= \phi^2 \cdot \exp(-\rho^2 \cdot D_{ij}) + \delta_{ij} \cdot 0.01 \\
K1 &= L_{K1} \cdot L_{K1}^T \\
\alpha_{site} &= L_{K1} \cdot Z \\
\phi &\sim \exp(4) \\
\rho &\sim \exp(1) \\
Z &\sim \mathcal{N}(0, 0.25) \\
K2[i, j] &= \sigma^{12} \cdot \exp\left(-\frac{2 \cdot \sin^2(\pi|t_i - t_j|/period)}{\zeta^2}\right) \\
K2 &= L_{K2} \cdot L_{K2}^T \\
f_{month} &= L_{K2} \cdot \eta \\
\sigma^{12} &\sim \text{cauchy}(0, 1) \\
\zeta &\sim \text{inv-gamma}(5, 5) \\
\eta &\sim \mathcal{N}(0, 0.5) \\
\beta_{1,...,9} &\sim \mathcal{N}(0, 0.5) \\
\sigma &\sim \exp(1)
\end{aligned}$$

### S3.1.0.2 $H'_2$ (likelihood and priors)

$$\begin{aligned}
H'_{2,i} &\sim \text{beta}(\tau, \psi) \\
\text{inv-logit}(\mu_i) &= \alpha_{site_i} + f_{month_i} + \beta_1 \cdot temperature_i + \\
&\quad \beta_2 \cdot rainfall_i + \beta_3 \cdot fruit\ production_i + \\
&\quad \beta_4 \cdot fruit\ S_i + \beta_5 \cdot nut.FS_i + \\
&\quad \beta_6 \cdot morph.FS_i + \beta_7 \cdot nut.FS_i + \\
&\quad \beta_8 \cdot total\ fruit\ FS_i + \beta_9 \cdot bird.FS_i \\
\tau &= \mu \cdot \sigma \\
\psi &= (1 - \mu) \cdot \sigma
\end{aligned}$$

$$\begin{aligned}
K1[i, j] &= \phi^2 \cdot \exp(-\rho^2 \cdot D_{ij}) + \delta_{ij} \cdot 0.01 \\
K1 &= L_{K1} \cdot L_{K1}^T \\
\alpha_{site} &= L_{K1} \cdot Z \\
\phi &\sim \exp(4) \\
\rho &\sim \exp(1) \\
Z &\sim \mathcal{N}(0, 0.25) \\
K2[i, j] &= \sigma_1^2 \cdot \exp\left(-\frac{2 \cdot \sin^2(\pi|t_i - t_j|/\text{period})}{\zeta^2}\right) \\
K2 &= L_{K2} \cdot L_{K2}^T \\
f_{month} &= L_{K2} \cdot \eta \\
\sigma_1 &\sim \text{cauchy}(0, 1) \\
\zeta &\sim \text{inv-gamma}(5, 5) \\
\eta &\sim \mathcal{N}(0, 0.5) \\
\beta_{1,\dots,9} &\sim \mathcal{N}(0, 0.5) \\
\sigma &\sim \exp(1)
\end{aligned}$$

### S3.1.0.3 Modularity (likelihood and priors)

$$\begin{aligned}
Modularity_i &\sim \text{beta}(\tau, \psi) \\
\text{inv-logit}(\mu_i) &= \alpha_{site_i} + f_{month_i} + \beta_1 \cdot \text{temperature}_i + \\
&\quad \beta_2 \cdot \text{rainfall}_i + \beta_3 \cdot \text{fruit production}_i + \\
&\quad \beta_4 \cdot \text{fruit } S_i + \beta_5 \cdot \text{nut.FS}_i + \\
&\quad \beta_6 \cdot \text{morph.FS}_i + \beta_7 \cdot \text{nut.FS}_i + \\
&\quad \beta_8 \cdot \text{total fruit FS}_i + \beta_9 \cdot \text{bird.FS}_i
\end{aligned}$$

$$\tau = \mu \cdot \sigma$$

$$\psi = (1 - \mu) \cdot \sigma$$

$$K1[i, j] = \phi^2 \cdot \exp(-\rho^2 \cdot D_{ij}) + \delta_{ij} \cdot 0.01$$

$$K1 = L_{K1} \cdot L_{K1}^T$$

$$\alpha_{site} = L_{K1} \cdot Z$$

$$\phi \sim \exp(4)$$

$$\rho \sim \exp(1)$$

$$Z \sim \mathcal{N}(0, 0.25)$$

$$K2[i, j] = \sigma_1^2 \cdot \exp\left(-\frac{2 \cdot \sin^2(\pi|t_i - t_j|/\text{period})}{\zeta^2}\right)$$

$$K2 = L_{K2} \cdot L_{K2}^T$$

$$f_{month} = L_{K2} \cdot \eta$$

$$\sigma_1 \sim \text{cauchy}(0, 1)$$

$$\zeta \sim \text{inv-gamma}(5, 5)$$

$$\eta \sim \mathcal{N}(0, 1)$$

$$\beta_{1, \dots, 9} \sim \mathcal{N}(0, 0.5)$$

$$\sigma \sim \exp(2)$$

### S3.1.1 Overall structure

Let  $Y$  be a network metric and  $\kappa$  its likelihood function. Then, the structure of the generative model given the linear equations will be:

$$Y_i \sim \kappa(\mu_i)$$

$$\mu_i = \alpha 1_{site_i} + f1_{month_i} + \beta 1 \cdot temperature_i +$$

$$\beta 2 \cdot rainfall_i + \beta 3 \cdot fruit\ production_i +$$

$$\beta 4 \cdot fruit\ S_i + \beta 5 \cdot nut.FS_i +$$

$$\beta 6 \cdot morph.FS_i + \beta 7 \cdot nut.FS_i +$$

$$\beta 8 \cdot total\ fruit\ FS_i + \beta 9 \cdot bird.FS_i$$

$$total\ fruit\ FS_i \sim \text{Student-T}(\nu = 7, \mu 1_i, \sigma 1)$$

$$\mu 1_i = \alpha 2_{site_i} + f2_{month_i} + \beta 10 \cdot temperature_i +$$

$$\beta 11 \cdot rainfall_i + \beta 12 \cdot fruit\ production_i +$$

$$\beta 13 \cdot fruit\ S_i + \beta 14 \cdot nut.FS_i +$$

$$\beta 15 \cdot morph.FS_i + \beta 16 \cdot nut.FS_i +$$

$$\beta 17 \cdot bird.FS_i$$

$$nut.FS_i \sim \text{Student-T}(\nu = 15, \mu 2_i, \sigma 2)$$

$$\mu 2_i = \alpha 3_{site_i} + f3_{month_i} + \beta 18 \cdot temperature_i +$$

$$\beta 19 \cdot rainfall_i + \beta 20 \cdot fruit\ production_i +$$

$$\beta 21 \cdot fruit\ S_i + \beta 22 \cdot bird.FS_i$$

$$morph.FS_i \sim \text{Student-T}(\nu = 7, \mu 3_i, \sigma 3)$$

$$\mu 3_i = \alpha 4_{site_i} + f4_{month_i} + \beta 23 \cdot temperature_i +$$

$$\beta 24 \cdot rainfall_i + \beta 25 \cdot fruit\ production_i +$$

$$\beta 26 \cdot fruit\ S_i + \beta 27 \cdot bird.FS_i$$

$$bird.FS_i \sim \text{Student-T}(\nu = 7, \mu 4_i, \sigma 4)$$

$$\mu 4_i = \alpha 5_{site_i} + f5_{month_i} + \beta 28 \cdot temperature_i +$$

$$\beta 29 \cdot rainfall_i + \beta 30 \cdot fruit\ production_i +$$

$$\beta 31 \cdot fruit\ S_i$$

$$fruit\ production_i \sim \text{Student-T}(\nu = 7, \mu 5_i, \sigma 5)$$

$$\mu 5_i = \alpha 6_{site_i} + f6_{month_i} + \beta 31 \cdot temperature_i +$$

$$\beta 32 \cdot rainfall_i$$

$$fruit\ S_i \sim \text{Student-T}(\nu = 7, \mu 6_i, \sigma 6)$$

$$\mu 6_i = \alpha 7_{site_i} + f7_{month_i} + \beta 33 \cdot temperature_i +$$

$$\beta 34 \cdot rainfall_i$$

Prior probabilities:

$$\begin{aligned}
K1[i, j]_{1,\dots,7} &= \phi_{1,\dots,7}^2 \cdot \exp(-\rho_{1,\dots,7}^2 \cdot D_{ij}) + \delta i j_{1,\dots,7} \cdot 0.01 \\
K1_{1,\dots,7} &= L_{K1_{1,\dots,7}} \cdot L_{K1_{1,\dots,7}}^T \\
\alpha_{site\ 1,\dots,7} &= L_{K1_{1,\dots,7}} \cdot Z_{1,\dots,7} \\
\phi_{1,\dots,7} &\sim \exp(4) \\
\rho_{1,\dots,7} &\sim \exp(1) \\
Z_{1,\dots,7} &\sim \mathcal{N}(0, 0.5) \\
K2[i, j]_{1,\dots,7} &= \sigma 7_{1,\dots,7}^2 \cdot \exp\left(-\frac{2 \cdot \sin^2(\pi|t_i - t_j|/\text{period})}{\zeta_{1,\dots,7}^2}\right) \\
K2_{1,\dots,7} &= L_{K2_{1,\dots,7}} \cdot L_{K2_{1,\dots,7}}^T \\
f_{month\ 1,\dots,7} &= L_{K2_{1,\dots,7}} \cdot \eta_{1,\dots,7} \\
\sigma 7_{1,\dots,7} &\sim \text{cauchy}(0, 1) \\
\zeta_{1,\dots,7} &\sim \text{inv-gamma}(5, 5) \\
\eta_{1,\dots,7} &\sim \mathcal{N}(0, 0.5) \\
\beta_{1,\dots,7} &\sim \mathcal{N}(0, 0.5) \\
\sigma_{1,\dots,6} &\sim \exp(1)
\end{aligned}$$

## S3.2 Nestedness

### S3.2.1 Stan code

```

cat(file = 'Nestedness_global.stan',
""

functions {
    vector GP_periodic(int period,           // periodicity
                        real gamma,          // smoothing term of the GP
                        real sigma,          // noise parameter
                        vector eta);        // latent variable for each month

    int M = period;
    matrix[M, M] K;
    matrix[M, M] L_K;

    for (i in 1:(M - 1)) {
        for (j in (i+1):M) {
            real distance = abs(i - j);
            real periodic_distance = fmin(distance, period - distance);
            K[i, j] = sigma^2 * exp(-2 * square(sin(pi()*periodic_distance/period))/gamma^2);
            K[j, i] = K[i, j];    // filling the lower triangle
        }
        K[i, i] = sigma^2 + 1e-9;   // small values to guarantee stability
    }
}

```

```

        K[M, M] = sigma^2 + 1e-9; // small values to guarantee stability
        return cholesky_decompose(K) * eta;
    }

matrix GP_quadratic(matrix x,
                     real eta,
                     real rho,
                     real delta) {
    int N = dims(x)[1];
    matrix[N, N] K;
    matrix[N, N] L_K;

    for (i in 1:(N-1)) {
        K[i, i] = eta + delta;
        for (j in (i+1):N) {
            K[i, j] = square(eta) * exp(-rho * square(x[i, j]));
            K[j, i] = K[i, j];
        }
    }

    K[N, N] = eta + delta;
    L_K = cholesky_decompose(K);
    return L_K;
}
}

data{
    int N;
    int N_sites;
    int N_month;
    array[N] int network_size;
    vector[N] modularity;
    vector[N] nestedness;
    vector[N] H2;
    array[N] int month;
    array[N] int site;
    vector[N] HV_network;
    vector[N] HV_plant;
    vector[N] HV_plants_morfo;
    vector[N] HV_plants_nut;
    vector[N] HV_bird;
    vector[N] z_temperature;
    vector[N] z_rainfall;
    vector[N] fruit_S;
    vector[N] fruit_abun;
    matrix[N_sites, N_sites] dist_sites;
}

parameters {

    //////////////////////////////// Fruit abundance //////////////////////

    // GP periodic
    real<lower = 0> gamma_fruitS;
    real<lower = 0> sigma_f_fruitS;
    vector[N_month] eta_fruitS;
    // GP quadratic
    vector[N_sites] z_sites_fruitS;
    real<lower = 0> eta_site_fruitS;
    real<lower = 0> rho_site_fruitS;

    // Pars linear model
    real beta_temp_fruitS;
}

```

```

real beta_rain_fruitS;
// noise for likelihood function
real<lower = 0> sigma_fruitS;

//////////////////////////// Fruit abundance //////////////////////

// GP periodic
real<lower = 0> gamma_fruitAB;
real<lower = 0> sigma_f_fruitAB;
vector[N_month] eta_fruitAB;
// GP quadratic
vector[N_sites] z_sites_fruitAB;
real<lower = 0> eta_site_fruitAB;
real<lower = 0> rho_site_fruitAB;

// Pars linear model

real beta_temp_fruitAB;
real beta_rain_fruitAB;
// noise for likelihood function
real<lower = 0> sigma_fruitAB;

//////////////////////////// Fru. Nut. HV //////////////////////

// GP periodic
real<lower = 0> gamma_nut_HV;
real<lower = 0> sigma_f_nut_HV;
vector[N_month] eta_nut_HV;
// GP quadratic
vector[N_sites] z_sites_nut_HV;
real<lower = 0> eta_site_nut_hv;
real<lower = 0> rho_site_nut_hv;

// Pars linear model

real beta_temp_nut_HV;
real beta_rain_nut_HV;
real beta_fruitAB_nut_HV;
real beta_fruitS_nut_HV;
real beta_BIRD_nut_HV;
// noise for likelihood function
real<lower = 0> sigma_nut_HV;

//////////////////////////// Fru. morpho. HV //////////////////////

// GP periodic
real<lower = 0> gamma_mor_HV;
real<lower = 0> sigma_f_mor_HV;
vector[N_month] eta_mor_HV;
// GP quadratic
vector[N_sites] z_sites_mor_HV;
real<lower = 0> eta_site_mor_hv;
real<lower = 0> rho_site_mor_hv;

// Pars linear model

real beta_temp_mor_HV;
real beta_rain_mor_HV;
real beta_fruitAB_mor_HV;
real beta_fruitS_mor_HV;
real beta_BIRD_mor_HV;
// noise for likelihood function
real<lower = 0> sigma_mor_HV;

```

```

real nu;

// ////////////////////////////// Plants HV //////////////////////////////
// GP periodic
real<lower = 0> gamma_PLAN_HV;
real<lower = 0> sigma_f_PLAN_HV;
vector[N_month] eta_PLAN_HV;
// GP quadratic
vector[N_sites] z_sites_PLAN_HV;
real<lower = 0> eta_site_PLAN_hv;
real<lower = 0> rho_site_PLAN_hv;

// Pars linear model
real beta_temp_PLAN_HV;
real beta_rain_PLAN_HV;
real beta_fruitAB_PLAN_HV;
real beta_fruits_PLAN_HV;
real beta_nut_PLAN_HV;
real beta_mor_PLAN_HV;
real beta_BIRD_PLAN_HV;
// noise for likelihood function
real<lower = 0> sigma_PLAN_HV;

////////////////////////////// Birds HV ///////////////////////////////
// GP periodic
real<lower = 0> gamma_BIRD_HV;
real<lower = 0> sigma_f_BIRD_HV;
vector[N_month] eta_BIRD_HV;
// GP quadratic
vector[N_sites] z_sites_BIRD_HV;
real<lower = 0> eta_site_BIRD_hv;
real<lower = 0> rho_site_BIRD_hv;

// Pars linear model
real beta_temp_BIRD_HV;
real beta_rain_BIRD_HV;
real beta_fruitAB_BIRD_HV;
real beta_fruits_BIRD_HV;
// noise for likelihood function
real<lower = 0> sigma_BIRD_HV;

//////////////////////////// Network metrics /////////////////////
// ////////////////////////////// Nestedness ///////////////////
// GP periodic
real<lower = 0> gamma_NSS;
real<lower = 0> sigma_f_NSS;
vector[N_month] eta_NSS;
// GP quadratic
vector[N_sites] z_sites_NSS;
real<lower = 0> eta_site_NSS;
real<lower = 0> rho_site_NSS;

// Pars linear model
real beta_temp_NSS;
real beta_rain_NSS;
real beta_fruitAB_NSS;
real beta_fruits_NSS;
real beta_nut_NSS;
real beta_mor_NSS;
real beta_PLAN_NSS;
real beta_BIRD_NSS;
// vector[N_sites] beta_NW_NSS;
// noise for likelihood function

```

```

    real<lower = 0> sigma_NSS;
}

transformed parameters {
    //////////////////////////////////////////////////////////////////// Fruit abundance ///////////////////////////////
    // GP periodic
    vector[N_month] f_fruitAB;

    f_fruitAB = GP_periodic(
        12,
        gamma_fruitAB,
        sigma_f_fruitAB,
        eta_fruitAB
    );

    // GP quadratic
    vector[N_sites] alpha_fruitAB;
    matrix[N_sites, N_sites] L_K_fruitAB;
    L_K_fruitAB = GP_quadratic(dist_sites,
        eta_site_fruitAB,
        rho_site_fruitAB, 0.001);
    alpha_fruitAB = L_K_fruitAB * z_sites_fruitAB;

    //////////////////////////////////////////////////////////////////// Fruit richness ///////////////////////////////
    // GP periodic
    vector[N_month] f_fruitS;

    f_fruitS = GP_periodic(
        12,
        gamma_fruitS,
        sigma_f_fruitS,
        eta_fruitS
    );

    // GP quadratic
    vector[N_sites] alpha_fruitS;
    matrix[N_sites, N_sites] L_K_fruitS;
    L_K_fruitS = GP_quadratic(dist_sites,
        eta_site_fruitS,
        rho_site_fruitS, 0.001);
    alpha_fruitS = L_K_fruitS * z_sites_fruitS;

    //////////////////////////////////////////////////////////////////// Fru. Nut. HV ///////////////////////////////
    // GP periodic
    vector[N_month] f_nut_HV;

    f_nut_HV = GP_periodic(
        12,
        gamma_nut_HV,
        sigma_f_nut_HV,
        eta_nut_HV
    );

    // GP quadratic
    vector[N_sites] alpha_nut_HV;
    matrix[N_sites, N_sites] L_K_nut_HV;
    L_K_nut_HV = GP_quadratic(dist_sites,
        eta_site_nut_hv,
        rho_site_nut_hv, 0.001);
    alpha_nut_HV = L_K_nut_HV * z_sites_nut_HV;
}

```

```

//   ////////////////////////////// Fru. morpho. HV ///////////////////////
// GP periodic
vector[N_month] f_mor_HV;

f_mor_HV = GP_periodic(
    12,
    gamma_mor_HV,
    sigma_f_mor_HV,
    eta_mor_HV
);

// GP quadratic
vector[N_sites] alpha_mor_HV;
matrix[N_sites, N_sites] L_K_mor_HV;
L_K_mor_HV = GP_quadratic(dist_sites,
                           eta_site_mor_hv,
                           rho_site_mor_hv, 0.001);
alpha_mor_HV = L_K_mor_HV * z_sites_mor_HV;
//   ////////////////////////////// Plants HV ///////////////////////
// GP periodic
vector[N_month] f_PLAN_HV;

f_PLAN_HV = GP_periodic(
    12,
    gamma_PLAN_HV,
    sigma_f_PLAN_HV,
    eta_PLAN_HV
);

// GP quadratic
vector[N_sites] alpha_PLAN_HV;
matrix[N_sites, N_sites] L_K_PLAN_HV;
L_K_PLAN_HV = GP_quadratic(dist_sites,
                           eta_site_PLAN_hv,
                           rho_site_PLAN_hv, 0.001);
alpha_PLAN_HV = L_K_PLAN_HV * z_sites_PLAN_HV;
////////////////// Birds HV ///////////////////////
// GP periodic
vector[N_month] f_BIRD_HV;

f_BIRD_HV = GP_periodic(
    12,
    gamma_BIRD_HV,
    sigma_f_BIRD_HV,
    eta_BIRD_HV
);

// GP quadratic
vector[N_sites] alpha_BIRD_HV;
matrix[N_sites, N_sites] L_K_BIRD_HV;
L_K_BIRD_HV = GP_quadratic(dist_sites,
                           eta_site_BIRD_hv,
                           rho_site_BIRD_hv, 0.001);
alpha_BIRD_HV = L_K_BIRD_HV * z_sites_BIRD_HV;
//   ////////////////////////////// Nestedness ///////////////////////
// GP periodic
vector[N_month] f_NSS;

f_NSS = GP_periodic(
    12,

```

```

        gamma_NSS,
        sigma_f_NSS,
        eta_NSS
    );

    // GP quadratic
    vector[N_sites] alpha_NSS;
    matrix[N_sites, N_sites] L_K_NSS;
    L_K_NSS = GP_quadratic(dist_sites,
        eta_site_NSS,
        rho_site_NSS, 0.001);
    alpha_NSS = L_K_NSS * z_sites_NSS;
}

model {
    //////////////////// Fruit abundance ///////////////////
    // priors for periodic GP
    eta_fruitAB ~ normal(0, 0.5);
    gamma_fruitAB ~ inv_gamma(5, 5);
    sigma_f_fruitAB ~ cauchy(0, 1);
    // GP quadratic
    z_sites_fruitAB ~ normal(0, 0.25);
    eta_site_fruitAB ~ exponential(4);
    rho_site_fruitAB ~ exponential(1);
    // Pars linear model
    beta_temp_fruitAB ~ normal(0, 0.5);
    beta_rain_fruitAB ~ normal(0, 0.5);
    // noise parameter of the likelihood
    sigma_fruitAB ~ exponential(1);

    //////////////////// Fruit richness ///////////////////
    // priors for periodic GP
    eta_fruitS ~ normal(0, 0.5);
    gamma_fruitS ~ inv_gamma(5, 5);
    sigma_f_fruitS ~ cauchy(0, 1);
    // GP quadratic
    z_sites_fruitS ~ normal(0, 0.25);
    eta_site_fruitS ~ exponential(4);
    rho_site_fruitS ~ exponential(1);
    // Pars linear model
    beta_temp_fruitS ~ normal(0, 0.5);
    beta_rain_fruitS ~ normal(0, 0.5);
    // noise parameter of the likelihood
    sigma_fruitS ~ exponential(1);

    //////////////////// Fru. Nut. HV ///////////////////
    // priors for periodic GP
    eta_nut_HV ~ normal(0, 0.5);
    gamma_nut_HV ~ inv_gamma(5, 5);
    sigma_f_nut_HV ~ cauchy(0, 1);
    // GP quadratic
    z_sites_nut_HV ~ normal(0, 0.25);
    eta_site_nut_hv ~ exponential(4);
    rho_site_nut_hv ~ exponential(1);
    // Pars linear model
    beta_temp_nut_HV ~ normal(0, 0.5);
    beta_rain_nut_HV ~ normal(0, 0.5);
    beta_fruitAB_nut_HV ~ normal(0, 0.5);
    beta_fruitS_nut_HV ~ normal(0, 0.5);
    beta_BIRD_nut_HV ~ normal(0, 0.5);
    // noise parameter of the likelihood
    sigma_nut_HV ~ exponential(1);
}

```

```

////////////////////////////////////////////////////////////////// Fru. morpho. HV //////////////////////////////////////////////////////////////////
// priors for periodic GP
eta_mor_HV ~ normal(0, 0.5);
gamma_mor_HV ~ inv_gamma(5, 5);
sigma_f_mor_HV ~ cauchy(0, 1);
// GP quadratic
z_sites_mor_HV ~ normal(0, 0.25);
eta_site_mor_hv ~ exponential(4);
rho_site_mor_hv ~ exponential(1);
// Pars linear model
beta_temp_mor_HV ~ normal(0, 0.5);
beta_rain_mor_HV ~ normal(0, 0.5);
beta_fruitAB_mor_HV ~ normal(0, 0.5);
beta_fruitS_mor_HV ~ normal(0, 0.5);
beta_BIRD_mor_HV ~ normal(0, 0.5);
// noise parameter of the likelihood
sigma_mor_HV ~ exponential(1);

////////////////////////////////////////////////////////////////// Plants HV //////////////////////////////////////////////////////////////////
// priors for periodic GP
eta_PLAN_HV ~ normal(0, 0.5);
gamma_PLAN_HV ~ inv_gamma(5, 5);
sigma_f_PLAN_HV ~ cauchy(0, 1);
// GP quadratic
z_sites_PLAN_HV ~ normal(0, 0.5);
eta_site_PLAN_hv ~ exponential(4);
rho_site_PLAN_hv ~ exponential(1);
// Pars linear model
beta_temp_PLAN_HV ~ normal(0, 0.5);
beta_rain_PLAN_HV ~ normal(0, 0.5);
beta_fruitAB_PLAN_HV ~ normal(0, 0.5);
beta_fruitS_PLAN_HV ~ normal(0, 0.5);
beta_nut_PLAN_HV ~ normal(0, 0.5);
beta_mor_PLAN_HV ~ normal(0, 0.5);
beta_BIRD_PLAN_HV ~ normal(0, 0.5);
// noise parameter of the likelihood
sigma_PLAN_HV ~ exponential(1);

////////////////////////////////////////////////////////////////// Birds HV //////////////////////////////////////////////////////////////////
// priors for periodic GP
eta_BIRD_HV ~ normal(0, 0.5);
gamma_BIRD_HV ~ inv_gamma(5, 5);
sigma_f_BIRD_HV ~ cauchy(0, 1);
// GP quadratic
z_sites_BIRD_HV ~ normal(0, 0.5);
eta_site_BIRD_hv ~ exponential(4);
rho_site_BIRD_hv ~ exponential(1);
// Pars linear model
beta_temp_BIRD_HV ~ normal(0, 0.5);
beta_rain_BIRD_HV ~ normal(0, 0.5);
beta_fruitAB_BIRD_HV ~ normal(0, 0.5);
beta_fruitS_BIRD_HV ~ normal(0, 0.5);
// noise parameter of the likelihood
sigma_BIRD_HV ~ exponential(1);

////////////////////////////////////////////////////////////////// Nestedness //////////////////////////////////////////////////////////////////
// priors for periodic GP
eta_NSS ~ normal(0, 0.5);
gamma_NSS ~ inv_gamma(5, 5);
sigma_f_NSS ~ cauchy(0, 1);
// GP quadratic
z_sites_NSS ~ normal(0, 0.25);

```

```

eta_site_NSS ~ exponential(4);
rho_site_NSS ~ exponential(1);
// Pars linear model
beta_temp_NSS ~ normal(0, 0.5);
beta_rain_NSS ~ normal(0, 0.5);
beta_fruitAB_NSS ~ normal(0, 0.5);
beta_fruitsS_NSS ~ normal(0, 0.5);
beta_nut_NSS ~ normal(0, 0.5);
beta_mor_NSS ~ normal(0, 0.5);
beta_PLAN_NSS ~ normal(0, 0.5);
beta_BIRD_NSS ~ normal(0, 0.5);
// beta_NW_NSS ~ normal(0, 0.5);
// noise parameter of the likelihood
sigma_NSS ~ exponential(1);

////////////////// fruit abundance ///////////////////
////////////////// fruit richness ///////////////////
for (i in 1:N) {
    fruit_abun[i] ~ student_t(7, alpha_fruitAB[site[i]] + f_fruitAB[month[i]] +
        beta_temp_fruitAB * z_temperature[i] +
        beta_rain_fruitAB * z_rainfall[i],
        sigma_fruitAB);
}

////////////////// Fru. Nut. HV ///////////////////
////////////////// Fru. morpho. HV ///////////////////
for (i in 1:N) {
    HV_plants_nut[i] ~ student_t(7, alpha_nut_HV[site[i]] + f_nut_HV[month[i]] +
        beta_temp_nut_HV * z_temperature[i] +
        beta_rain_nut_HV * z_rainfall[i] +
        beta_BIRD_nut_HV * HV_bird[i] +
        beta_fruitAB_nut_HV * fruit_abun[i] +
        beta_fruitsS_nut_HV * fruit_S[i],
        sigma_nut_HV);
}

// ////////////////// Plants HV ///////////////////
for (i in 1:N) {
    HV_plant[i] ~ student_t(7, alpha_PLAN_HV[site[i]] + f_PLAN_HV[month[i]] +
        beta_temp_PLAN_HV * z_temperature[i] +
        beta_rain_PLAN_HV * z_rainfall[i] +
        beta_fruitAB_PLAN_HV * fruit_abun[i] +
        beta_fruitsS_PLAN_HV * fruit_S[i] +
        beta_nut_PLAN_HV * HV_plants_nut[i] +
}

```

```

        beta_mor_PLAN_HV * HV_plants_morfo[i] +
        beta_BIRD_PLAN_HV * HV_bird[i],
        sigma_PLAN_HV);
}

////////// Birds HV ///////////
////////// Birds HV ///////////
for (i in 1:N) {
    HV_bird[i] ~ student_t(7, alpha_BIRD_HV[site[i]] + f_BIRD_HV[month[i]] +
        beta_temp_BIRD_HV * z_temperature[i] +
        beta_rain_BIRD_HV * z_rainfall[i] +
        beta_fruitAB_BIRD_HV * fruit_abun[i] +
        beta_fruitS_BIRD_HV * fruit_S[i],
        sigma_BIRD_HV);
}

// ////////////// Nestedness ///////////////
// ////////////// Nestedness ///////////////
for (i in 1:N) {
    nestedness[i] ~ student_t(7, alpha_NSS[site[i]] + f_NSS[month[i]] +
        beta_temp_NSS * z_temperature[i] +
        beta_rain_NSS * z_rainfall[i] +
        beta_fruitAB_NSS * fruit_abun[i] +
        beta_fruitS_NSS * fruit_S[i] +
        beta_nut_NSS * HV_plants_nut[i] +
        beta_mor_NSS * HV_plants_morfo[i] +
        beta_PLAN_NSS * HV_plant[i] +
        beta_BIRD_NSS * HV_bird[i],
        sigma_NSS);
}

generated quantities {

    /////////// Fruit abundance ///////////
    array[N] real ppcheck_fruitAB;
    vector[N] mu_fruitAB;

    for (i in 1:N) {
        mu_fruitAB[i] = alpha_fruitAB[site[i]] + f_fruitAB[month[i]] +
            beta_temp_fruitAB * z_temperature[i] +
            beta_rain_fruitAB * z_rainfall[i];
    }

    ppcheck_fruitAB = student_t_rng(7, mu_fruitAB, sigma_fruitAB);

    /////////// Fruit richness ///////////
    array[N] real ppcheck_fruitS;
    vector[N] mu_fruitS;

    for (i in 1:N) {
        mu_fruitS[i] = alpha_fruitS[site[i]] + f_fruitS[month[i]] +
            beta_temp_fruitS * z_temperature[i] +
            beta_rain_fruitS * z_rainfall[i];
    }

    ppcheck_fruitS = student_t_rng(7, mu_fruitS, sigma_fruitS);

    /////////// Fru. Nut. HV ///////////
    array[N] real ppcheck_nut_HV;
    vector[N] mu_nut_HV;
}

```

```

for (i in 1:N) {
  mu_nut_HV[i] = alpha_nut_HV[site[i]] + f_nut_HV[month[i]] +
    beta_temp_nut_HV * z_temperature[i] +
    beta_rain_nut_HV * z_rainfall[i] +
    beta_fruitAB_nut_HV * fruit_abun[i] +
    beta_fruitS_nut_HV * fruit_S[i] +
    beta_BIRD_nut_HV * HV_bird[i];
}

ppcheck_nut_HV = student_t_rng(7, mu_nut_HV, sigma_nut_HV);
////////////////////////////////////////////////////////////////// Fru. morpho. HV //////////////////////////////////////////////////////////////////
array[N] real ppcheck_mor_HV;
vector[N] mu_mor_HV;

for (i in 1:N) {
  mu_mor_HV[i] = alpha_mor_HV[site[i]] + f_mor_HV[month[i]] +
    beta_temp_mor_HV * z_temperature[i] +
    beta_rain_mor_HV * z_rainfall[i] +
    beta_fruitAB_mor_HV * fruit_abun[i] +
    beta_fruitS_mor_HV * fruit_S[i] +
    beta_BIRD_mor_HV * HV_bird[i];
}

ppcheck_mor_HV = student_t_rng(15, mu_mor_HV, sigma_mor_HV);

////////////////////////////////////////////////////////////////// Plants HV //////////////////////////////////////////////////////////////////
array[N] real ppcheck_PLAN_HV;
vector[N] mu_PLAN_HV;

for (i in 1:N) {
  mu_PLAN_HV[i] = alpha_PLAN_HV[site[i]] + f_PLAN_HV[month[i]] +
    beta_temp_PLAN_HV * z_temperature[i] +
    beta_rain_PLAN_HV * z_rainfall[i] +
    beta_fruitAB_PLAN_HV * fruit_abun[i] +
    beta_fruitS_PLAN_HV * fruit_S[i] +
    beta_nut_PLAN_HV * HV_plants_nut[i] +
    beta_mor_PLAN_HV * HV_plants_morfo[i] +
    beta_BIRD_PLAN_HV * HV_bird[i];
}

ppcheck_PLAN_HV = student_t_rng(7, mu_PLAN_HV, sigma_PLAN_HV);

////////////////////////////////////////////////////////////////// Birds HV //////////////////////////////////////////////////////////////////
array[N] real ppcheck_BIRD_HV;
vector[N] mu_BIRD_HV;

for (i in 1:N) {
  mu_BIRD_HV[i] = alpha_BIRD_HV[site[i]] + f_BIRD_HV[month[i]] +
    beta_temp_BIRD_HV * z_temperature[i] +
    beta_rain_BIRD_HV * z_rainfall[i] +
    beta_fruitAB_BIRD_HV * fruit_abun[i] +
    beta_fruitS_BIRD_HV * fruit_S[i];
}

ppcheck_BIRD_HV = student_t_rng(7, mu_BIRD_HV, sigma_BIRD_HV);

////////////////////////////////////////////////////////////////// Nestedness //////////////////////////////////////////////////////////////////
array[N] real ppcheck_NSS;
vector[N] mu_NSS;

for (i in 1:N) {

```

```

        mu_NSS[i] = alpha_NSS[site[i]] + f_NSS[month[i]] +
            beta_temp_NSS * z_temperature[i] +
            beta_rain_NSS * z_rainfall[i] +
            beta_fruitAB_NSS * fruit_abun[i] +
            beta_fruitS_NSS * fruit_S[i] +
            beta_nut_NSS * HV_plants_nut[i] +
            beta_mor_NSS * HV_plants_morfo[i] +
            beta_PLAN_NSS * HV_plant[i] +
            beta_BIRD_NSS * HV_bird[i];
    }

    ppcheck_NSS = student_t_rng(7, mu_NSS, sigma_NSS);
}

"
)

```

### S3.2.2 Fitting the model

```

file <- paste0(getwd(), '/Nestedness_global.stan')
fit_nestednessG <- cmdstan_model(file, compile = T)

mod_nestednessG <-
  fit_nestednessG$sample(
    data = dat,
    iter_sampling = 2e3,
    iter_warmup = 500,
    thin = 5,
    chains = 3,
    parallel_chains = 3,
    seed = 5
  )

```

### S3.2.3 Sampling diagnostics

```

summary_nestednessG <- mod_nestednessG$summary()

par(mfrow = c(3, 3), mar = c(4, 4, 2, 1))
mod_diagnostics(mod_nestednessG,
  summary_nestednessG[grep('fruitAB', summary_nestednessG$variable), ])
title('Fruit abundance')
mod_diagnostics(mod_nestednessG,
  summary_nestednessG[grep('fruitS', summary_nestednessG$variable), ])
title('Fruit richness')
mod_diagnostics(mod_nestednessG,
  summary_nestednessG[grep('nut_HV', summary_nestednessG$variable), ])
title('Frut. Nut. HV')
mod_diagnostics(mod_nestednessG,
  summary_nestednessG[grep('mor_HV', summary_nestednessG$variable), ])
title('Frut. Morpho. HV')
mod_diagnostics(mod_nestednessG,
  summary_nestednessG[grep('PLAN_HV', summary_nestednessG$variable), ])
title('Plants HV')
mod_diagnostics(mod_nestednessG,
  summary_nestednessG[grep('BIRD_HV', summary_nestednessG$variable), ])

```

```
title('Birds HV')
mod_diagnostics(mod_nestednessG,
                 summary_nestednessG[grep('NSS', summary_nestednessG$variable), ])
title('nestednessG')
par(mfrow = c(1, 1))
```

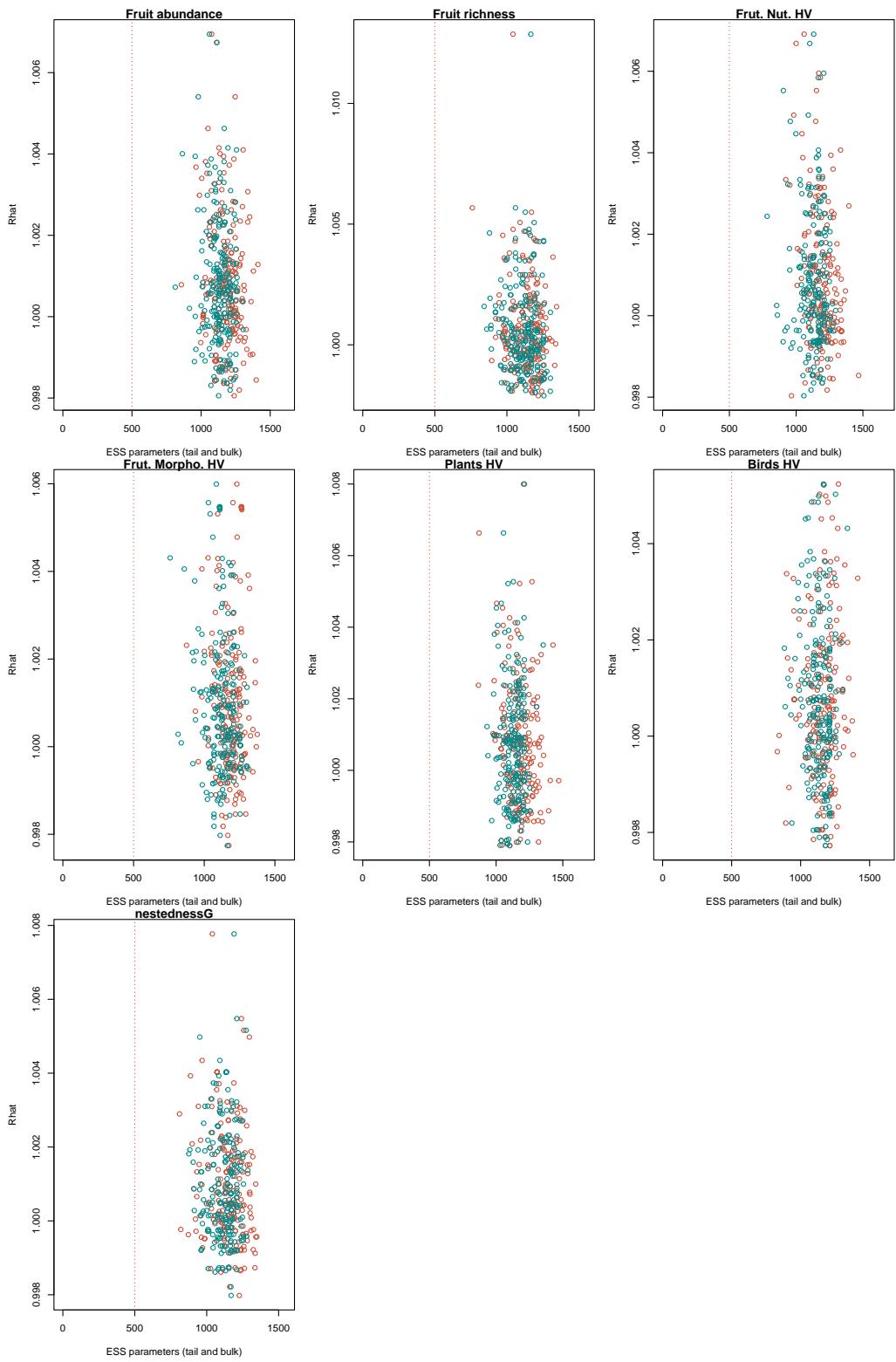


Figure S1: Rhat values vs effective sampling size (ess). Rhat  $< 1.05$  indicates that the Markov Chains converged to the same stationary distribution. ess must be at least 100 per chain in order to be reliable

```

par(mfrow = c(3, 3), mar = c(4, 4, 1, 1))
ppcheck_nestednessG <- mod_nestednessG$draws('ppcheck_fruitAB', format = 'matrix')
plot(density(dat$fruit_abun), main = '', xlab = 'Fruit abundance',
      ylim = c(0, 1))
for (i in 1:100) lines(density(ppcheck_nestednessG[i, ]), lwd = 0.1)
lines(density(dat$fruit_abun), col = 'red', lwd = 2)

ppcheck_nestednessG <- mod_nestednessG$draws('ppcheck_fruitS', format = 'matrix')
plot(density(dat$fruit_S), main = '', xlab = 'Fruit richness',
      ylim = c(0, 0.7))
for (i in 1:100) lines(density(ppcheck_nestednessG[i, ]), lwd = 0.1)
lines(density(dat$fruit_S), col = 'red', lwd = 2)

ppcheck_nestednessG <- mod_nestednessG$draws('ppcheck_nut_HV', format = 'matrix')
plot(density(dat$HV_plants_nut), main = '', xlab = 'Frut. Nut. HV',
      ylim = c(0, 0.5))
for (i in 1:100) lines(density(ppcheck_nestednessG[i, ]), lwd = 0.1)
lines(density(dat$HV_plants_nut), col = 'red', lwd = 2)

ppcheck_nestednessG <- mod_nestednessG$draws('ppcheck_mor_HV', format = 'matrix')
plot(density(dat$HV_plants_morfo), main = '', xlab = 'Frut. Morpho. HV',
      ylim = c(0, 0.5))
for (i in 1:100) lines(density(ppcheck_nestednessG[i, ]), lwd = 0.1)
lines(density(dat$HV_plants_morfo), col = 'red', lwd = 2)

ppcheck_nestednessG <- mod_nestednessG$draws('ppcheck_PLAN_HV', format = 'matrix')
plot(density(dat$HV_plant), main = '', xlab = 'Plants HV',
      ylim = c(0, 0.7))
for (i in 1:100) lines(density(ppcheck_nestednessG[i, ]), lwd = 0.1)
lines(density(dat$HV_plant), col = 'red', lwd = 2)

ppcheck_nestednessG <- mod_nestednessG$draws('ppcheck_BIRD_HV', format = 'matrix')
plot(density(dat$HV_bird), main = '', xlab = 'Birds HV',
      ylim = c(0, 0.5))
for (i in 1:100) lines(density(ppcheck_nestednessG[i, ]), lwd = 0.1)
lines(density(dat$HV_bird), col = 'red', lwd = 2)

ppcheck_nestednessG <- mod_nestednessG$draws('ppcheck_NSS', format = 'matrix')
plot(density(dat$nestedness), main = '', xlab = 'Nestedness',
      ylim = c(0, 0.5))
for (i in 1:200) lines(density(ppcheck_nestednessG[i, ]), lwd = 0.1)
lines(density(dat$nestedness), col = 'red', lwd = 2)
par(mfrow = c(1, 1))

```

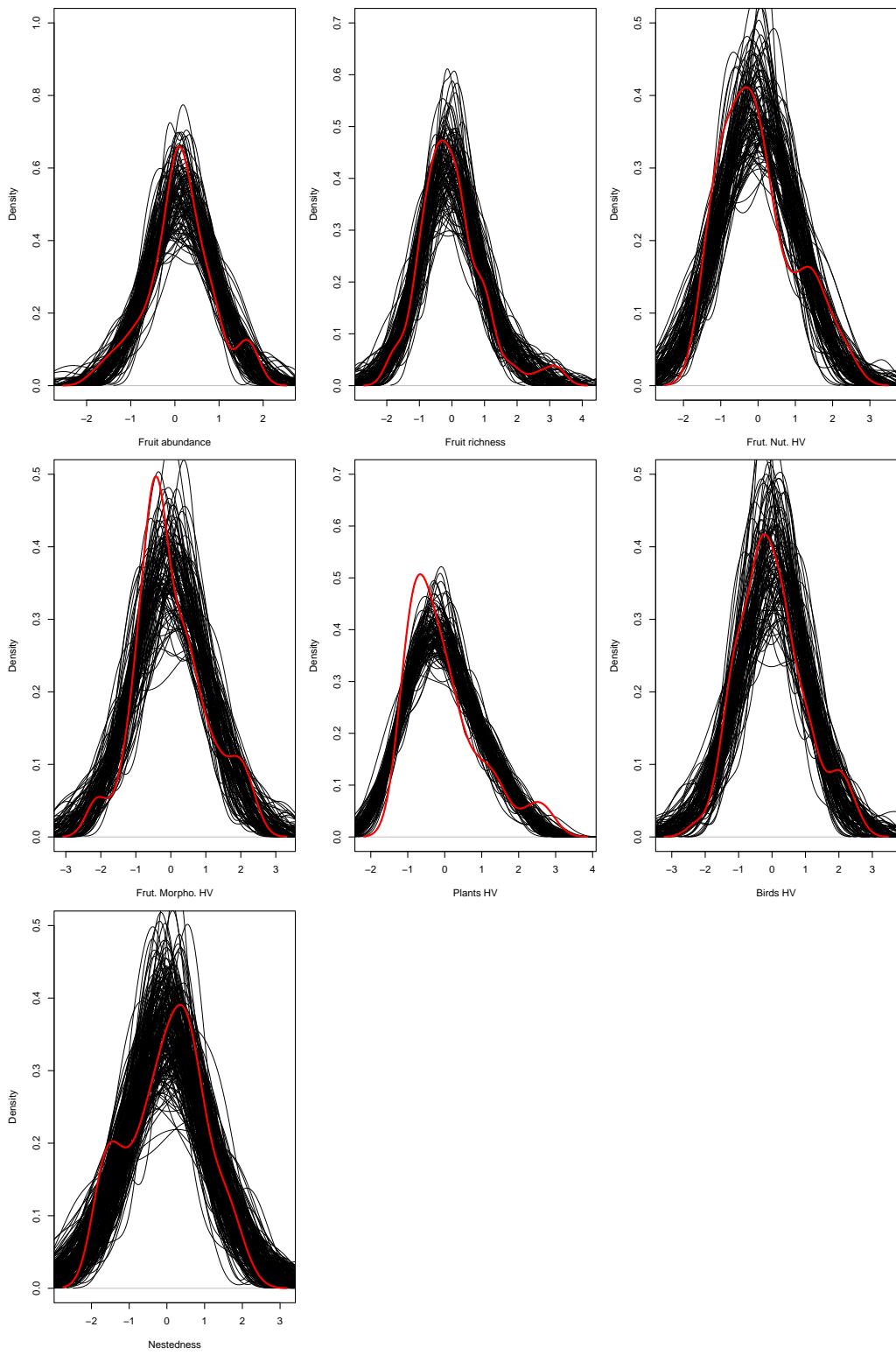


Figure S2: Posterior predictive checks of the of variables integrating the generative model explaining the nestedness of seed seed dispersal networks

### S3.2.4 Extracting posterior draws

```
pars_nestednessG <- summary_nestednessG[grep('^beta', summary_nestednessG$variable), ]$variable  
betas_nestednessG <- mod_nestednessG$draws(pars_nestednessG, format = 'df')  
betas_nestednessG <- betas_nestednessG[, -c((ncol(betas_nestednessG)-2):ncol(betas_nestednessG))]
```

Warning: Dropping 'draws\_df' class as required metadata was removed.

```
betas_nestednessG <-  
  lapply(1:ncol(betas_nestednessG),  
    function(i) {  
      x <- betas_nestednessG[[i]]  
      var <- colnames(betas_nestednessG)[[i]]  
  
      df <-  
        tibble(var = var,  
               mu = mean(x),  
               li = quantile(x, 0.025),  
               ls = quantile(x, 0.975),  
               p_pos = mean(x > 0),  
               p_neg = mean(x < 0))  
  
      df$effect <- df$p_pos > 0.7 | df$p_neg > 0.7  
      df  
    })
```

```
betas_nestednessG <- do.call('rbind', betas_nestednessG)  
betas_nestednessG_LABS <-  
  betas_nestednessG[grep('NSS', betas_nestednessG$var), ]
```

```
betas_nestednessG_z <- betas_nestednessG  
betas_nestednessG_z$var <-  
  gsub('^(beta_)([A-Za-z]*)(_)(.*)$', '\\\\2 -> \\\\4', betas_nestednessG_z$var)  
colnames(betas_nestednessG_z) <-  
  c('Directional effect', 'Mean', '0.025%', '0.975%', 'P(effect > 0)',  
    'P(effect < 0)', 'effect')  
knitr::kable(betas_nestednessG_z,  
            digits = 2,  
            caption = 'Average values, 95% CIs, and probabilities of directional effects for the relationships a')
```

Table S1: Average values, 95% CIs, and probabilities of directional effects for the relationships assessed in the generative model. An effect is marked as TRUE if  $P(\text{effect} > 0) > 0.7$  or  $P(\text{effect} < 0) > 0.7$  or.

Directional effect	Mean	0.025%	0.975%	$P(\text{effect} > 0)$	$P(\text{effect} < 0)$	effect
temp -> fruitS	-0.15	-0.41	0.12	0.14	0.86	TRUE
rain -> fruitS	0.46	0.16	0.74	1.00	0.00	TRUE
temp -> fruitAB	0.00	-0.24	0.24	0.50	0.50	FALSE
rain -> fruitAB	-0.10	-0.30	0.11	0.15	0.85	TRUE
temp -> nut_HV	0.36	0.06	0.64	0.99	0.01	TRUE
rain -> nut_HV	-0.03	-0.33	0.27	0.44	0.56	FALSE
fruitAB -> nut_HV	0.21	-0.15	0.57	0.87	0.13	TRUE
fruitS -> nut_HV	-0.02	-0.30	0.29	0.43	0.57	FALSE
BIRD -> nut_HV	0.03	-0.17	0.22	0.61	0.39	FALSE
temp -> mor_HV	0.04	-0.24	0.32	0.62	0.38	FALSE
rain -> mor_HV	0.06	-0.26	0.36	0.64	0.36	FALSE
fruitAB -> mor_HV	0.20	-0.18	0.58	0.85	0.15	TRUE
fruitS -> mor_HV	-0.02	-0.34	0.28	0.46	0.54	FALSE
BIRD -> mor_HV	0.04	-0.21	0.28	0.62	0.38	FALSE
temp -> PLAN_HV	-0.01	-0.13	0.11	0.43	0.57	FALSE
rain -> PLAN_HV	0.07	-0.06	0.20	0.84	0.16	TRUE
fruitAB -> PLAN_HV	-0.14	-0.29	0.02	0.04	0.96	TRUE
fruitS -> PLAN_HV	0.07	-0.06	0.18	0.88	0.12	TRUE
nut -> PLAN_HV	0.59	0.48	0.70	1.00	0.00	TRUE
mor -> PLAN_HV	0.47	0.38	0.56	1.00	0.00	TRUE
BIRD -> PLAN_HV	0.03	-0.06	0.13	0.74	0.26	TRUE
temp -> BIRD_HV	0.18	-0.13	0.50	0.87	0.13	TRUE
rain -> BIRD_HV	-0.13	-0.46	0.25	0.22	0.78	TRUE
fruitAB -> BIRD_HV	0.23	-0.08	0.57	0.92	0.08	TRUE
fruitS -> BIRD_HV	-0.05	-0.37	0.27	0.37	0.63	FALSE
temp -> NSS	0.18	-0.16	0.54	0.84	0.16	TRUE
rain -> NSS	-0.17	-0.50	0.16	0.16	0.84	TRUE
fruitAB -> NSS	-0.22	-0.64	0.20	0.15	0.85	TRUE
fruitS -> NSS	0.02	-0.33	0.35	0.56	0.44	FALSE

Directional effect	Mean	0.025%	0.975%	P(effect > 0)	P(effect < 0)	effect
nut -> NSS	0.02	-0.39	0.45	0.52	0.48	FALSE
mor -> NSS	0.22	-0.13	0.58	0.89	0.11	TRUE
PLAN -> NSS	-0.29	-0.78	0.20	0.12	0.88	TRUE
BIRD -> NSS	-0.08	-0.33	0.17	0.25	0.75	TRUE

```

betas_nestednessG <- betas_nestednessG[betas_nestednessG$effect == T, ]
betas_nestednessGEFFECT <- betas_nestednessG[grep('NSS', betas_nestednessG$var), ]
pars_nestednessG <-
  summary_nestednessG[c(grep('^beta', summary_nestednessG$variable),
                        grep('^alpha', summary_nestednessG$variable),
                        grep('^f', summary_nestednessG$variable),
                        grep('^sigma', summary_nestednessG$variable)), ]$variable
post_nestednessG <- mod_nestednessG$draws(pars_nestednessG, format = 'df')

post_nestednessG <-
  lapply(c('^beta', '^alpha', '^f', '^sigma'), FUN =
    function(x) {
      post_nestednessG[, grep(x, colnames(post_nestednessG))]
```

Warning: Dropping 'draws\_df' class as required metadata was removed.  
 Warning: Dropping 'draws\_df' class as required metadata was removed.  
 Warning: Dropping 'draws\_df' class as required metadata was removed.  
 Warning: Dropping 'draws\_df' class as required metadata was removed.

```

names(post_nestednessG) <- c('beta', 'alpha', 'f', 'sigma')
estimated_nestednessG_effect <-
  lapply(1, FUN =
    function(x) {

      f_ <- 'f_NSS['
      s <- grep('.*(NSS).*', colnames(post_nestednessG$alpha))
      est <- lapply(1:12, FUN =
        function(j) {
          f_ <- paste0(f_, j, ']')
          effect <-
            apply(post_nestednessG$alpha[, s], 1, mean) +
            post_nestednessG$beta$beta_temp_NSS * mean(dat$z_temperature) +
            post_nestednessG$beta$beta_rain_NSS * mean(dat$z_rainfall) +
            post_nestednessG$beta$beta_fruitAB_NSS * mean(dat$fruit_abun) +
            post_nestednessG$beta$beta_fruitS_NSS * mean(dat$fruit_S) +
            post_nestednessG$beta$beta_nut_NSS * mean(dat$HV_plants_nut) +
            post_nestednessG$beta$beta_mor_NSS * mean(dat$HV_plants_morfo) +
            post_nestednessG$beta$beta_PLAN_NSS * mean(dat$HV_plant) +
            post_nestednessG$beta$beta_BIRD_NSS * mean(dat$HV_bird)
          set.seed(123)
          effect <- sample(effect, 1e3)
```

```

    tibble(est = effect,
           month = j,
           class = 'mNSSu')

  })

est <- do.call('rbind', est)
est

})

estimated_nestednessG_effect <-
do.call('rbind', estimated_nestednessG_effect)

month_abbr <- na.omit(ymd(paste0("2023-", estimated_nestednessG_effect$month, "-01"),
                           label = TRUE, abbr = TRUE))

```

Warning: 2 failed to parse.

```
estimated_nestednessG_effect$month2 <- month_abbr
```

### S3.2.5 Plotting posterior distribution

```

estimated_nestednessG_effect |>
  group_by(month2) |>
  transmute(mu = mean(est),
            li = quantile(est, 0.025),
            ls = quantile(est, 0.975)) |>
  unique() |>
  ggplot() +
  geom_ribbon(aes(month2, ymin = li, ymax = ls), alpha = 0.5) +
  geom_line(aes(month2, mu)) +
  scale_x_date(date_labels = '%b', date_breaks = '2 month')

```

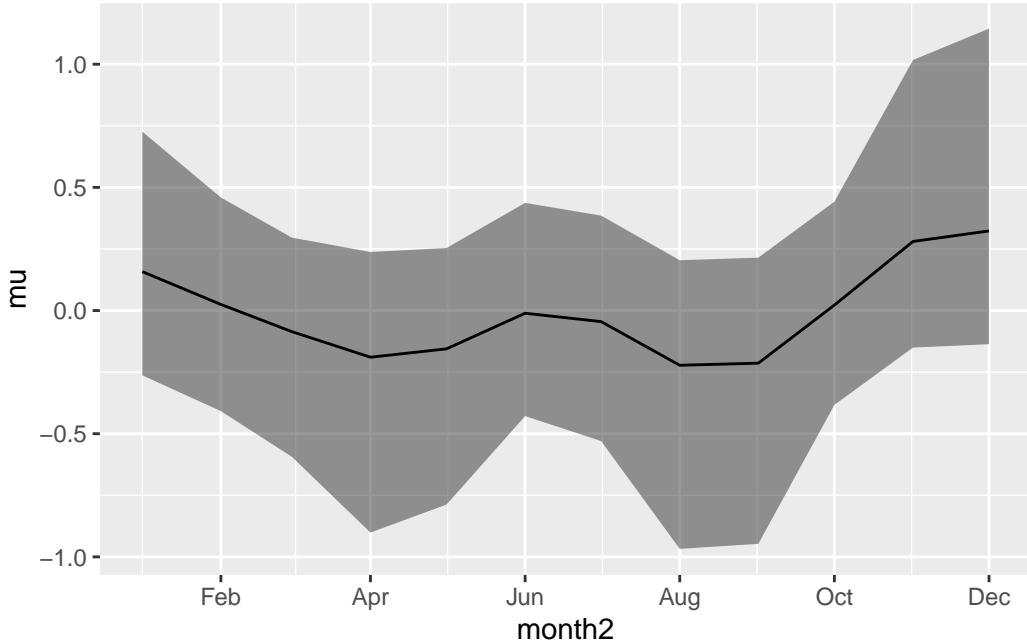


Figure S3: Monthly temporal pattern of nestedness in seed dispersal networks

### S3.3 Modularity

#### S3.3.1 Stan code

```
cat(file = 'Modularity_global.stan',
"
functions {
    vector GP_periodic(int period,
                        real gamma,           // periodicity
                        real sigma,          // smoothing term of the GP
                        real eta);           // noise parameter
    vector eta) {           // latent variable for each month

        int M = period;
        matrix[M, M] K;
        matrix[M, M] L_K;

        for (i in 1:(M - 1)) {
            for (j in (i+1):M) {
                real distance = abs(i - j);
                real periodic_distance = fmin(distance, period - distance);
                K[i, j] = sigma^2 * exp(-2 * square(sin(pi()*periodic_distance/period))/gamma^2);
                K[j, i] = K[i, j];    // filling the lower triangle
            }
            K[i, i] = sigma^2 + 1e-9;   // small values to guarantee stability
        }
        K[M, M] = sigma^2 + 1e-9;   // small values to guarantee stability
        return cholesky_decompose(K) * eta;
    }
}
```

```

        }

matrix GP_quadratic(matrix x,
                     real eta,
                     real rho,
                     real delta) {

    int N = dims(x)[1];
    matrix[N, N] K;
    matrix[N, N] L_K;

    for (i in 1:(N-1)) {
        K[i, i] = eta + delta;
        for (j in (i+1):N) {
            K[i, j] = square(eta) * exp(-rho * square(x[i, j]));
            K[j, i] = K[i, j];
        }
    }

    K[N, N] = eta + delta;
    L_K = cholesky_decompose(K);
    return L_K;
}

data{
    int N;
    int N_sites;
    int N_month;
    array[N] int network_size;
    vector[N] modularity;
    vector[N] nestedness;
    vector[N] H2;
    array[N] int month;
    array[N] int site;
    vector[N] HV_network;
    vector[N] HV_plant;
    vector[N] HV_plants_morfo;
    vector[N] HV_plants_nut;
    vector[N] HV_bird;
    vector[N] z_temperature;
    vector[N] z_rainfall;
    vector[N] fruit_S;
    vector[N] fruit_abun;
    matrix[N_sites, N_sites] dist_sites;
}

parameters {

    //////////////////////////////// Fruit abundance //////////////////////////////

    // GP periodic
    real<lower = 0> gamma_fruitS;
    real<lower = 0> sigma_f_fruitS;
    vector[N_month] eta_fruitS;
    // GP quadratic
    vector[N_sites] z_sites_fruitS;
    real<lower = 0> eta_site_fruitS;
    real<lower = 0> rho_site_fruitS;

    // Pars linear model
    real beta_temp_fruitS;
    real beta_rain_fruitS;
    // noise for likelihood function
    real<lower = 0> sigma_fruitS;
}

```

```

////////////////// Fruit abundance //////////////////
// GP periodic
real<lower = 0> gamma_fruitAB;
real<lower = 0> sigma_f_fruitAB;
vector[N_month] eta_fruitAB;
// GP quadratic
vector[N_sites] z_sites_fruitAB;
real<lower = 0> eta_site_fruitAB;
real<lower = 0> rho_site_fruitAB;

// Pars linear model

real beta_temp_fruitAB;
real beta_rain_fruitAB;
// noise for likelihood function
real<lower = 0> sigma_fruitAB;

////////////////// Fru. Nut. HV //////////////////
// GP periodic
real<lower = 0> gamma_nut_HV;
real<lower = 0> sigma_f_nut_HV;
vector[N_month] eta_nut_HV;
// GP quadratic
vector[N_sites] z_sites_nut_HV;
real<lower = 0> eta_site_nut_hv;
real<lower = 0> rho_site_nut_hv;

// Pars linear model

real beta_temp_nut_HV;
real beta_rain_nut_HV;
real beta_fruitAB_nut_HV;
real beta_fruitS_nut_HV;
real beta_BIRD_nut_HV;
// noise for likelihood function
real<lower = 0> sigma_nut_HV;

////////////////// Fru. morpho. HV //////////////////
// GP periodic
real<lower = 0> gamma_mor_HV;
real<lower = 0> sigma_f_mor_HV;
vector[N_month] eta_mor_HV;
// GP quadratic
vector[N_sites] z_sites_mor_HV;
real<lower = 0> eta_site_mor_hv;
real<lower = 0> rho_site_mor_hv;

// Pars linear model

real beta_temp_mor_HV;
real beta_rain_mor_HV;
real beta_fruitAB_mor_HV;
real beta_fruitS_mor_HV;
real beta_BIRD_mor_HV;
// noise for likelihood function
real<lower = 0> sigma_mor_HV;
real nu;

////////////////// Plants HV //////////////////

```

```

// GP periodic
real<lower = 0> gamma_PLAN_HV;
real<lower = 0> sigma_f_PLAN_HV;
vector[N_month] eta_PLAN_HV;
// GP quadratic
vector[N_sites] z_sites_PLAN_HV;
real<lower = 0> eta_site_PLAN_hv;
real<lower = 0> rho_site_PLAN_hv;

// Pars linear model
real beta_temp_PLAN_HV;
real beta_rain_PLAN_HV;
real beta_fruitAB_PLAN_HV;
real beta_fruitS_PLAN_HV;
real beta_nut_PLAN_HV;
real beta_mor_PLAN_HV;
real beta_BIRD_PLAN_HV;
// noise for likelihood function
real<lower = 0> sigma_PLAN_HV;

////////////////////////////// BirsdHV ///////////////////
// GP periodic
real<lower = 0> gamma_BIRD_HV;
real<lower = 0> sigma_f_BIRD_HV;
vector[N_month] eta_BIRD_HV;
// GP quadratic
vector[N_sites] z_sites_BIRD_HV;
real<lower = 0> eta_site_BIRD_hv;
real<lower = 0> rho_site_BIRD_hv;

// Pars linear model
real beta_temp_BIRD_HV;
real beta_rain_BIRD_HV;
real beta_fruitAB_BIRD_HV;
real beta_fruitS_BIRD_HV;
// noise for likelihood function
real<lower = 0> sigma_BIRD_HV;

////////////////// Network metrics ///////////////////
// Modularity ///////////////////
// GP periodic
real<lower = 0> gamma_MODU;
real<lower = 0> sigma_f_MODU;
vector[N_month] eta_MODU;
// GP quadratic
vector[N_sites] z_sites_MODU;
real<lower = 0> eta_site_MODU;
real<lower = 0> rho_site_MODU;

// Pars linear model
real beta_temp_MODU;
real beta_rain_MODU;
real beta_fruitAB_MODU;
real beta_fruitS_MODU;
real beta_nut_MODU;
real beta_mor_MODU;
real beta_PLAN_MODU;
real beta_BIRD_MODU;
// noise for likelihood function
real<lower = 0> sigma_MODU;

}

transformed parameters {

```

```

////////// Fruit abundance //////////
// GP periodic
vector[N_month] f_fruitAB;

f_fruitAB = GP_periodic(
    12,
    gamma_fruitAB,
    sigma_f_fruitAB,
    eta_fruitAB
);

// GP quadratic
vector[N_sites] alpha_fruitAB;
matrix[N_sites, N_sites] L_K_fruitAB;
L_K_fruitAB = GP_quadratic(dist_sites,
                            eta_site_fruitAB,
                            rho_site_fruitAB, 0.001);
alpha_fruitAB = L_K_fruitAB * z_sites_fruitAB;

////////// Fruit richness //////////
// GP periodic
vector[N_month] f_fruitS;

f_fruitS = GP_periodic(
    12,
    gamma_fruitS,
    sigma_f_fruitS,
    eta_fruitS
);

// GP quadratic
vector[N_sites] alpha_fruitS;
matrix[N_sites, N_sites] L_K_fruitS;
L_K_fruitS = GP_quadratic(dist_sites,
                           eta_site_fruitS,
                           rho_site_fruitS, 0.001);
alpha_fruitS = L_K_fruitS * z_sites_fruitS;

////////// Fru. Nut. HV //////////
// GP periodic
vector[N_month] f_nut_HV;

f_nut_HV = GP_periodic(
    12,
    gamma_nut_HV,
    sigma_f_nut_HV,
    eta_nut_HV
);

// GP quadratic
vector[N_sites] alpha_nut_HV;
matrix[N_sites, N_sites] L_K_nut_HV;
L_K_nut_HV = GP_quadratic(dist_sites,
                           eta_site_nut_hv,
                           rho_site_nut_hv, 0.001);
alpha_nut_HV = L_K_nut_HV * z_sites_nut_HV;

//   ////////// Fru. morpho. HV //////////
// GP periodic
vector[N_month] f_mor_HV;

```

```

f_mor_HV = GP_periodic(
    12,
    gamma_mor_HV,
    sigma_f_mor_HV,
    eta_mor_HV
);

// GP quadratic
vector[N_sites] alpha_mor_HV;
matrix[N_sites, N_sites] L_K_mor_HV;
L_K_mor_HV = GP_quadratic(dist_sites,
                           eta_site_mor_hv,
                           rho_site_mor_hv, 0.001);
alpha_mor_HV = L_K_mor_HV * z_sites_mor_HV;
// //////////////////////////////// Plants HV ///////////////////////////////
// GP periodic
vector[N_month] f_PLAN_HV;

f_PLAN_HV = GP_periodic(
    12,
    gamma_PLAN_HV,
    sigma_f_PLAN_HV,
    eta_PLAN_HV
);

// GP quadratic
vector[N_sites] alpha_PLAN_HV;
matrix[N_sites, N_sites] L_K_PLAN_HV;
L_K_PLAN_HV = GP_quadratic(dist_sites,
                           eta_site_PLAN_hv,
                           rho_site_PLAN_hv, 0.001);
alpha_PLAN_HV = L_K_PLAN_HV * z_sites_PLAN_HV;
////////////////////////////// Birdsds HV ///////////////////////////////
// GP periodic
vector[N_month] f_BIRD_HV;

f_BIRD_HV = GP_periodic(
    12,
    gamma_BIRD_HV,
    sigma_f_BIRD_HV,
    eta_BIRD_HV
);

// GP quadratic
vector[N_sites] alpha_BIRD_HV;
matrix[N_sites, N_sites] L_K_BIRD_HV;
L_K_BIRD_HV = GP_quadratic(dist_sites,
                           eta_site_BIRD_hv,
                           rho_site_BIRD_hv, 0.001);
alpha_BIRD_HV = L_K_BIRD_HV * z_sites_BIRD_HV;
// //////////////////////////////// Modularity ///////////////////////////////
// GP periodic
vector[N_month] f_MODU;

f_MODU = GP_periodic(
    12,
    gamma_MODU,
    sigma_f_MODU,
    eta_MODU);

// GP quadratic

```

```

vector[N_sites] alpha_MODU;
matrix[N_sites, N_sites] L_K_MODU;
L_K_MODU = GP_quadratic(dist_sites,
                         eta_site_MODU,
                         rho_site_MODU, 0.001);
alpha_MODU = L_K_MODU * z_sites_MODU;

}

model {

    //////////////////////////////////////////////////////////////////// Fruit abundance ///////////////////////////////////////////////////////////////////
    // priors for periodic GP
    eta_fruitAB ~ normal(0, 0.5);
    gamma_fruitAB ~ inv_gamma(5, 5);
    sigma_f_fruitAB ~ cauchy(0, 1);
    // GP quadratic
    z_sites_fruitAB ~ normal(0, 0.25);
    eta_site_fruitAB ~ exponential(4);
    rho_site_fruitAB ~ exponential(1);
    // Pars linear model
    beta_temp_fruitAB ~ normal(0, 0.5);
    beta_rain_fruitAB ~ normal(0, 0.5);
    // noise parameter of the likelihood
    sigma_fruitAB ~ exponential(1);

    //////////////////////////////////////////////////////////////////// Fruit richness ///////////////////////////////////////////////////////////////////
    // priors for periodic GP
    eta_fruits ~ normal(0, 0.5);
    gamma_fruits ~ inv_gamma(5, 5);
    sigma_f_fruits ~ cauchy(0, 1);
    // GP quadratic
    z_sites_fruits ~ normal(0, 0.25);
    eta_site_fruits ~ exponential(4);
    rho_site_fruits ~ exponential(1);
    // Pars linear model
    beta_temp_fruits ~ normal(0, 0.5);
    beta_rain_fruits ~ normal(0, 0.5);
    // noise parameter of the likelihood
    sigma_fruits ~ exponential(1);

    //////////////////////////////////////////////////////////////////// Fru. Nut. HV ///////////////////////////////////////////////////////////////////
    // priors for periodic GP
    eta_nut_HV ~ normal(0, 0.5);
    gamma_nut_HV ~ inv_gamma(5, 5);
    sigma_f_nut_HV ~ cauchy(0, 1);
    // GP quadratic
    z_sites_nut_HV ~ normal(0, 0.25);
    eta_site_nut_hv ~ exponential(4);
    rho_site_nut_hv ~ exponential(1);
    // Pars linear model
    beta_temp_nut_HV ~ normal(0, 0.5);
    beta_rain_nut_HV ~ normal(0, 0.5);
    beta_fruitAB_nut_HV ~ normal(0, 0.5);
    beta_fruitS_nut_HV ~ normal(0, 0.5);
    beta_BIRD_nut_HV ~ normal(0, 0.5);
    // noise parameter of the likelihood
    sigma_nut_HV ~ exponential(1);
    //

    //////////////////////////////////////////////////////////////////// Fru. morpho. HV ///////////////////////////////////////////////////////////////////
    // priors for periodic GP
    eta_mor_HV ~ normal(0, 0.5);
    gamma_mor_HV ~ inv_gamma(5, 5);
}

```

```

sigma_f_mor_HV ~ cauchy(0, 1);
// GP quadratic
z_sites_mor_HV ~ normal(0, 0.25);
eta_site_mor_hv ~ exponential(4);
rho_site_mor_hv ~ exponential(1);
// Pars linear model
beta_temp_mor_HV ~ normal(0, 0.5);
beta_rain_mor_HV ~ normal(0, 0.5);
beta_fruitAB_mor_HV ~ normal(0, 0.5);
beta_fruitS_mor_HV ~ normal(0, 0.5);
beta_BIRD_mor_HV ~ normal(0, 0.5);
// noise parameter of the likelihood
sigma_mor_HV ~ exponential(1);

//////////////////////////////////////////////////////////////// Plants HV ///////////////////////////////
// priors for periodic GP
eta_PLAN_HV ~ normal(0, 0.5);
gamma_PLAN_HV ~ inv_gamma(5, 5);
sigma_f_PLAN_HV ~ cauchy(0, 1);
// GP quadratic
z_sites_PLAN_HV ~ normal(0, 0.5);
eta_site_PLAN_hv ~ exponential(4);
rho_site_PLAN_hv ~ exponential(1);
// Pars linear model
beta_temp_PLAN_HV ~ normal(0, 0.5);
beta_rain_PLAN_HV ~ normal(0, 0.5);
beta_fruitAB_PLAN_HV ~ normal(0, 0.5);
beta_fruitS_PLAN_HV ~ normal(0, 0.5);
beta_nut_PLAN_HV ~ normal(0, 0.5);
beta_mor_PLAN_HV ~ normal(0, 0.5);
beta_BIRD_PLAN_HV ~ normal(0, 0.5);
// noise parameter of the likelihood
sigma_PLAN_HV ~ exponential(1);

//////////////////////////////////////////////////////////////// Birds HV ///////////////////////////////
// priors for periodic GP
eta_BIRD_HV ~ normal(0, 0.5);
gamma_BIRD_HV ~ inv_gamma(5, 5);
sigma_f_BIRD_HV ~ cauchy(0, 1);
// GP quadratic
z_sites_BIRD_HV ~ normal(0, 0.5);
eta_site_BIRD_hv ~ exponential(4);
rho_site_BIRD_hv ~ exponential(1);
// Pars linear model
beta_temp_BIRD_HV ~ normal(0, 0.5);
beta_rain_BIRD_HV ~ normal(0, 0.5);
beta_fruitAB_BIRD_HV ~ normal(0, 0.5);
beta_fruitS_BIRD_HV ~ normal(0, 0.5);
// noise parameter of the likelihood
sigma_BIRD_HV ~ exponential(1);

//////////////////////////////////////////////////////////////// Modularity ///////////////////////////////
vector[N] mu_MODU;
vector[N] p1_MODU;
vector[N] p2_MODU;
// priors for periodic GP
eta_MODU ~ normal(0, 1);
gamma_MODU ~ inv_gamma(5, 5);
sigma_f_MODU ~ cauchy(0, 1);
// GP quadratic
z_sites_MODU ~ normal(0, 0.25);
eta_site_MODU ~ exponential(4);
rho_site_MODU ~ exponential(1);
// Pars linear model

```

```

beta_fruitAB_MODU ~ normal(0, 0.055);
beta_fruitS_MODU ~ normal(0, 0.055);
beta_nut_MODU ~ normal(0, 0.055);
beta_mor_MODU ~ normal(0, 0.055);
beta_PLAN_MODU ~ normal(0, 0.055);
beta_BIRD_MODU ~ normal(0, 0.055);
// beta_NW_MODU ~ normal(0, 0.055);
// noise parameter of the likelihood
sigma_MODU ~ exponential(2);

//////////////////////////// fruit abundance /////////////////////
//////////////////////////// for (i in 1:N) {
    fruit_abun[i] ~ student_t(7, alpha_fruitAB[site[i]] + f_fruitAB[month[i]] +
        beta_temp_fruitAB * z_temperature[i] +
        beta_rain_fruitAB * z_rainfall[i],
        sigma_fruitAB);
}

//////////////////////////// fruit richness ///////////////////
//////////////////////////// for (i in 1:N) {
    fruit_S[i] ~ student_t(7, alpha_fruitS[site[i]] + f_fruitS[month[i]] +
        beta_temp_fruitS * z_temperature[i] +
        beta_rain_fruitS * z_rainfall[i],
        sigma_fruitS);
}

//////////////////////////// Fru. Nut. HV ///////////////////
//////////////////////////// for (i in 1:N) {
    HV_plants_nut[i] ~ student_t(7, alpha_nut_HV[site[i]] + f_nut_HV[month[i]] +
        beta_temp_nut_HV * z_temperature[i] +
        beta_rain_nut_HV * z_rainfall[i] +
        beta_BIRD_nut_HV * HV_bird[i] +
        beta_fruitAB_nut_HV * fruit_abun[i] +
        beta_fruitS_nut_HV * fruit_S[i],
        sigma_nut_HV);
}

// /////////////////// Fru. morpho. HV ///////////////////
// /////////////////// for (i in 1:N) {
    HV_plants_morfo[i] ~ student_t(15, alpha_mor_HV[site[i]] + f_mor_HV[month[i]] +
        beta_temp_mor_HV * z_temperature[i] +
        beta_rain_mor_HV * z_rainfall[i] +
        beta_fruitAB_mor_HV * fruit_abun[i] +
        beta_fruitS_mor_HV * fruit_S[i] +
        beta_BIRD_mor_HV * HV_bird[i],
        sigma_mor_HV);
}

//////////////////////////// Plants HV ///////////////////
//////////////////////////// for (i in 1:N) {
    HV_plant[i] ~ student_t(7, alpha_PLAN_HV[site[i]] + f_PLAN_HV[month[i]] +
        beta_temp_PLAN_HV * z_temperature[i] +
        beta_rain_PLAN_HV * z_rainfall[i] +
        beta_fruitAB_PLAN_HV * fruit_abun[i] +
        beta_fruitS_PLAN_HV * fruit_S[i] +
        beta_nut_PLAN_HV * HV_plants_nut[i] +
        beta_mor_PLAN_HV * HV_plants_morfo[i] +
        beta_BIRD_PLAN_HV * HV_bird[i],
        sigma_PLAN_HV);
}

//////////////////////////// Birds HV ///////////////////

```

```

////////// HV_bird[i] ~ student_t(7, alpha_BIRD_HV[site[i]] + f_BIRD_HV[month[i]] +
for (i in 1:N) {
    HV_bird[i] ~ student_t(7, alpha_BIRD_HV[site[i]] + f_BIRD_HV[month[i]] +
        beta_temp_BIRD_HV * z_temperature[i] +
        beta_rain_BIRD_HV * z_rainfall[i] +
        beta_fruitAB_BIRD_HV * fruit_abun[i] +
        beta_fruitS_BIRD_HV * fruit_S[i],
        sigma_BIRD_HV);
}
// /////////////// Modularity ///////////////
// /////////////// for (i in 1:N) {
mu_MODU[i] = inv_logit(alpha_MODU[site[i]] + f_MODU[month[i]] +
    beta_temp_MODU * z_temperature[i] +
    beta_rain_MODU * z_rainfall[i] +
    beta_fruitAB_MODU * fruit_abun[i] +
    beta_fruitS_MODU * fruit_S[i] +
    beta_nut_MODU * HV_plants_nut[i] +
    beta_mor_MODU * HV_plants_morfo[i] +
    beta_PLAN_MODU * HV_plant[i] +
    beta_BIRD_MODU * HV_bird[i]);
}
p1_MODU = mu_MODU * sigma_MODU;
p2_MODU = (1 - mu_MODU) * sigma_MODU;
modularity ~ beta(p1_MODU, p2_MODU);
}

generated quantities {
    /////////////// Fruit abundance ///////////////
    array[N] real ppcheck_fruitAB;
    vector[N] mu_fruitAB;

    for (i in 1:N) {
        mu_fruitAB[i] = alpha_fruitAB[site[i]] + f_fruitAB[month[i]] +
            beta_temp_fruitAB * z_temperature[i] +
            beta_rain_fruitAB * z_rainfall[i];
    }
    ppcheck_fruitAB = student_t_rng(7, mu_fruitAB, sigma_fruitAB);

    /////////////// Fruit richness ///////////////
    array[N] real ppcheck_fruitS;
    vector[N] mu_fruitS;

    for (i in 1:N) {
        mu_fruitS[i] = alpha_fruitS[site[i]] + f_fruitS[month[i]] +
            beta_temp_fruitS * z_temperature[i] +
            beta_rain_fruitS * z_rainfall[i];
    }
    ppcheck_fruitS = student_t_rng(7, mu_fruitS, sigma_fruitS);

    /////////////// Fru. Nut. HV ///////////////
    array[N] real ppcheck_nut_HV;
    vector[N] mu_nut_HV;

    for (i in 1:N) {
        mu_nut_HV[i] = alpha_nut_HV[site[i]] + f_nut_HV[month[i]] +

```

```

        beta_temp_nut_HV * z_temperature[i] +
        beta_rain_nut_HV * z_rainfall[i] +
        beta_fruitAB_nut_HV * fruit_abun[i] +
        beta_fruitS_nut_HV * fruit_S[i] +
        beta_BIRD_nut_HV * HV_bird[i];
    }

ppcheck_nut_HV = student_t_rng(7, mu_nut_HV, sigma_nut_HV);
////////////////////////////////////////////////////////////////// Fru. morpho. HV //////////////////////////////////////////////////////////////////
array[N] real ppcheck_mor_HV;
vector[N] mu_mor_HV;

for (i in 1:N) {
    mu_mor_HV[i] = alpha_mor_HV[site[i]] + f_mor_HV[month[i]] +
        beta_temp_mor_HV * z_temperature[i] +
        beta_rain_mor_HV * z_rainfall[i] +
        beta_fruitAB_mor_HV * fruit_abun[i] +
        beta_fruitS_mor_HV * fruit_S[i] +
        beta_BIRD_mor_HV * HV_bird[i];
}

ppcheck_mor_HV = student_t_rng(15, mu_mor_HV, sigma_mor_HV);

////////////////////////////////////////////////////////////////// Plants HV //////////////////////////////////////////////////////////////////
array[N] real ppcheck_PLAN_HV;
vector[N] mu_PLAN_HV;

for (i in 1:N) {
    mu_PLAN_HV[i] = alpha_PLAN_HV[site[i]] + f_PLAN_HV[month[i]] +
        beta_temp_PLAN_HV * z_temperature[i] +
        beta_rain_PLAN_HV * z_rainfall[i] +
        beta_fruitAB_PLAN_HV * fruit_abun[i] +
        beta_fruitS_PLAN_HV * fruit_S[i] +
        beta_nut_PLAN_HV * HV_plants_nut[i] +
        beta_mor_PLAN_HV * HV_plants_morfo[i] +
        beta_BIRD_PLAN_HV * HV_bird[i];
}

ppcheck_PLAN_HV = student_t_rng(7, mu_PLAN_HV, sigma_PLAN_HV);

////////////////////////////////////////////////////////////////// Birds HV //////////////////////////////////////////////////////////////////
array[N] real ppcheck_BIRD_HV;
vector[N] mu_BIRD_HV;

for (i in 1:N) {
    mu_BIRD_HV[i] = alpha_BIRD_HV[site[i]] + f_BIRD_HV[month[i]] +
        beta_temp_BIRD_HV * z_temperature[i] +
        beta_rain_BIRD_HV * z_rainfall[i] +
        beta_fruitAB_BIRD_HV * fruit_abun[i] +
        beta_fruitS_BIRD_HV * fruit_S[i];
}

ppcheck_BIRD_HV = student_t_rng(7, mu_BIRD_HV, sigma_BIRD_HV);

////////////////////////////////////////////////////////////////// Modularity //////////////////////////////////////////////////////////////////
array[N] real ppcheck_MODU;
vector[N] mu_MODU;
vector[N] p1_MODU;
vector[N] p2_MODU;
real variance_fixed;
real variance_GP_temp;

```

```

real variance_GP_space;
real variance_total;
real prop_GP_temp;
real prop_GP_space;
real prop_fixed;

for (i in 1:N) {
  mu_MODU[i] = alpha_MODU[site[i]] + f_MODU[month[i]] +
    beta_temp_MODU * z_temperature[i] +
    beta_rain_MODU * z_rainfall[i] +
    beta_fruitAB_MODU * fruit_abun[i] +
    beta_fruitsS_MODU * fruit_S[i] +
    beta_nut_MODU * HV_plants_nut[i] +
    beta_mor_MODU * HV_plants_morfo[i] +
    beta_PLAN_MODU * HV_plant[i] +
    beta_BIRD_MODU * HV_bird[i];
}

// variance decomposition
variance_fixed = variance(mu_MODU);
variance_GP_space = eta_site_MODU^2;
variance_GP_temp = sigma_f_MODU^2;
variance_total = variance_fixed + variance_GP_temp +
  variance_GP_space + ((pi()^2)/3);
prop_GP_temp = variance_GP_temp / variance_total;
prop_GP_space = variance_GP_space / variance_total;
prop_fixed = variance_fixed / variance_total;

mu_MODU = inv_logit(mu_MODU);
p1_MODU = mu_MODU * sigma_MODU;
p2_MODU = (1 - mu_MODU) * sigma_MODU;
ppcheck_MODU = beta_rng(p1_MODU, p2_MODU);

}
"
)

```

### S3.3.2 Fitting the model

```

file <- paste0(getwd(), '/Modularity_global.stan')
fit_MODU_G <- cmdstan_model(file, compile = T)

mod_MODU_G <-
  fit_MODU_G$sample(
    data = dat,
    iter_sampling = 2e3,
    iter_warmup = 500,
    thin = 3,
    chains = 3,
    parallel_chains = 3,
    seed = 5
  )

```

### S3.3.3 Sampling diagnostics

```

summary_MODU_G <- mod_MODU_G$summary()
na.omit(summary_MODU_G[summary_MODU_G$ess_bulk < 500,]) |> print(n = 1e3)

# A tibble: 0 x 10
# i 10 variables: variable <chr>, mean <dbl>, median <dbl>, sd <dbl>,
#   mad <dbl>, q5 <dbl>, q95 <dbl>, rhat <dbl>, ess_bulk <dbl>, ess_tail <dbl>

par(mfrow = c(3, 3), mar = c(4, 4, 2, 1))
mod_diagnostics(mod_MODU_G,
                 summary_MODU_G[grep('fruitAB', summary_MODU_G$variable), ])
title('Fruit abundance')
mod_diagnostics(mod_MODU_G,
                 summary_MODU_G[grep('fruitsS', summary_MODU_G$variable), ])
title('Fruit richness')
mod_diagnostics(mod_MODU_G,
                 summary_MODU_G[grep('nut_HV', summary_MODU_G$variable), ])
title('Frut. Nut. HV')
mod_diagnostics(mod_MODU_G,
                 summary_MODU_G[grep('mor_HV', summary_MODU_G$variable), ])
title('Frut. Morpho. HV')
mod_diagnostics(mod_MODU_G,
                 summary_MODU_G[grep('PLAN_HV', summary_MODU_G$variable), ])
title('Plants HV')
mod_diagnostics(mod_MODU_G,
                 summary_MODU_G[grep('BIRD_HV', summary_MODU_G$variable), ])
title('Birds HV')

mod_diagnostics(mod_MODU_G,
                 summary_MODU_G[grep('MODU', summary_MODU_G$variable), ])
title('Modularity')
par(mfrow = c(1, 1))

```

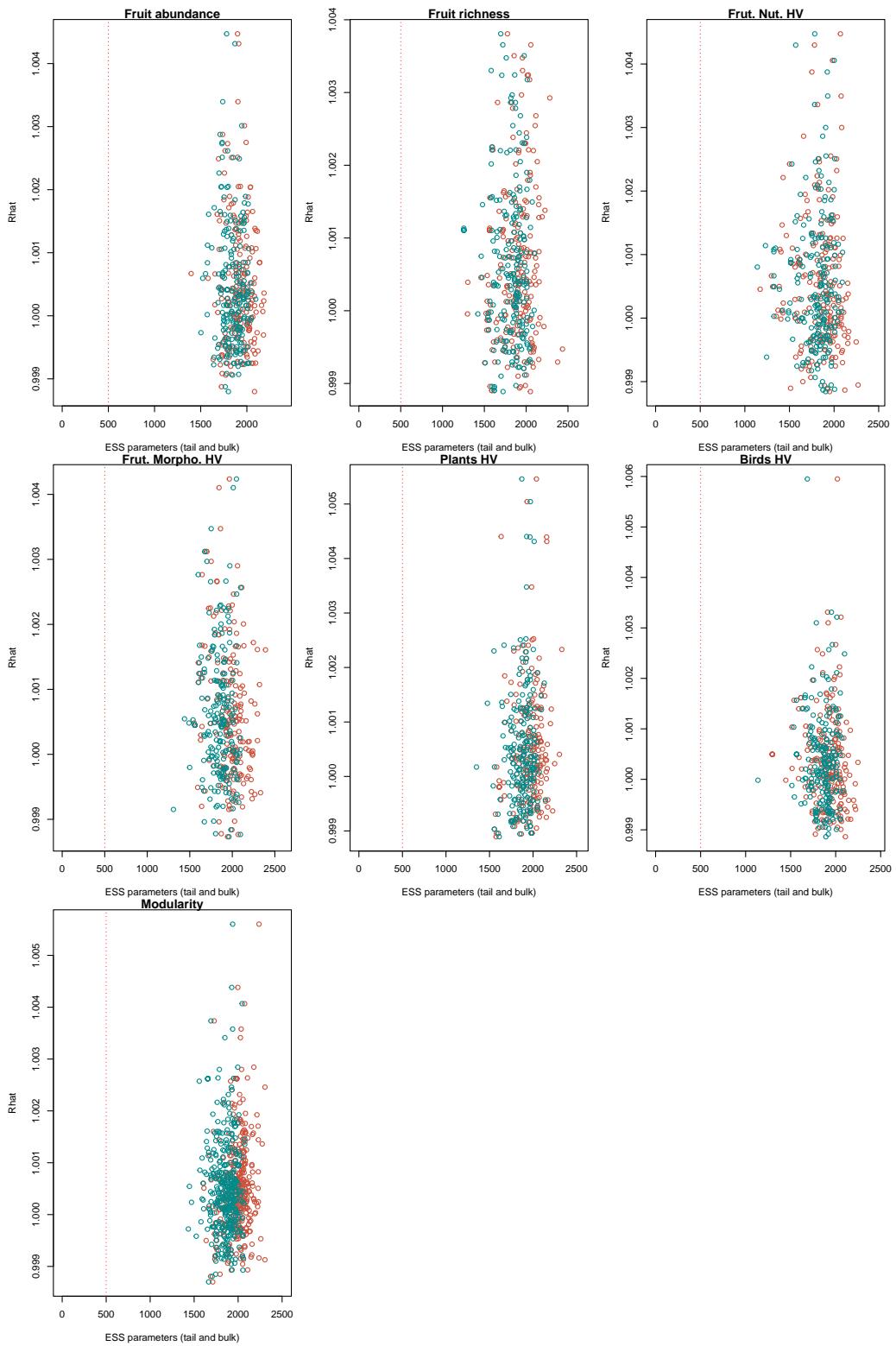


Figure S4: Rhat values vs effective sampling size (ess). Rhat  $< 1.05$  indicates that the Markov Chains converged to the same stationary distribution. ess must be at least 100 per chain in order to be reliable

```

par(mfrow = c(3, 3), mar = c(4, 4, 1, 1))
ppcheck_MODU_G <- mod_MODU_G$draws('ppcheck_fruitAB', format = 'matrix')
plot(density(dat$fruit_abun), main = '', xlab = 'Fruit abundance',
      ylim = c(0, 1))
for (i in 1:100) lines(density(ppcheck_MODU_G[i, ]), lwd = 0.1)
lines(density(dat$fruit_abun), col = 'red', lwd = 2)

ppcheck_MODU_G <- mod_MODU_G$draws('ppcheck_fruitS', format = 'matrix')
plot(density(dat$fruit_S), main = '', xlab = 'Fruit richness',
      ylim = c(0, 1))
for (i in 1:100) lines(density(ppcheck_MODU_G[i, ]), lwd = 0.1)
lines(density(dat$fruit_S), col = 'red', lwd = 2)

ppcheck_MODU_G <- mod_MODU_G$draws('ppcheck_nut_HV', format = 'matrix')
plot(density(dat$HV_plants_nut), main = '', xlab = 'Frut. Nut. HV',
      ylim = c(0, 1))
for (i in 1:100) lines(density(ppcheck_MODU_G[i, ]), lwd = 0.1)
lines(density(dat$HV_plants_nut), col = 'red', lwd = 2)

ppcheck_MODU_G <- mod_MODU_G$draws('ppcheck_mor_HV', format = 'matrix')
plot(density(dat$HV_plants_morfo), main = '', xlab = 'Frut. Morpho. HV',
      ylim = c(0, 1))
for (i in 1:100) lines(density(ppcheck_MODU_G[i, ]), lwd = 0.1)
lines(density(dat$HV_plants_morfo), col = 'red', lwd = 2)

ppcheck_MODU_G <- mod_MODU_G$draws('ppcheck_PLAN_HV', format = 'matrix')
plot(density(dat$HV_plant), main = '', xlab = 'Plants HV',
      ylim = c(0, 1))
for (i in 1:100) lines(density(ppcheck_MODU_G[i, ]), lwd = 0.1)
lines(density(dat$HV_plant), col = 'red', lwd = 2)

ppcheck_MODU_G <- mod_MODU_G$draws('ppcheck_BIRD_HV', format = 'matrix')
plot(density(dat$HV_bird), main = '', xlab = 'Birds HV',
      ylim = c(0, 1))
for (i in 1:100) lines(density(ppcheck_MODU_G[i, ]), lwd = 0.1)
lines(density(dat$HV_bird), col = 'red', lwd = 2)

ppcheck_MODU_G <- mod_MODU_G$draws('ppcheck_MODU', format = 'matrix')
plot(density(dat$modularity), main = '', xlab = 'Modularity',
      ylim = c(0, 5))
for (i in 1:100) lines(density(ppcheck_MODU_G[i, ]), lwd = 0.1)
lines(density(dat$modularity), col = 'red', lwd = 2)
par(mfrow = c(1, 1))

```

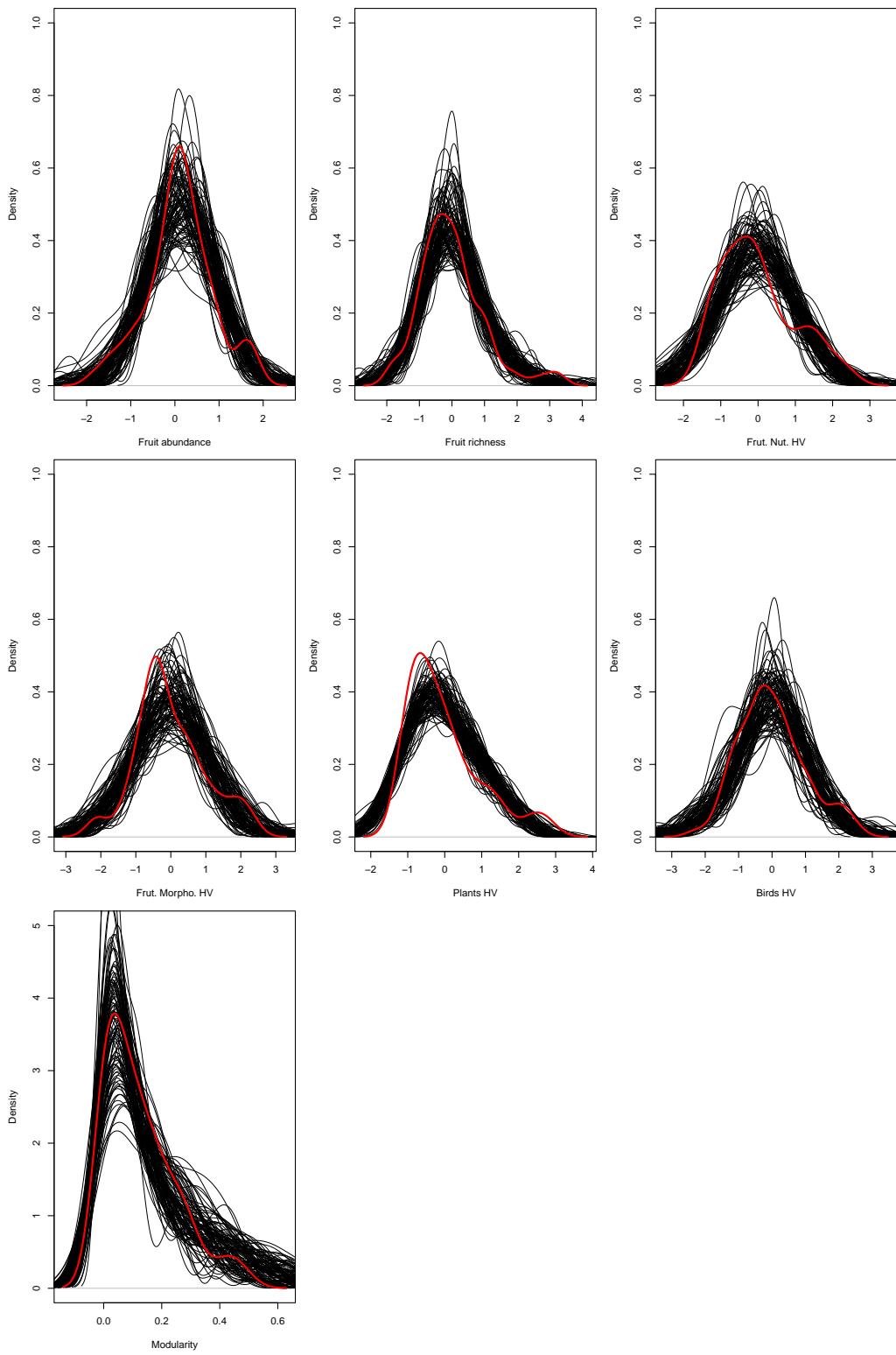


Figure S5: Posterior predictive checks of the of variables integrating the generative model explaining the modularity of seed seed dispersal networks

### S3.3.4 Extracting posterior draws

```

pars_MODU_G <- summary_MODU_G[grep('^beta', summary_MODU_G$variable), ]$variable
betas_MODU_G <- mod_MODU_G$draws(pars_MODU_G, format = 'df')
betas_MODU_G <- betas_MODU_G[, -c((ncol(betas_MODU_G)-2):ncol(betas_MODU_G))]

```

Warning: Dropping 'draws\_df' class as required metadata was removed.

```

betas_MODU_G <-
  lapply(1:ncol(betas_MODU_G),
    function(i) {
      x <- betas_MODU_G[[i]]
      var <- colnames(betas_MODU_G)[[i]]

      df <-
        tibble(var = var,
          mu = mean(x),
          li = quantile(x, 0.025),
          ls = quantile(x, 0.975),
          p_pos = mean(x > 0),
          p_neg = mean(x < 0))

      df$effect <- df$p_pos > 0.7 | df$p_neg > 0.7
      df
    })

betas_MODU_G <- do.call('rbind', betas_MODU_G)

betas_MODU_G_LABS <- betas_MODU_G[c(grep('PLAN_MODU', betas_MODU_G$var),
                                      grep('mor_MODU', betas_MODU_G$var)), ]
betas_MODU_GEFFECT <- betas_MODU_G[grep('MODU', betas_MODU_G$var), ]

```

```

betas_MODU_G_z <- betas_MODU_G

betas_MODU_G_z$var <-
  gsub('^(beta_)([A-Za-z]*)(_)(.*)$', '\\\\2 -> \\\\4', betas_MODU_G$var)

colnames(betas_MODU_G_z) <-
  c('Directional effect', 'Mean', '0.025%', '0.975%', 'P(effect > 0)',
    'P(effect < 0)', 'effect')

knitr::kable(betas_MODU_G_z,
  digits = 2,
  caption = 'Average values, 95% CIs, and probabilities of directional effects for the relationships'

```

Table S2: Average values, 95% CIs, and probabilities of directional effects for the relationships assessed in the generative model. An effect is marked as TRUE if  $P(\text{effect} > 0) > 0.7$  or  $P(\text{effect} < 0) > 0.7$  or.

Directional effect	Mean	0.025%	0.975%	$P(\text{effect} > 0)$	$P(\text{effect} < 0)$	effect
temp -> fruitS	-0.16	-0.43	0.11	0.13	0.87	TRUE
rain -> fruitS	0.46	0.18	0.73	1.00	0.00	TRUE
temp -> fruitAB	0.00	-0.24	0.25	0.50	0.50	FALSE
rain -> fruitAB	-0.10	-0.31	0.11	0.18	0.82	TRUE
temp -> nut_HV	0.36	0.05	0.67	0.99	0.01	TRUE
rain -> nut_HV	-0.02	-0.32	0.29	0.45	0.55	FALSE
fruitAB -> nut_HV	0.22	-0.16	0.61	0.87	0.13	TRUE
fruitS -> nut_HV	-0.03	-0.33	0.29	0.42	0.58	FALSE
BIRD -> nut_HV	0.02	-0.16	0.22	0.61	0.39	FALSE
temp -> mor_HV	0.03	-0.25	0.32	0.60	0.40	FALSE
rain -> mor_HV	0.06	-0.29	0.38	0.64	0.36	FALSE
fruitAB -> mor_HV	0.20	-0.17	0.57	0.85	0.15	TRUE
fruitS -> mor_HV	-0.01	-0.33	0.30	0.47	0.53	FALSE
BIRD -> mor_HV	0.04	-0.22	0.29	0.63	0.37	FALSE
temp -> PLAN_HV	-0.01	-0.14	0.11	0.41	0.59	FALSE
rain -> PLAN_HV	0.07	-0.07	0.20	0.86	0.14	TRUE
fruitAB -> PLAN_HV	-0.14	-0.29	0.01	0.03	0.97	TRUE
fruitS -> PLAN_HV	0.07	-0.06	0.20	0.88	0.12	TRUE
nut -> PLAN_HV	0.59	0.48	0.70	1.00	0.00	TRUE
mor -> PLAN_HV	0.47	0.38	0.56	1.00	0.00	TRUE
BIRD -> PLAN_HV	0.03	-0.06	0.12	0.74	0.26	TRUE
temp -> BIRD_HV	0.20	-0.12	0.50	0.90	0.10	TRUE
rain -> BIRD_HV	-0.15	-0.48	0.19	0.19	0.81	TRUE
fruitAB -> BIRD_HV	0.24	-0.12	0.60	0.91	0.09	TRUE
fruitS -> BIRD_HV	-0.05	-0.33	0.25	0.39	0.61	FALSE
temp -> MODU	-0.05	-0.34	0.27	0.38	0.62	FALSE
rain -> MODU	0.16	-0.10	0.41	0.89	0.11	TRUE
fruitAB -> MODU	0.02	-0.08	0.12	0.68	0.32	FALSE
fruitS -> MODU	0.01	-0.10	0.10	0.55	0.45	FALSE

Directional effect	Mean	0.025%	0.975%	P(effect > 0)	P(effect < 0)	effect
nut -> MODU	0.04	-0.05	0.14	0.79	0.21	TRUE
mor -> MODU	0.01	-0.08	0.11	0.60	0.40	FALSE
PLAN -> MODU	0.04	-0.06	0.14	0.77	0.23	TRUE
BIRD -> MODU	0.00	-0.09	0.10	0.50	0.50	FALSE

```

betas_MODU_G <- betas_MODU_G[betas_MODU_G$effect == T, ]

pars_MODU_G <-
  summary_MODU_G[c(grep('`beta', summary_MODU_G$variable),
                    grep('`alpha', summary_MODU_G$variable),
                    grep('`f', summary_MODU_G$variable),
                    grep('`sigma', summary_MODU_G$variable)), ]$variable

post_MODU_G <- mod_MODU_G$draws(pars_MODU_G, format = 'df')

post_MODU_G <-
  lapply(c(`^beta`, `^alpha`, `^f`, `^sigma`), FUN =
    function(x) {
      post_MODU_G[, grep(x, colnames(post_MODU_G))]
```

Warning: Dropping 'draws\_df' class as required metadata was removed.  
 Warning: Dropping 'draws\_df' class as required metadata was removed.  
 Warning: Dropping 'draws\_df' class as required metadata was removed.  
 Warning: Dropping 'draws\_df' class as required metadata was removed.

```

names(post_MODU_G) <- c('beta', 'alpha', 'f', 'sigma')

post_MODU_G$f`f_fruitAB[1]`
```

```

[1] 0.568746000 0.199923000 0.360340000 0.539783000 0.529446000
[6] 0.405509000 0.481006000 0.624027000 0.361598000 0.417692000
[11] 0.631961000 0.770167000 0.429592000 0.426298000 0.424877000
[16] 0.350454000 0.282243000 0.548235000 0.337623000 0.482122000
[21] 0.656151000 0.354013000 0.235524000 0.552587000 0.345426000
[26] 0.151315000 0.416769000 0.426581000 0.215653000 0.388295000
[31] 0.568302000 0.578274000 0.124070000 0.543354000 0.622024000
[36] 0.530570000 0.379208000 0.328924000 0.502011000 0.465817000
[41] 0.337172000 0.242992000 0.176691000 0.571236000 0.552644000
[46] 0.227828000 0.457426000 0.627272000 0.540161000 0.385340000
[51] 0.551496000 0.509665000 0.446020000 0.461944000 0.521612000
[56] 0.301795000 0.577025000 0.562935000 0.516664000 0.638875000
[61] 0.033672000 0.359616000 0.457656000 0.453744000 0.457425000
[66] 0.478896000 0.396656000 0.635097000 0.305243000 0.515136000
[71] 0.632079000 0.471878000 0.303108000 0.624335000 -0.127278000
```

[76]	0.065229800	0.113123000	0.272025000	0.516666000	0.477657000
[81]	0.057000100	0.255203000	0.418538000	0.559067000	0.365411000
[86]	0.386162000	0.280337000	0.369654000	0.549369000	0.438829000
[91]	0.394512000	0.428869000	0.488661000	0.629154000	0.310567000
[96]	0.657083000	0.659276000	0.288682000	0.600845000	0.096111300
[101]	0.619758000	0.442277000	0.331142000	0.377859000	0.411842000
[106]	0.471729000	0.498098000	0.117986000	0.209589000	0.466719000
[111]	0.550387000	0.457335000	0.326928000	0.354004000	0.647940000
[116]	0.569989000	0.439840000	0.139938000	0.390295000	0.545970000
[121]	0.491564000	0.359198000	0.705109000	0.431284000	0.418153000
[126]	0.139809000	0.526335000	0.514827000	0.543826000	0.488100000
[131]	0.273081000	0.717278000	0.639676000	0.506665000	0.536904000
[136]	0.580704000	0.676129000	0.748942000	0.354810000	0.328640000
[141]	0.314168000	0.696747000	0.173229000	0.006508000	0.041792800
[146]	0.346635000	0.380629000	0.480430000	0.257144000	0.512762000
[151]	0.568526000	0.550398000	0.269981000	0.697229000	0.344119000
[156]	0.299541000	0.422337000	0.648469000	0.284932000	0.414293000
[161]	0.240889000	0.635570000	0.256201000	0.249783000	0.684080000
[166]	0.471204000	0.272960000	0.330658000	0.574645000	0.602331000
[171]	0.637487000	0.471369000	0.480845000	0.488566000	0.173783000
[176]	0.632347000	0.238111000	0.632070000	0.476707000	0.253768000
[181]	0.043736300	0.808983000	0.360527000	0.340040000	0.296360000
[186]	0.273660000	0.214275000	0.716353000	0.686652000	0.451066000
[191]	0.348654000	0.294202000	0.610399000	0.540477000	0.399813000
[196]	0.176818000	0.468971000	0.459367000	0.326807000	0.353294000
[201]	0.482053000	0.496283000	0.253990000	0.585131000	0.220402000
[206]	0.528209000	0.089693900	0.639155000	0.374787000	0.587338000
[211]	0.479013000	0.168848000	0.447111000	0.499349000	0.403220000
[216]	0.648789000	0.657121000	0.440090000	0.396531000	0.477093000
[221]	0.424465000	0.482673000	0.346412000	0.456815000	0.586620000
[226]	0.317878000	0.358675000	0.266498000	0.347870000	0.291462000
[231]	0.211716000	0.377889000	0.192891000	0.324182000	0.619078000
[236]	0.536207000	0.604671000	0.152295000	0.216211000	0.641916000
[241]	0.255902000	0.226817000	0.610524000	0.393963000	0.032116000
[246]	0.466261000	0.376868000	0.484065000	0.293276000	0.419176000
[251]	0.328241000	0.453421000	0.409832000	0.379975000	0.781435000
[256]	0.178575000	0.366363000	0.270307000	0.107077000	0.354692000
[261]	0.836498000	0.678909000	-0.046551600	0.183077000	0.331330000
[266]	0.231180000	0.356831000	0.415327000	0.315795000	0.344518000
[271]	0.670547000	0.116564000	0.323958000	0.408449000	0.232385000
[276]	0.498531000	0.846805000	0.452970000	0.348680000	0.375179000
[281]	0.082532000	0.619929000	0.502976000	0.431419000	0.193046000
[286]	0.297480000	0.365788000	0.585788000	0.267729000	0.185448000

[291]	0.147157000	0.501519000	0.772642000	0.639325000	0.513030000
[296]	0.409094000	0.490268000	0.363477000	0.232267000	0.662271000
[301]	0.509299000	0.397230000	0.445725000	0.221530000	0.384754000
[306]	0.306921000	0.361963000	0.284239000	0.197341000	0.385171000
[311]	0.393915000	0.759240000	0.214012000	0.412076000	0.593816000
[316]	0.325219000	0.532248000	0.318257000	0.631292000	0.327862000
[321]	0.368384000	0.550514000	0.582776000	0.349497000	0.288905000
[326]	0.457205000	0.423629000	0.443682000	0.311459000	0.299916000
[331]	0.270343000	0.706831000	0.271514000	0.262220000	0.521772000
[336]	0.355996000	0.570424000	0.561183000	0.052254500	0.426152000
[341]	0.523627000	0.471902000	0.411813000	0.558755000	0.425598000
[346]	0.186592000	0.452927000	0.422171000	0.013311500	0.398265000
[351]	0.545076000	0.224566000	0.238079000	0.339131000	0.376414000
[356]	0.540067000	0.647403000	0.456199000	0.271396000	0.364887000
[361]	0.470093000	0.514044000	0.441590000	0.325003000	0.312031000
[366]	0.247287000	0.486292000	0.515934000	0.552993000	0.549956000
[371]	0.407024000	0.482069000	0.299575000	0.340349000	0.214816000
[376]	0.591481000	0.589131000	0.408931000	0.389146000	0.522003000
[381]	0.808420000	0.291346000	0.320506000	0.591332000	0.301839000
[386]	0.361913000	0.548641000	0.398371000	0.368085000	0.422965000
[391]	0.461537000	0.401650000	0.375851000	0.650592000	0.568874000
[396]	0.655572000	0.631297000	0.471255000	0.533938000	0.363481000
[401]	0.498635000	0.457590000	0.090473100	0.348854000	0.204503000
[406]	0.456860000	0.843585000	0.724298000	0.310270000	0.420235000
[411]	0.505773000	0.417753000	0.204330000	0.384342000	0.622569000
[416]	-0.047831100	0.331687000	0.468882000	0.381586000	0.122649000
[421]	0.865881000	0.577328000	0.500962000	0.310256000	0.544752000
[426]	0.415831000	0.572239000	0.344326000	0.496445000	0.478932000
[431]	0.339858000	0.677719000	0.257096000	0.690746000	0.463939000
[436]	0.405640000	0.576834000	0.128328000	0.521415000	0.470075000
[441]	0.310471000	0.293495000	0.531575000	0.211908000	0.364146000
[446]	0.793158000	0.254780000	0.169044000	0.463107000	0.684894000
[451]	0.507378000	0.453532000	0.334169000	0.582231000	0.148075000
[456]	0.226900000	0.208567000	0.124325000	0.463481000	0.501460000
[461]	0.320428000	0.341463000	0.623786000	0.423408000	0.364871000
[466]	0.218179000	0.386873000	0.266649000	0.257264000	0.526007000
[471]	0.153022000	0.559826000	0.588411000	0.580675000	0.673790000
[476]	0.444306000	0.464149000	0.410450000	0.393953000	0.616887000
[481]	0.200202000	0.291122000	0.282011000	0.487832000	0.513056000
[486]	0.268386000	0.405461000	0.456749000	0.279985000	0.516488000
[491]	0.444733000	0.548743000	0.359194000	0.483591000	0.412132000
[496]	0.700931000	0.594182000	0.109199000	0.217932000	0.829344000
[501]	0.338240000	0.360653000	0.360860000	0.565029000	0.725347000

[506]	0.223271000	0.338390000	0.363493000	0.596120000	0.347514000
[511]	0.354996000	0.339363000	0.699644000	0.370523000	0.358680000
[516]	0.450111000	0.491765000	0.505816000	0.608569000	0.506916000
[521]	0.361761000	-0.068206400	0.452132000	0.918549000	0.505898000
[526]	0.482884000	0.534208000	0.392358000	0.423334000	0.285126000
[531]	0.413238000	0.436923000	0.468900000	0.661269000	0.291395000
[536]	0.527673000	0.526113000	0.170659000	0.329506000	0.270539000
[541]	0.777722000	0.452377000	0.013972300	0.623794000	0.398662000
[546]	0.365272000	0.192378000	0.322762000	0.316980000	0.454592000
[551]	0.263546000	0.327888000	0.364578000	0.376755000	0.313924000
[556]	0.499073000	0.616636000	0.840550000	0.074062700	0.251186000
[561]	0.054771100	0.445940000	0.551521000	0.502257000	0.389015000
[566]	0.361735000	0.219498000	0.189805000	0.326247000	0.344517000
[571]	0.484211000	0.377764000	0.608701000	0.441702000	0.383246000
[576]	0.586746000	0.364820000	0.522268000	0.465504000	0.178776000
[581]	0.311719000	0.568311000	0.485426000	0.259739000	0.697211000
[586]	0.514800000	0.258294000	0.416335000	0.512668000	0.282684000
[591]	0.517908000	0.140646000	0.410239000	0.482800000	0.449781000
[596]	0.497184000	0.604706000	0.418095000	0.544643000	0.285145000
[601]	0.224175000	0.494453000	0.506415000	0.636008000	0.468646000
[606]	0.704234000	-0.037500000	0.279368000	0.544486000	0.242469000
[611]	0.350709000	0.160535000	0.387361000	0.286990000	0.642856000
[616]	0.411428000	0.227884000	0.150960000	0.369545000	0.520177000
[621]	0.165857000	0.127394000	0.469639000	0.311782000	0.334611000
[626]	0.448919000	0.494595000	0.239023000	0.285496000	0.345134000
[631]	0.749082000	0.220655000	0.653153000	0.470132000	0.503939000
[636]	0.342595000	0.063801400	0.446202000	0.421056000	0.392394000
[641]	0.342206000	0.589533000	0.578385000	0.323997000	0.534539000
[646]	0.235196000	0.365518000	0.597478000	0.516167000	0.486066000
[651]	0.363746000	0.246519000	0.504422000	0.519452000	0.069428900
[656]	0.531856000	0.330132000	0.484826000	0.800748000	0.498854000
[661]	0.371960000	0.439240000	0.444465000	0.457827000	0.257351000
[666]	0.580840000	0.631966000	0.256435000	0.352377000	0.709658000
[671]	0.670749000	0.598594000	0.443061000	0.306158000	0.557066000
[676]	0.501045000	-0.057071200	0.087091700	0.270926000	0.436224000
[681]	0.242094000	0.378851000	0.276932000	0.570336000	0.396548000
[686]	0.452685000	0.745527000	0.310981000	0.439543000	0.642277000
[691]	0.549256000	0.349164000	0.348519000	0.320771000	0.271511000
[696]	0.431905000	0.121929000	0.505123000	0.217022000	0.419529000
[701]	0.424248000	0.483715000	0.218731000	0.640412000	0.197832000
[706]	0.512580000	0.495594000	0.070291000	0.530476000	0.659912000
[711]	0.581924000	0.642128000	0.520471000	0.745806000	0.308717000
[716]	0.492063000	0.443048000	0.468164000	0.513282000	0.315624000

[721]	0.320674000	0.528504000	0.644603000	0.212310000	0.639908000
[726]	0.470497000	0.496516000	0.343213000	0.270204000	0.379038000
[731]	0.418108000	0.430943000	0.401455000	0.575252000	0.541547000
[736]	0.393414000	0.212738000	0.340396000	0.499440000	0.235623000
[741]	0.338879000	0.282706000	0.206294000	0.435824000	0.665141000
[746]	0.259052000	0.594452000	0.394524000	0.586816000	0.744482000
[751]	0.530794000	0.289621000	0.549959000	0.611997000	0.325231000
[756]	0.550366000	0.515123000	0.473736000	0.311607000	0.344096000
[761]	0.293142000	0.364606000	0.411226000	-0.094442800	0.293154000
[766]	0.374756000	0.342346000	0.342178000	0.560478000	0.341297000
[771]	0.574183000	0.224137000	0.478849000	0.447845000	0.369016000
[776]	0.351541000	0.500585000	0.680716000	0.187488000	0.548050000
[781]	0.641333000	0.483065000	0.622159000	0.615860000	0.357389000
[786]	0.800863000	0.426419000	0.495869000	0.478116000	0.450173000
[791]	0.206841000	0.457978000	0.518452000	0.231314000	0.354156000
[796]	0.095460800	0.170124000	0.366679000	0.483465000	0.712594000
[801]	0.353645000	0.316419000	0.254696000	0.762491000	0.678649000
[806]	0.579744000	0.615396000	0.511854000	0.375604000	0.320843000
[811]	0.554467000	0.510075000	0.924599000	0.147991000	0.544493000
[816]	0.393849000	0.474514000	0.384017000	0.922423000	0.338610000
[821]	0.532540000	0.584665000	0.173133000	0.476022000	-0.066287800
[826]	0.410223000	0.351364000	0.416501000	0.373275000	0.398297000
[831]	0.264561000	0.421238000	0.562921000	0.445709000	0.284368000
[836]	0.552462000	0.479165000	0.551044000	0.268601000	0.695576000
[841]	0.669226000	0.333283000	0.702960000	0.092193300	0.432055000
[846]	0.395065000	0.226392000	0.262272000	0.481834000	0.520796000
[851]	0.593507000	0.382094000	0.556806000	0.447103000	0.671676000
[856]	0.291013000	0.483122000	0.705429000	0.517781000	0.291123000
[861]	0.104221000	0.306840000	0.290636000	0.489997000	0.584703000
[866]	0.309228000	0.543839000	0.566293000	0.544280000	0.735511000
[871]	0.467892000	0.538692000	0.620280000	0.418298000	0.300483000
[876]	0.747756000	0.425497000	0.120479000	0.383490000	0.459773000
[881]	0.210132000	0.825866000	0.164896000	0.732588000	0.440238000
[886]	0.348547000	0.587806000	0.296891000	0.690000000	0.233048000
[891]	0.283786000	0.368916000	0.272067000	0.664856000	0.354695000
[896]	0.539105000	0.468086000	0.541323000	0.414105000	0.625241000
[901]	0.405026000	0.518818000	0.360469000	0.342839000	0.521464000
[906]	0.641419000	0.461478000	0.375566000	0.729574000	0.663008000
[911]	0.420705000	0.566021000	0.401666000	0.566656000	0.444817000
[916]	0.439147000	0.416931000	0.624148000	0.213480000	0.604366000
[921]	0.351505000	0.410805000	0.315089000	0.305651000	0.573664000
[926]	0.401782000	0.498152000	0.478445000	0.349685000	0.317631000
[931]	0.466053000	0.473464000	0.301654000	0.624360000	0.519383000

[936]	0.318701000	0.647995000	0.352499000	0.667065000	0.172598000
[941]	0.536580000	0.342035000	0.459616000	0.386392000	0.088130100
[946]	0.366637000	0.792778000	0.281764000	0.437871000	0.270198000
[951]	0.427848000	0.642483000	0.261209000	0.725766000	0.319146000
[956]	0.484518000	0.335976000	0.442056000	0.365064000	0.568045000
[961]	0.718109000	0.588922000	0.179580000	0.504605000	0.883473000
[966]	0.305365000	0.511605000	0.167203000	0.423565000	0.436512000
[971]	0.356612000	0.642507000	0.567444000	0.306239000	0.943887000
[976]	0.609598000	0.599444000	0.476215000	0.536275000	0.503654000
[981]	0.454704000	0.405286000	0.466209000	0.424515000	0.305183000
[986]	0.349470000	0.249990000	0.412602000	0.545365000	0.501043000
[991]	0.507888000	0.227949000	0.505925000	0.184584000	0.445258000
[996]	0.584438000	0.533552000	0.284308000	0.444802000	0.488400000
[1001]	0.421328000	0.705777000	0.536055000	0.479271000	0.616458000
[1006]	0.420399000	0.539628000	0.460340000	0.634101000	0.350519000
[1011]	0.373353000	0.414492000	0.466741000	0.745862000	0.624658000
[1016]	0.431750000	0.345610000	0.258336000	0.540977000	0.506518000
[1021]	0.510738000	0.355150000	0.559515000	0.487455000	0.815170000
[1026]	0.479928000	0.315849000	0.454087000	0.433380000	0.240653000
[1031]	0.873389000	0.591226000	0.502187000	0.267547000	0.372052000
[1036]	0.706394000	0.499758000	0.386187000	0.299295000	0.414673000
[1041]	0.331229000	0.297325000	0.567044000	0.282387000	0.697794000
[1046]	0.141496000	0.289559000	0.192034000	0.228596000	0.453216000
[1051]	0.473314000	0.406696000	0.324123000	0.201944000	0.559943000
[1056]	0.460991000	0.456338000	0.424442000	0.172585000	0.401756000
[1061]	0.348796000	0.567386000	0.362077000	0.632892000	0.538074000
[1066]	0.256177000	0.565637000	0.783358000	0.441620000	0.579897000
[1071]	0.379274000	0.523832000	0.596703000	0.728775000	0.638444000
[1076]	0.424937000	0.455783000	0.455396000	0.471061000	0.302995000
[1081]	0.356457000	0.500952000	0.323124000	0.866250000	0.865534000
[1086]	0.421853000	0.504101000	0.576917000	0.233390000	0.549037000
[1091]	0.305777000	0.552933000	0.340239000	0.696863000	0.612350000
[1096]	0.160666000	0.190688000	0.369034000	0.291935000	0.435060000
[1101]	0.663138000	0.455001000	0.406429000	0.540043000	0.624933000
[1106]	0.524382000	0.294659000	0.298424000	0.418144000	0.494627000
[1111]	0.395046000	0.753183000	0.666729000	0.298613000	0.650827000
[1116]	0.439683000	0.389458000	0.417685000	0.417403000	0.288269000
[1121]	0.163443000	0.187670000	0.603067000	0.829394000	0.484822000
[1126]	0.306439000	0.586301000	0.367506000	0.342679000	0.236131000
[1131]	0.363381000	0.323687000	0.524780000	0.733783000	0.585058000
[1136]	0.439881000	0.641861000	0.740489000	0.741217000	0.485712000
[1141]	0.593374000	0.479685000	-0.017194100	0.273130000	0.480352000
[1146]	0.555645000	0.532456000	0.274620000	0.304232000	0.495762000

[1151]	0.270865000	0.027022600	0.777781000	0.300974000	0.611645000
[1156]	0.462732000	0.333784000	0.894743000	0.521686000	0.481962000
[1161]	0.093346500	0.523632000	0.466650000	0.925692000	0.427607000
[1166]	0.440827000	0.563032000	0.627645000	0.627576000	0.589511000
[1171]	0.443641000	0.292376000	0.473474000	0.317821000	0.384331000
[1176]	0.753848000	0.238910000	0.567901000	0.381543000	0.290969000
[1181]	0.340329000	0.455871000	0.445628000	0.441652000	0.541175000
[1186]	0.469142000	0.496670000	0.511749000	0.364261000	0.333491000
[1191]	0.244227000	0.257646000	0.385717000	0.394200000	0.355136000
[1196]	0.124119000	0.665299000	0.539423000	0.437342000	0.556097000
[1201]	0.514993000	0.207131000	0.339955000	0.473330000	0.223099000
[1206]	0.372382000	0.482891000	0.353971000	0.453048000	0.448081000
[1211]	0.663677000	0.343589000	0.479828000	0.240017000	0.162900000
[1216]	0.486361000	0.483171000	0.351009000	0.457208000	0.400811000
[1221]	0.486689000	0.381067000	0.737750000	0.629007000	0.249488000
[1226]	0.082049700	0.298583000	0.392305000	0.453910000	0.321298000
[1231]	0.305202000	0.621668000	0.359130000	0.755010000	0.753340000
[1236]	0.522683000	0.285431000	0.496001000	0.537563000	0.594065000
[1241]	0.525071000	0.316931000	0.433967000	0.000968910	0.931932000
[1246]	0.370476000	0.216497000	0.540307000	0.267988000	0.582289000
[1251]	0.430166000	0.271100000	0.465894000	0.531175000	0.463238000
[1256]	0.238821000	0.675052000	0.214935000	0.176050000	0.292307000
[1261]	0.438527000	0.152921000	0.030245700	0.523100000	0.673898000
[1266]	0.630370000	0.407060000	0.719715000	0.603332000	0.344394000
[1271]	0.698658000	-0.061430100	0.338883000	0.510869000	0.288539000
[1276]	0.510332000	0.504013000	0.350669000	0.316644000	0.199133000
[1281]	0.456621000	0.019649600	0.123512000	0.411492000	0.631659000
[1286]	0.462566000	0.283231000	0.361990000	0.591776000	0.323664000
[1291]	0.245521000	0.711017000	0.407174000	0.282130000	0.696752000
[1296]	0.655708000	0.399407000	0.430534000	0.592696000	0.448691000
[1301]	0.499824000	0.207456000	0.430312000	0.373193000	0.528898000
[1306]	0.476586000	0.365780000	0.470225000	0.628676000	0.539700000
[1311]	0.581914000	0.411323000	0.674968000	0.386816000	0.621647000
[1316]	0.943301000	0.415640000	0.075525100	0.530848000	0.640164000
[1321]	0.404920000	0.559171000	0.349410000	0.445416000	0.554649000
[1326]	0.391931000	0.308280000	0.252981000	0.638386000	0.408114000
[1331]	0.347819000	0.532271000	0.331134000	0.344840000	0.706753000
[1336]	0.403575000	0.527803000	0.311392000	0.266385000	0.705924000
[1341]	0.470840000	0.559627000	0.086917500	0.298359000	0.488550000
[1346]	0.234536000	-0.015699400	0.400004000	0.538273000	0.365434000
[1351]	0.698655000	0.578299000	0.442571000	0.325490000	0.594698000
[1356]	0.377640000	0.439193000	0.252267000	0.730877000	0.674769000
[1361]	0.360362000	0.172088000	0.433337000	0.520333000	0.571345000

[1366]	0.376532000	0.532947000	0.362843000	0.245068000	0.358989000
[1371]	0.762875000	0.203972000	0.461270000	0.563766000	0.227767000
[1376]	0.443235000	0.267580000	0.394853000	0.496347000	0.371976000
[1381]	0.296030000	0.375157000	0.608530000	0.313142000	0.198738000
[1386]	0.383366000	0.584321000	0.358047000	0.409141000	0.575545000
[1391]	0.450693000	0.291959000	0.456938000	0.528966000	0.436001000
[1396]	0.224097000	0.631273000	0.197475000	0.292759000	0.227467000
[1401]	0.301513000	0.528280000	0.378339000	0.219584000	0.321151000
[1406]	0.357555000	0.110373000	0.414266000	0.375525000	0.082788000
[1411]	0.512833000	0.720828000	0.169545000	0.369957000	0.489479000
[1416]	0.284617000	0.559183000	0.315658000	0.493984000	0.302695000
[1421]	0.699633000	0.540306000	0.202419000	0.304166000	0.449071000
[1426]	0.548738000	0.465302000	0.722823000	0.625322000	0.668174000
[1431]	0.592271000	0.324905000	0.510486000	0.482155000	0.280979000
[1436]	0.186887000	0.764666000	0.166646000	0.504098000	0.719573000
[1441]	0.438492000	0.423484000	0.428671000	0.209582000	0.387191000
[1446]	0.519834000	0.908822000	0.480963000	0.239605000	0.487250000
[1451]	0.082224200	0.264479000	0.375974000	0.207507000	0.149346000
[1456]	0.359365000	0.468565000	0.252066000	0.058155100	0.292766000
[1461]	0.418330000	0.355468000	0.784456000	0.427777000	0.334939000
[1466]	0.589164000	0.460866000	0.512665000	0.122371000	0.676016000
[1471]	0.513069000	0.251348000	0.496553000	0.271871000	0.460000000
[1476]	0.533494000	0.068099800	0.272410000	0.565933000	0.281959000
[1481]	0.477968000	0.526945000	0.316578000	0.722016000	0.401276000
[1486]	0.284041000	0.433052000	0.457835000	0.446933000	0.214085000
[1491]	0.263695000	0.413489000	0.499261000	0.521390000	0.392394000
[1496]	0.359938000	0.657597000	0.010236900	0.292803000	0.289602000
[1501]	0.416141000	0.063670200	0.719514000	0.503756000	0.432605000
[1506]	0.472540000	0.195511000	0.471263000	0.425760000	0.799927000
[1511]	0.233106000	0.471587000	0.570355000	0.493492000	0.426580000
[1516]	0.178417000	0.556871000	0.163165000	0.392153000	0.156862000
[1521]	0.303088000	0.333426000	0.469127000	0.444842000	0.214365000
[1526]	0.370109000	0.466505000	0.176115000	0.748524000	0.416876000
[1531]	0.797816000	0.519060000	0.623772000	0.331599000	0.374058000
[1536]	0.670158000	-0.000277412	0.269233000	0.065661100	0.368638000
[1541]	0.466615000	0.532457000	0.584000000	0.503665000	0.543550000
[1546]	0.405708000	0.463010000	0.506107000	0.455478000	0.415923000
[1551]	0.354188000	0.267672000	0.508219000	0.440361000	0.541203000
[1556]	0.490865000	0.227165000	0.498549000	0.563035000	0.381316000
[1561]	0.179159000	1.144380000	0.338147000	0.065797100	0.539937000
[1566]	0.599832000	0.173492000	0.351253000	0.264599000	0.629364000
[1571]	0.223833000	0.598635000	0.374699000	0.507145000	0.636434000
[1576]	0.240180000	0.322784000	0.191313000	0.498399000	0.391480000

[1581]	0.555742000	0.766829000	0.291186000	0.673788000	0.604655000
[1586]	0.504245000	0.308491000	0.383236000	0.427266000	0.338059000
[1591]	0.151754000	0.214349000	0.674110000	0.443501000	0.504497000
[1596]	0.543791000	0.755722000	0.130687000	0.273503000	0.465918000
[1601]	0.518653000	0.713734000	0.564782000	0.685346000	0.613461000
[1606]	0.503519000	0.058013900	0.609507000	0.052321300	0.305479000
[1611]	0.696061000	0.660421000	0.264702000	0.447268000	0.281354000
[1616]	0.179029000	0.705529000	0.433019000	0.308796000	0.244943000
[1621]	0.149303000	0.538548000	0.219460000	0.488989000	0.382184000
[1626]	0.598140000	0.363724000	0.488944000	0.397976000	0.557953000
[1631]	0.451410000	0.704458000	0.528600000	0.480539000	0.526434000
[1636]	0.515502000	0.496218000	0.188872000	0.530987000	0.494752000
[1641]	0.320259000	0.108619000	0.261277000	0.353496000	-0.005107330
[1646]	0.312999000	0.219381000	0.365664000	0.425053000	0.483458000
[1651]	0.708917000	0.607820000	0.362810000	0.538324000	0.442739000
[1656]	0.404151000	0.424539000	0.271030000	0.271285000	0.397328000
[1661]	0.492976000	0.232566000	0.334886000	0.290498000	0.505438000
[1666]	0.466423000	0.555363000	0.624151000	0.425090000	0.559802000
[1671]	0.411857000	0.429790000	0.527531000	0.473813000	0.229904000
[1676]	0.935129000	0.490329000	0.624291000	0.531374000	0.445318000
[1681]	0.532620000	0.324133000	0.314902000	0.551489000	0.543412000
[1686]	0.300677000	0.434019000	0.358965000	0.293311000	0.474085000
[1691]	0.663344000	0.635092000	0.249730000	0.354034000	0.382173000
[1696]	0.055421400	0.776711000	0.519827000	0.276627000	0.098966700
[1701]	0.559086000	0.187644000	0.477698000	0.389040000	0.545658000
[1706]	0.650178000	0.346288000	0.207958000	0.294603000	0.578541000
[1711]	0.546270000	0.449490000	0.372403000	0.299039000	0.502356000
[1716]	0.255529000	0.439573000	0.695059000	0.281493000	0.178421000
[1721]	0.643501000	0.423527000	0.199001000	0.332301000	0.310815000
[1726]	0.491407000	0.387838000	0.085145400	0.252071000	0.328484000
[1731]	0.293533000	0.370451000	0.448210000	0.397448000	0.283775000
[1736]	0.192155000	0.762303000	0.394176000	0.658054000	0.447243000
[1741]	0.242265000	0.263459000	0.230038000	0.504734000	0.160006000
[1746]	0.060375600	0.419532000	0.448979000	0.119536000	0.186286000
[1751]	0.932559000	0.526315000	0.293008000	0.708545000	0.426758000
[1756]	0.456410000	0.216028000	0.286654000	0.211586000	0.399880000
[1761]	0.325431000	0.235605000	0.428918000	0.783326000	0.339449000
[1766]	0.404407000	0.309246000	0.459373000	0.457158000	0.466350000
[1771]	0.254120000	0.856470000	0.944769000	0.294358000	0.295455000
[1776]	0.533225000	0.312598000	0.607874000	0.599097000	0.537526000
[1781]	0.561192000	0.562489000	0.571695000	0.243225000	0.497123000
[1786]	0.375625000	0.321240000	0.310815000	0.194141000	0.258216000
[1791]	0.379607000	0.071312900	0.487842000	0.373417000	0.313459000

[1796]	0.315969000	0.282570000	0.404434000	0.266741000	0.523366000
[1801]	0.134031000	0.372323000	0.273481000	0.355899000	0.715096000
[1806]	0.587610000	0.697878000	0.317147000	0.377386000	0.590355000
[1811]	0.487595000	0.652255000	0.261515000	0.663902000	0.316677000
[1816]	0.495094000	0.180271000	0.371365000	0.548773000	0.471231000
[1821]	0.921382000	0.799241000	0.619094000	0.459513000	0.844523000
[1826]	0.667814000	0.272902000	0.711809000	0.613627000	0.662045000
[1831]	0.293450000	0.268354000	0.318129000	0.661551000	0.174311000
[1836]	0.479766000	0.654369000	0.510874000	0.440034000	0.638316000
[1841]	0.356126000	0.244233000	0.572358000	0.564832000	0.437756000
[1846]	0.527302000	0.125884000	0.280985000	0.600917000	0.183075000
[1851]	0.367586000	0.481375000	0.329196000	0.425191000	0.451019000
[1856]	0.328292000	0.437342000	0.272996000	0.585265000	0.609470000
[1861]	0.598068000	0.693866000	0.525540000	0.770957000	0.501503000
[1866]	0.474212000	0.501494000	0.476704000	0.666643000	0.381432000
[1871]	0.240602000	0.331358000	0.488445000	0.466944000	0.374804000
[1876]	0.252544000	0.143865000	0.433580000	0.258853000	0.282418000
[1881]	0.415911000	0.192186000	0.214573000	0.561819000	0.454534000
[1886]	0.285205000	0.377932000	0.587595000	0.548505000	0.412074000
[1891]	0.445422000	0.458676000	0.316867000	0.439486000	0.213897000
[1896]	0.463375000	0.337398000	0.370164000	0.045814500	0.532329000
[1901]	0.335711000	0.183619000	0.109904000	0.873407000	0.586685000
[1906]	0.526682000	0.633221000	0.379599000	0.528578000	0.355794000
[1911]	0.697603000	0.505477000	0.598669000	0.450702000	0.547151000
[1916]	0.197719000	0.541820000	0.368323000	0.196558000	0.260464000
[1921]	0.795324000	0.202869000	0.488417000	0.524108000	0.750523000
[1926]	0.332195000	0.504942000	0.423103000	0.428022000	0.492765000
[1931]	0.337275000	0.346942000	0.558164000	0.426102000	0.378510000
[1936]	0.433394000	0.679676000	0.357069000	0.070179300	0.295653000
[1941]	0.493559000	0.733532000	0.541965000	0.278290000	0.127191000
[1946]	0.164056000	0.374429000	0.621150000	0.416985000	0.419174000
[1951]	0.182727000	0.500728000	0.468374000	0.466443000	0.870790000
[1956]	0.580139000	0.418264000	0.198653000	0.237574000	0.295927000
[1961]	0.445496000	0.587132000	0.439039000	0.745079000	0.156954000
[1966]	0.200124000	0.618509000	0.334444000	0.465142000	0.488561000
[1971]	0.349108000	0.402642000	0.394801000	0.495198000	0.392423000
[1976]	0.456714000	0.067358900	0.444283000	0.336842000	0.245902000
[1981]	0.652555000	0.554870000	0.137729000	0.167548000	0.556948000
[1986]	0.520072000	0.392647000	0.590255000	0.421941000	0.302788000
[1991]	0.560196000	0.516304000	0.534066000	0.802780000	0.643950000
[1996]	0.230359000	0.816865000	0.689767000	0.540282000	0.544309000
[2001]	0.666211000				

```
post_MODU_G$alpha$`alpha_fruitAB[1]`
```

```
[1] -3.26141e-01 1.15291e-01 -2.71057e-01 -1.52363e-01 -2.00212e-01
[6] -1.72336e-01 -1.97616e-01 -1.58813e-01 5.68117e-02 -9.66968e-02
[11] -4.73112e-02 -2.20166e-01 -1.65034e-01 -2.36847e-01 -2.66913e-01
[16] -2.71105e-01 -9.25404e-02 -1.52579e-01 -2.85097e-01 -6.53541e-02
[21] -2.96656e-01 -5.52005e-02 1.11494e-01 5.76822e-04 -1.53369e-01
[26] 1.41681e-01 -3.57180e-01 -2.04561e-01 -2.96894e-01 -4.45695e-01
[31] -3.11519e-02 -1.10742e-01 1.15837e-01 -5.46389e-01 -2.09110e-01
[36] -3.27497e-01 -1.51845e-01 -2.10595e-01 -2.72184e-01 -3.90907e-01
[41] 8.24023e-02 -2.71698e-01 -3.81769e-01 -2.13046e-01 -1.73457e-01
[46] -2.10510e-01 -1.64487e-01 -1.15978e-01 -1.21975e-01 -9.64718e-02
[51] -1.69670e-01 -1.20369e-01 -2.18640e-01 -1.46519e-01 -2.38577e-01
[56] -1.31152e-01 -7.56698e-02 -2.49173e-01 -4.74596e-01 -1.12208e-01
[61] 7.38394e-02 -2.45382e-01 -4.83752e-02 -6.03026e-02 -4.00158e-01
[66] -1.17017e-01 -1.91744e-01 -4.60706e-01 -4.38895e-01 -1.30667e-01
[71] -2.22903e-01 -7.77635e-02 -1.03676e-01 -2.73280e-01 -1.31166e-01
[76] -6.47780e-02 -1.06487e-01 -6.36951e-04 -6.33722e-02 -2.19589e-01
[81] -1.17429e-01 9.93421e-02 -3.72156e-01 -1.06997e-01 -1.09684e-01
[86] -1.36896e-01 -1.11249e-01 -4.31149e-01 -9.68829e-02 -1.35976e-01
[91] -6.76349e-02 -2.74903e-01 -4.34467e-01 -2.43331e-01 -2.14368e-01
[96] 7.37082e-02 -4.26860e-01 -2.05314e-01 -1.50201e-01 -2.40031e-01
[101] -1.63383e-01 -2.81276e-01 -1.19374e-01 -1.67190e-01 -2.57041e-01
[106] -2.51214e-01 -3.47397e-01 -2.33572e-01 -1.54680e-01 -2.70316e-01
[111] -1.30412e-01 -5.07391e-01 -1.42368e-01 -3.26092e-01 -2.02442e-01
[116] -3.23647e-01 -1.17951e-01 1.62764e-02 -3.91427e-02 -1.59240e-01
[121] -1.07436e-01 -3.66664e-01 -1.76410e-01 -2.38645e-01 -5.25529e-01
[126] -2.70070e-01 -2.30422e-01 -1.08357e-01 -4.30191e-01 -2.21207e-01
[131] 1.00285e-01 -6.99847e-01 -3.62629e-01 -1.25570e-01 -2.37478e-02
[136] -1.72371e-01 -4.20067e-01 -3.34822e-01 -1.68903e-01 -3.69480e-01
[141] -3.67817e-01 -2.75776e-01 -2.56687e-01 1.28139e-01 -1.89054e-01
[146] -1.26200e-01 -1.24673e-01 -8.94188e-02 8.73654e-02 -4.30766e-01
[151] -9.71012e-02 -3.41755e-01 -3.27486e-02 -3.14489e-01 -9.31697e-02
[156] -2.20502e-03 -2.18364e-01 -1.59861e-01 2.27872e-02 -2.69021e-01
[161] 4.73688e-01 -2.48683e-01 -6.03325e-02 -2.75215e-01 -4.06414e-01
[166] -2.42776e-01 1.05327e-01 -2.46283e-01 -4.05208e-01 -2.20290e-01
[171] -1.79733e-01 -2.61823e-01 6.87422e-03 -1.31765e-01 -9.68790e-02
[176] -1.32467e-01 -2.34051e-01 -1.91626e-01 -3.71317e-01 -2.68548e-01
[181] -2.84382e-01 -3.16093e-01 -4.89163e-01 -2.46467e-01 -2.37725e-01
[186] -1.83192e-01 -3.47014e-01 -4.01839e-01 -2.01526e-01 -1.20562e-01
[191] -1.95842e-01 -4.93328e-01 -1.73787e-01 -1.52034e-01 -3.93317e-01
[196] -2.97930e-01 -2.65162e-01 -1.80165e-02 -1.04426e-01 -5.01970e-01
```

[201]	-3.06004e-01	-3.42596e-01	5.62055e-02	1.82335e-03	-8.60558e-02
[206]	3.75254e-02	-8.84877e-02	-1.36909e-01	-7.14019e-02	-4.48179e-01
[211]	-1.06064e-02	-1.16365e-02	-2.23217e-01	-3.25375e-01	-3.28106e-01
[216]	-9.57279e-02	-5.64116e-01	-2.27167e-01	-2.30764e-01	-5.69850e-03
[221]	-3.62809e-01	-1.38098e-01	-1.29982e-01	-3.29882e-01	-3.81112e-01
[226]	-1.12543e-01	6.45100e-02	-4.94436e-01	-2.80153e-01	-1.51032e-01
[231]	-1.64116e-01	-4.58519e-01	5.56082e-02	-3.61003e-01	-3.17617e-02
[236]	-1.61805e-01	1.64509e-01	-2.93812e-01	-1.26415e-01	-8.69917e-02
[241]	6.45939e-02	1.13266e-01	-3.40484e-01	-1.77414e-01	1.22141e-01
[246]	-2.50994e-01	-2.25380e-01	-2.24256e-01	4.60737e-02	1.28609e-02
[251]	-7.83196e-02	-5.83662e-02	-1.86000e-01	2.19315e-02	-1.88691e-01
[256]	1.01898e-01	5.86207e-02	-9.81569e-02	-2.66106e-01	-1.36766e-01
[261]	-6.04476e-02	-3.51038e-01	-7.39432e-02	-1.88230e-01	-3.70745e-01
[266]	-1.09879e-01	-1.30868e-01	-7.53516e-02	-1.21322e-01	-1.06916e-02
[271]	-3.23962e-01	-1.67896e-01	-1.60729e-01	-9.56939e-02	-3.75517e-01
[276]	-2.19980e-01	-5.90665e-01	-5.72068e-02	5.90794e-03	6.12326e-02
[281]	-5.99055e-02	-4.60280e-01	-3.54697e-01	-4.04511e-01	-6.07721e-02
[286]	-4.61228e-01	-5.61975e-01	-3.33527e-01	-5.97969e-02	-1.32399e-01
[291]	3.45234e-04	-2.04658e-01	-6.15673e-01	-4.40705e-01	-7.72077e-02
[296]	-1.42301e-01	-1.81655e-01	-1.36919e-01	-3.05097e-02	-1.74112e-01
[301]	-1.49082e-01	-1.47074e-01	-2.98201e-01	1.77655e-02	-1.15193e-01
[306]	-1.22753e-01	-2.02049e-01	-2.48524e-01	-5.32642e-02	-3.49233e-01
[311]	-1.41643e-02	-4.47393e-01	2.76232e-02	-1.41215e-01	-1.38451e-01
[316]	-2.25926e-01	1.55518e-01	-1.50243e-02	-1.76947e-01	-4.23959e-01
[321]	-3.20573e-02	-1.84759e-01	-4.77421e-01	-4.40820e-01	-4.31537e-02
[326]	-9.26259e-02	-1.57356e-01	4.81050e-04	-9.58059e-03	-4.11650e-02
[331]	-1.56459e-01	-3.87491e-01	-1.26574e-01	-4.05463e-01	-2.35544e-01
[336]	-2.42953e-01	-2.76062e-01	-1.33388e-01	1.12120e-02	-3.10587e-01
[341]	-2.38651e-01	-2.83394e-01	-4.21912e-02	-2.84223e-01	2.64651e-02
[346]	-4.45491e-01	-3.96156e-01	1.59215e-01	-1.65008e-01	-1.57079e-01
[351]	-1.79386e-01	-6.44495e-02	-3.50339e-01	-2.32497e-01	-2.97950e-01
[356]	-1.44615e-01	-8.94199e-02	-2.11230e-01	-2.50395e-01	-3.83279e-01
[361]	-1.19223e-02	-3.78857e-01	-3.33412e-01	-2.38432e-01	-2.98837e-01
[366]	-5.59083e-02	-1.28681e-01	-1.77742e-01	-2.28686e-01	-4.59947e-01
[371]	-2.96232e-02	-3.83509e-01	-3.09863e-01	-1.58282e-01	7.24758e-02
[376]	-1.07018e-01	-3.93331e-01	-4.50038e-01	-6.08335e-01	-3.17375e-01
[381]	-3.98863e-01	-2.43802e-01	-3.17487e-01	-3.29734e-01	-1.88792e-01
[386]	6.49624e-02	-7.92376e-02	-5.73702e-02	-3.10096e-01	-2.35773e-01
[391]	-2.99867e-01	-5.42120e-02	-2.80221e-02	-3.43134e-01	-2.12194e-01
[396]	-1.77418e-01	-1.72004e-01	-3.51563e-01	-2.95774e-01	-1.24635e-01
[401]	-1.55918e-01	-1.74071e-01	-2.77650e-01	-7.01388e-02	-3.03082e-01
[406]	-7.35031e-02	-3.29047e-01	-1.01909e-01	-1.65560e-01	-2.00134e-01
[411]	-2.33510e-01	-2.46717e-01	-2.22787e-01	-2.98045e-01	-2.96187e-01

[416] -2.53541e-04 -1.18858e-01 1.46560e-01 -1.82658e-02 -1.14958e-01  
 [421] -2.44487e-01 -5.58946e-02 -3.47006e-01 -5.02138e-03 -1.95526e-01  
 [426] 1.74518e-02 -5.73510e-03 -2.90574e-01 -1.04898e-01 -2.58796e-01  
 [431] -5.53269e-02 -2.32815e-01 -1.19114e-01 -4.16850e-01 -1.42467e-01  
 [436] -9.24084e-03 -4.11657e-01 -5.15047e-02 -1.66566e-01 -7.86755e-02  
 [441] -2.51501e-01 -1.37225e-01 -1.52523e-01 -1.33078e-01 -4.13672e-01  
 [446] -1.55941e-01 -2.06108e-01 -1.63161e-01 -1.69348e-01 -2.41164e-01  
 [451] -2.97156e-01 -2.27250e-01 -1.38438e-01 -2.05202e-01 -8.08557e-02  
 [456] -3.76395e-01 -1.42867e-01 -1.07510e-01 -3.69273e-01 -1.67685e-02  
 [461] -2.35361e-01 -1.89302e-01 -1.11297e-02 -1.44118e-01 -5.45788e-02  
 [466] 6.21365e-02 -3.92521e-01 -3.50480e-01 -3.43477e-01 -5.57761e-01  
 [471] -1.32729e-01 -2.30257e-01 -3.00240e-02 -7.92230e-02 1.31137e-01  
 [476] -4.63254e-02 -9.67495e-02 -2.99811e-01 -1.26028e-01 -3.11593e-01  
 [481] -2.79889e-01 -5.55162e-02 -9.99174e-02 -2.59290e-01 -1.18842e-01  
 [486] -3.37160e-01 -2.63413e-01 -1.78408e-01 -1.79508e-01 -1.30232e-01  
 [491] -2.21528e-01 -2.56628e-01 -5.63394e-02 -4.98256e-01 -5.56780e-02  
 [496] -1.75980e-01 -7.67189e-02 -3.07272e-01 -1.65517e-01 -3.84691e-01  
 [501] -1.33228e-01 -4.28578e-01 -1.73542e-01 -3.70586e-01 -9.53184e-02  
 [506] -5.03030e-01 -1.35598e-01 -2.85567e-01 -3.72765e-01 -3.53029e-01  
 [511] -6.01250e-01 -7.52619e-02 -7.21182e-01 -2.11250e-02 -1.35097e-01  
 [516] -1.24358e-01 -4.17993e-01 -3.64309e-01 -4.72288e-01 -2.89699e-01  
 [521] -3.62266e-01 -9.19146e-02 -3.08525e-01 -3.79276e-01 -7.99999e-02  
 [526] -1.99121e-01 -3.39436e-01 -2.13951e-01 -3.22245e-01 -4.16230e-02  
 [531] -6.99135e-02 -2.84453e-01 -3.79851e-01 -1.93163e-01 -1.86360e-01  
 [536] -2.69337e-01 -2.60203e-01 2.86608e-02 -3.78947e-01 -1.39945e-01  
 [541] -3.41684e-01 3.16775e-02 1.11754e-01 -2.07875e-01 -2.22921e-01  
 [546] -1.01710e-01 1.02410e-01 -6.56006e-01 -1.81888e-01 -2.85727e-01  
 [551] -2.70472e-01 1.06443e-03 2.54143e-02 -2.74728e-01 -3.44314e-01  
 [556] 8.17874e-03 -4.31781e-01 -2.61787e-01 -3.17021e-01 -2.85792e-01  
 [561] -3.76669e-01 -2.86564e-01 -2.04840e-01 -5.69526e-01 -1.57354e-01  
 [566] -3.97414e-01 -1.17488e-01 -3.15200e-01 -7.90162e-02 -3.12519e-01  
 [571] -1.39552e-01 -1.97373e-02 -7.54147e-02 -6.15451e-02 -1.29297e-01  
 [576] -2.04527e-01 7.16000e-02 -2.66997e-01 -1.55352e-01 -2.08397e-02  
 [581] -6.85263e-02 -2.59851e-01 -2.42688e-01 -3.11647e-01 -1.89551e-01  
 [586] 5.71692e-02 -4.92695e-01 -2.71886e-02 -4.92476e-02 -3.61033e-01  
 [591] -1.44322e-01 8.24253e-02 -2.73729e-01 -3.21325e-01 -3.61651e-01  
 [596] -2.69894e-01 -1.21448e-01 -3.09323e-01 -1.36193e-01 -2.47091e-01  
 [601] 5.74667e-02 -6.48257e-02 -4.02678e-01 -3.56669e-01 -1.32996e-01  
 [606] -2.85494e-02 -1.31918e-01 -3.98308e-01 1.09354e-01 -1.32913e-01  
 [611] -5.44735e-02 -1.01928e-01 -1.54839e-01 -2.77781e-01 -2.48069e-01  
 [616] -2.84354e-01 -5.56809e-02 -1.06750e-01 2.18611e-02 -6.75990e-02  
 [621] -2.86710e-01 8.22308e-02 -4.12436e-02 -4.28769e-01 -2.15628e-01  
 [626] -9.60483e-02 -8.16551e-02 -1.57308e-01 -6.28116e-01 -2.10585e-01

[631] -4.34813e-01 -1.02557e-01 -2.94191e-01 -3.10985e-01 -2.50890e-01  
 [636] -1.35190e-01 -2.60692e-01 -2.32142e-01 -1.41312e-01 -1.51015e-01  
 [641] -2.98538e-01 -1.20974e-01 -9.36782e-02 -1.46368e-01 -1.59328e-01  
 [646] -2.53750e-01 -1.21512e-01 -3.28254e-01 -3.38074e-01 -1.11526e-01  
 [651] -3.41325e-01 3.50582e-02 -2.94540e-01 -3.19686e-01 8.53587e-02  
 [656] -4.48214e-01 5.11893e-03 -1.82991e-01 -3.73443e-01 -1.40698e-01  
 [661] -4.10425e-01 -2.70061e-01 -1.29015e-01 -1.08146e-01 -5.42340e-01  
 [666] -5.22681e-01 -3.06697e-01 -9.99505e-03 -2.08365e-01 -2.52651e-01  
 [671] -1.72473e-01 -3.57949e-01 -2.77663e-01 -1.81849e-01 -1.91027e-01  
 [676] -1.20578e-01 7.66820e-02 -7.60690e-02 -6.05343e-02 -2.26277e-01  
 [681] -2.24969e-01 -1.02618e-01 -2.56960e-01 -1.57020e-01 -5.78258e-01  
 [686] -9.96709e-02 -2.35657e-01 -8.98931e-02 -5.00226e-01 -2.77399e-01  
 [691] -4.02719e-01 -2.21921e-01 -1.38112e-01 -3.78824e-01 -1.71082e-01  
 [696] -4.81639e-01 8.77399e-02 -5.86875e-01 1.11408e-01 -3.86595e-02  
 [701] -3.15605e-01 -1.41559e-01 -9.98558e-02 -1.02479e-01 4.84692e-01  
 [706] -2.85642e-01 -2.63882e-01 -3.48114e-01 -1.99716e-01 -1.84959e-01  
 [711] -5.96019e-01 -5.12297e-01 -5.79632e-02 -9.30476e-02 -6.20341e-02  
 [716] -2.53098e-01 -1.44747e-01 -1.70120e-01 -1.76945e-01 -1.92516e-01  
 [721] -1.07651e-01 -2.63004e-01 -7.66467e-01 -2.81237e-01 -2.35760e-01  
 [726] -1.61749e-01 -1.16761e-01 -9.31559e-02 -1.78053e-01 -1.34833e-02  
 [731] -2.14248e-01 -1.58863e-01 -7.05335e-02 -3.17709e-01 -2.50394e-01  
 [736] -4.47311e-01 -8.42222e-02 -9.71839e-03 -2.13466e-01 -2.44509e-03  
 [741] -1.07839e-01 -2.87912e-01 -1.65288e-02 -2.68403e-01 -4.46585e-01  
 [746] 3.00939e-02 -1.30932e-01 -2.92901e-01 -4.39873e-01 -3.19943e-01  
 [751] -4.92468e-01 -1.95537e-01 8.39851e-02 -2.05291e-01 -2.67600e-01  
 [756] 3.96536e-02 -2.35300e-01 -1.26525e-01 -6.48544e-02 -2.19468e-01  
 [761] -1.08161e-01 -1.89231e-01 -1.05406e-01 3.80600e-02 -1.99498e-01  
 [766] -3.19992e-01 -8.36904e-02 -2.12341e-01 -1.97686e-01 -2.15349e-01  
 [771] -3.45048e-01 -1.80848e-01 -3.74619e-01 -1.98874e-02 7.55444e-03  
 [776] -5.43559e-02 -3.32980e-01 -3.53453e-01 -4.34859e-01 -9.86796e-02  
 [781] -2.98480e-01 -3.25331e-01 -1.30392e-01 -5.13251e-01 -1.89568e-01  
 [786] -3.20832e-01 -3.56528e-01 -1.87269e-01 -3.01243e-01 -2.70126e-01  
 [791] -2.60652e-01 -1.27183e-01 -1.48783e-01 -1.68510e-01 -3.00944e-01  
 [796] -8.48547e-02 -2.25983e-02 -4.44172e-01 -6.22964e-02 -3.91262e-01  
 [801] -1.41940e-01 -7.22976e-02 6.43324e-02 -5.71177e-01 -2.96321e-01  
 [806] -4.70371e-01 -1.51043e-01 -2.27130e-01 -3.08814e-01 -1.94959e-01  
 [811] 2.22585e-02 -1.82717e-01 -5.71973e-01 2.70916e-02 -3.72013e-01  
 [816] -3.30247e-01 -2.05177e-01 -2.49872e-01 -4.33160e-01 -1.01801e-01  
 [821] -1.65745e-01 -3.04095e-01 -1.52116e-01 -4.21739e-01 -1.74111e-01  
 [826] 9.68817e-02 -1.41881e-01 -3.85567e-01 -1.02904e-01 -2.52742e-01  
 [831] -3.14516e-01 -1.55827e-02 -2.84574e-01 -1.45377e-01 -3.63488e-01  
 [836] -1.30608e-01 -1.04024e-01 -3.43387e-01 -1.28836e-01 -4.25498e-01  
 [841] -2.63993e-01 -4.46115e-02 -6.33637e-01 -3.66525e-01 -3.14505e-01

[846]	-9.64583e-02	4.04934e-02	-2.71694e-01	-2.41803e-01	-1.47778e-01
[851]	-2.57883e-01	-1.40407e-01	-1.84446e-02	-2.18443e-01	-1.53814e-01
[856]	-1.17149e-01	-1.37094e-01	-3.16328e-02	-1.12743e-01	-9.96409e-02
[861]	-1.27352e-01	-4.93088e-01	-1.38084e-01	6.46594e-03	-3.10573e-01
[866]	-1.20776e-01	4.59304e-03	-3.14775e-01	-2.83066e-01	-3.67003e-01
[871]	-2.86478e-01	-3.81047e-01	-1.55161e-01	3.07719e-02	-2.95580e-01
[876]	-4.24002e-01	9.20519e-02	-1.38621e-02	-2.31091e-01	-9.45461e-02
[881]	-4.06512e-01	-4.87166e-01	-5.22910e-01	-3.83331e-01	-5.09368e-01
[886]	-2.11229e-01	-3.37951e-01	-1.30599e-01	-4.92451e-01	-1.94461e-01
[891]	1.81342e-02	-1.35520e-01	-1.88395e-01	-4.56119e-01	-7.44916e-02
[896]	-5.90713e-02	-5.07657e-02	-1.43366e-01	-5.31548e-02	-2.63097e-01
[901]	-3.50667e-02	-2.30419e-01	-1.07250e-02	-2.80218e-01	-1.80238e-01
[906]	-2.55171e-01	-3.22369e-01	-2.91651e-01	-3.62697e-01	-4.45858e-01
[911]	-2.63121e-01	1.24557e-02	-4.42220e-01	-5.54087e-01	-1.85045e-01
[916]	-2.19675e-01	3.60092e-02	-4.89213e-01	-5.39911e-02	5.04505e-03
[921]	-4.31920e-01	-6.99649e-02	-1.29417e-01	-2.91261e-01	-3.18417e-01
[926]	-3.66023e-01	-1.49271e-01	-2.90925e-01	-1.75951e-01	-4.01060e-01
[931]	-1.64208e-01	-4.13935e-01	-2.40454e-01	-1.42042e-01	-5.60258e-02
[936]	-1.74474e-01	-2.62473e-01	-3.81580e-01	-2.05324e-01	9.25678e-02
[941]	-1.98057e-01	-2.20613e-01	-1.45226e-01	-4.03926e-01	-4.15732e-01
[946]	-9.17586e-02	-3.65268e-01	-4.50395e-02	-7.34770e-02	-2.20209e-01
[951]	-3.94944e-01	-7.59249e-02	3.67537e-03	-1.49708e-01	-1.36154e-01
[956]	-2.48378e-01	-7.77555e-02	-3.04588e-01	-2.99664e-01	-3.99181e-01
[961]	-3.18129e-01	-1.96281e-01	1.03291e-01	-1.92117e-01	-2.30045e-01
[966]	-4.01906e-01	-2.32460e-01	-4.46737e-02	-1.45257e-01	-1.77481e-01
[971]	-1.31692e-01	-1.35567e-01	-4.76129e-01	-6.60953e-02	-9.51548e-02
[976]	-1.58986e-01	-1.45448e-02	-3.66068e-01	-1.90999e-01	-5.61317e-01
[981]	-1.14110e-01	4.42375e-03	-1.94721e-01	-8.66693e-02	-1.24946e-01
[986]	-3.24809e-01	-5.45372e-02	-1.77929e-01	-2.74833e-01	-3.18411e-01
[991]	-2.22439e-01	-1.63993e-01	-5.76329e-02	-2.60274e-01	-2.07120e-01
[996]	-8.26327e-02	-1.83842e-01	-1.27336e-01	-1.37494e-01	-2.04282e-01
[1001]	-3.95390e-01	-4.08762e-01	-4.87273e-01	-1.82861e-01	-1.30567e-01
[1006]	-1.69335e-01	-3.88915e-01	-5.45178e-02	-2.54919e-02	-8.08355e-02
[1011]	-3.73797e-02	-2.93740e-01	-2.11052e-01	-1.20015e-01	-3.20029e-02
[1016]	-1.88955e-01	-1.26078e-01	-1.43926e-01	-5.77129e-01	-2.31465e-01
[1021]	-2.23889e-01	-2.66487e-01	1.39983e-01	-9.78735e-02	-2.50805e-01
[1026]	-8.46290e-02	-1.10051e-01	-3.27118e-01	-3.82473e-01	-1.30436e-01
[1031]	-9.18945e-03	-8.87313e-02	-3.44462e-01	-4.75067e-01	-2.77615e-01
[1036]	-2.94642e-01	-4.10784e-01	-7.29965e-02	-3.13256e-01	-1.52549e-01
[1041]	-1.88252e-01	-1.51343e-01	1.02722e-01	-3.30184e-01	-3.07314e-01
[1046]	-2.97510e-01	4.60311e-02	5.28795e-02	-8.67775e-02	-3.77975e-01
[1051]	-5.20188e-02	-2.55762e-01	-3.11053e-01	-1.35642e-01	-1.55656e-01
[1056]	-2.41996e-01	-1.41081e-01	-1.73045e-01	-2.55637e-01	-2.77335e-01

[1061]	-2.41175e-01	-3.41008e-01	-1.61223e-01	-5.13424e-01	9.59757e-02
[1066]	-6.34717e-02	-3.28188e-01	-5.25410e-01	-1.16136e-01	-5.31534e-01
[1071]	-2.08292e-01	1.27407e-01	-2.81599e-01	-2.43489e-01	-4.11634e-01
[1076]	-1.36046e-01	-1.52354e-01	-2.46498e-01	-1.54911e-01	-1.56925e-01
[1081]	-2.19903e-02	-2.59439e-01	-2.20297e-01	-3.43529e-01	-4.27026e-01
[1086]	-3.23225e-01	-1.51431e-01	-1.09784e-01	-2.91297e-01	-3.46545e-01
[1091]	-3.43018e-01	-1.08394e-01	-1.26912e-01	-2.11394e-01	-7.40178e-02
[1096]	6.33652e-02	-4.48260e-02	-4.82279e-02	1.78151e-02	-2.99937e-01
[1101]	-3.19822e-01	-1.69228e-02	3.08418e-02	-1.39478e-01	-3.01982e-01
[1106]	-2.16495e-01	-8.67351e-02	-1.18628e-01	-2.96772e-01	-1.58919e-01
[1111]	-3.05363e-01	-2.56049e-01	-1.39022e-01	-2.53299e-01	-2.63805e-01
[1116]	-9.68760e-02	-2.35892e-01	-1.84015e-01	-4.66621e-02	-1.92081e-01
[1121]	5.55575e-03	-1.32947e-01	-1.71362e-01	-1.82007e-01	-4.52338e-01
[1126]	-3.46825e-01	-1.07829e-02	-1.68035e-01	-3.28001e-02	2.70988e-01
[1131]	-2.16280e-01	-1.99183e-01	-3.92198e-01	-5.28833e-01	-7.00687e-02
[1136]	-1.36978e-01	-4.49646e-01	-2.33322e-01	-3.95329e-01	-2.60884e-01
[1141]	-1.92896e-01	-4.29195e-01	-2.75522e-01	-2.62317e-01	-3.97239e-01
[1146]	-4.43868e-01	-1.54286e-01	-1.38703e-01	6.40452e-02	-3.23859e-02
[1151]	-2.67629e-01	-1.74824e-01	-9.97203e-02	-5.53865e-01	-2.62285e-01
[1156]	2.23038e-03	2.09142e-01	-1.95896e-01	-2.20801e-01	-9.41058e-02
[1161]	2.00536e-01	-2.72810e-01	-1.56568e-01	-1.27324e-01	-2.64633e-01
[1166]	-3.72101e-01	-1.22544e-01	-1.86770e-01	-3.19881e-01	-8.44695e-02
[1171]	-7.87423e-02	-4.00841e-02	-4.26901e-01	-2.16740e-03	-3.32917e-01
[1176]	-1.42270e-01	-3.59358e-01	-2.22024e-01	-3.34636e-01	-5.74110e-01
[1181]	-1.39692e-01	1.87909e-02	-1.16361e-01	2.08368e-01	-5.16453e-01
[1186]	-2.63243e-01	-2.14047e-01	-3.32846e-01	-1.93245e-01	-3.08397e-01
[1191]	-1.90863e-01	-6.06082e-02	-1.35960e-01	-1.84631e-01	-1.29867e-01
[1196]	-2.33192e-01	-3.84685e-01	-2.31810e-01	-3.19772e-01	-2.66415e-01
[1201]	-3.33397e-01	-4.18404e-01	-1.11502e-01	-2.05984e-03	-1.95717e-02
[1206]	4.36004e-02	-2.00326e-01	-2.17723e-01	-2.10554e-01	-1.57747e-01
[1211]	-4.74689e-01	2.28352e-01	-3.79426e-01	-1.00299e-01	-1.95927e-01
[1216]	-2.06977e-04	-1.60326e-01	-1.94044e-02	-1.12888e-01	-1.66064e-01
[1221]	-1.28924e-01	-6.55798e-02	-1.36603e-01	-4.89091e-01	-5.20780e-02
[1226]	1.16620e-02	-3.77275e-01	-4.00393e-01	2.77208e-02	-2.62891e-01
[1231]	-3.97973e-02	-4.21382e-01	-1.43881e-01	-2.26759e-01	-1.97049e-01
[1236]	-4.93420e-01	2.09747e-01	1.42367e-02	-3.04204e-01	-3.35003e-01
[1241]	-7.51147e-02	-2.30517e-01	-2.81871e-01	-3.05429e-02	-2.70582e-01
[1246]	1.03881e-01	-5.02235e-02	-2.42403e-01	-1.37577e-01	-4.84352e-01
[1251]	-2.05937e-01	-1.49329e-01	-1.41427e-01	-1.93846e-01	-7.27489e-02
[1256]	5.91581e-02	-4.54690e-01	-1.19835e-01	-1.24819e-01	-4.07997e-01
[1261]	-3.32606e-01	-1.05950e-01	2.05243e-01	3.92064e-02	-4.17203e-01
[1266]	-3.92479e-01	-1.73332e-01	-2.40946e-01	-3.04454e-01	-3.53604e-01
[1271]	-1.57881e-01	-1.95052e-01	-1.29823e-01	-5.56654e-01	-8.01362e-02

[1276]	-2.04983e-01	-2.35158e-01	-1.74335e-01	-4.58172e-01	-2.01832e-01
[1281]	-1.79943e-01	9.13235e-02	8.89792e-02	-2.63924e-01	-2.19377e-01
[1286]	-2.80977e-01	7.56660e-02	-1.45692e-01	-1.34409e-01	-1.73215e-01
[1291]	-1.76310e-01	-2.61541e-01	-1.55126e-01	-1.48336e-01	-2.00450e-01
[1296]	-1.24723e-01	-8.01564e-02	-2.80232e-01	-5.08495e-01	-1.98809e-01
[1301]	-2.57736e-01	-1.73144e-01	2.35564e-02	-3.69013e-01	-2.90831e-01
[1306]	-2.19108e-01	-1.24914e-01	-2.13395e-01	-2.43609e-01	-1.71895e-01
[1311]	-2.62971e-04	-3.33844e-01	-3.77582e-01	-4.23048e-01	-1.72560e-01
[1316]	-5.02960e-01	1.62475e-02	-7.48923e-03	-3.62685e-02	2.96126e-02
[1321]	-2.32342e-01	-2.11669e-01	-1.98346e-01	-3.62692e-01	-2.66663e-01
[1326]	-2.97203e-01	-1.96162e-01	-4.39640e-02	-2.82915e-01	-1.33798e-01
[1331]	-1.38322e-01	-2.23139e-01	-8.74077e-04	-1.80131e-01	-2.28495e-01
[1336]	-3.92672e-01	-1.73715e-01	-2.92850e-01	-1.61435e-01	-2.10285e-01
[1341]	-2.49132e-01	-5.22489e-01	-2.10118e-01	-3.95645e-01	-3.43147e-01
[1346]	-1.16435e-01	-7.22562e-02	-4.43198e-02	2.60155e-02	-2.91105e-01
[1351]	-1.59745e-01	2.72931e-01	-2.72402e-01	-1.40913e-01	-4.62617e-01
[1356]	-2.16922e-01	-1.95759e-01	-8.80593e-03	-2.22823e-01	-2.44995e-01
[1361]	-2.37217e-01	-1.67397e-01	-8.40705e-02	-6.95926e-02	-3.29403e-01
[1366]	1.50924e-01	-7.56459e-02	-3.66638e-01	2.83016e-02	-3.19810e-03
[1371]	-2.51931e-01	-5.31282e-02	-1.15819e-01	-3.11020e-01	-1.00733e-01
[1376]	-3.26196e-01	-1.89672e-01	-5.32115e-01	-2.10092e-01	-1.16109e-01
[1381]	-1.93941e-01	-1.89380e-01	-2.48274e-01	-4.25088e-02	1.83068e-01
[1386]	-4.99386e-01	-1.71677e-01	-5.08104e-02	-1.01518e-01	-7.08699e-02
[1391]	-7.84582e-02	-1.64205e-01	-9.80339e-02	-3.04711e-01	-1.50241e-01
[1396]	-3.21769e-01	-3.39959e-01	-4.13011e-02	-2.67018e-01	-1.26667e-01
[1401]	-1.39825e-01	-2.81463e-01	-3.07242e-01	1.68495e-01	-1.94195e-02
[1406]	-4.72442e-02	-1.11877e-01	-1.04478e-01	-4.93717e-01	2.24966e-01
[1411]	-1.82467e-01	-8.13548e-02	1.18012e-01	-2.42790e-01	-6.85065e-02
[1416]	-1.15003e-01	-1.28673e-01	-3.90276e-01	-1.85990e-01	-4.54864e-02
[1421]	-3.11168e-01	-1.83175e-01	-1.01576e-01	-2.23874e-01	-1.66561e-01
[1426]	-2.40426e-01	-2.87824e-01	-3.69087e-01	-2.76534e-01	-2.43132e-01
[1431]	-4.96614e-01	-1.83152e-01	-1.12090e-01	1.44503e-02	-4.10455e-01
[1436]	-1.98586e-01	-3.59369e-01	-2.12211e-01	-1.56750e-01	-3.32328e-01
[1441]	-1.00232e-01	-6.12604e-02	-5.56495e-02	-1.85779e-02	6.69386e-02
[1446]	-1.41560e-01	-4.23863e-01	-3.68542e-01	1.96804e-02	-2.55741e-01
[1451]	1.52533e-02	-1.46872e-01	-4.53671e-02	-1.90443e-01	-3.39103e-01
[1456]	-5.81703e-02	-4.22805e-02	-1.01742e-01	1.48435e-01	-4.56374e-01
[1461]	-1.21050e-02	-1.06081e-01	2.07405e-02	-2.73422e-01	-1.43112e-01
[1466]	-3.20715e-01	-1.72567e-01	-2.52670e-01	-8.16171e-02	-2.11371e-01
[1471]	-1.68513e-01	-2.11120e-01	-2.52439e-01	-2.57358e-01	-6.37195e-01
[1476]	-3.63340e-01	-1.98957e-01	-2.53065e-02	-1.50531e-01	-1.39318e-01
[1481]	-3.80359e-01	-3.69613e-01	-5.37596e-01	-5.69318e-01	-3.78839e-02
[1486]	-2.18257e-02	-2.59598e-01	-5.32544e-02	-1.78852e-01	-2.37992e-01

[1491]	-2.08462e-01	-7.34278e-02	-3.50878e-01	-4.64693e-01	-4.26961e-01
[1496]	-1.44262e-01	-5.01543e-01	-2.11457e-01	-2.42441e-01	-9.75591e-02
[1501]	-2.41873e-01	1.63861e-01	-2.16282e-01	-1.56194e-01	-1.36596e-01
[1506]	-2.39906e-01	-1.39741e-01	-6.04435e-01	-3.55049e-01	-3.12596e-01
[1511]	-3.20341e-01	-5.77724e-02	-2.28896e-01	-1.33341e-01	-3.99282e-01
[1516]	-2.72764e-01	-2.56959e-01	-5.50952e-02	-1.52389e-01	-1.51883e-01
[1521]	-9.75871e-02	-9.90708e-02	7.10441e-02	-5.88143e-02	5.27540e-02
[1526]	-3.58840e-01	-4.15372e-01	-4.42677e-02	-3.13832e-01	-2.64658e-01
[1531]	-5.77136e-01	-1.10143e-01	-3.69931e-01	-3.66067e-01	-3.01940e-02
[1536]	-3.96780e-01	-2.75096e-01	-1.66479e-01	-3.59299e-02	-1.81544e-01
[1541]	-8.73745e-02	-1.26055e-01	-1.46624e-01	-4.63894e-01	-4.49340e-01
[1546]	-2.38128e-01	2.69989e-02	-2.69566e-01	6.85623e-02	-3.81589e-01
[1551]	5.46419e-02	-8.75259e-02	-2.30923e-01	-2.91351e-01	-2.96170e-01
[1556]	-2.38134e-01	-1.44420e-01	-2.50204e-01	-2.34977e-01	-2.18567e-01
[1561]	-2.06349e-01	-3.60312e-01	-3.55159e-01	-4.49190e-01	-1.59619e-01
[1566]	-4.36617e-01	5.02446e-03	-5.81320e-02	-2.96128e-01	-3.90559e-01
[1571]	-2.42441e-01	-4.41192e-01	-9.53714e-02	1.33093e-02	-7.93114e-02
[1576]	-6.08771e-02	-2.88954e-01	-6.46564e-02	-2.06622e-01	-1.45263e-01
[1581]	-2.38876e-01	-2.40817e-01	-1.73725e-01	-4.24440e-01	4.34863e-02
[1586]	1.07824e-01	-3.34445e-01	-6.50139e-01	-5.74002e-02	-3.71217e-01
[1591]	-2.64631e-01	-3.15505e-01	-3.34516e-01	-2.98224e-01	-4.28854e-01
[1596]	-1.94859e-01	-6.30262e-01	-5.27503e-03	-1.72649e-01	-1.24311e-01
[1601]	-8.06432e-03	-5.59767e-02	-1.86598e-01	-2.75841e-01	-4.80648e-01
[1606]	-2.20936e-01	-1.88309e-01	-9.40498e-03	9.10415e-02	-2.45329e-01
[1611]	-2.34269e-01	-2.03106e-01	-1.51041e-01	-1.22203e-01	-1.73939e-02
[1616]	-2.07672e-01	-2.32326e-01	-1.38561e-01	-1.97657e-02	-2.79036e-02
[1621]	-2.31261e-01	-2.91613e-01	1.05441e-01	-2.27375e-02	-2.44956e-01
[1626]	-1.18134e-01	-2.81338e-01	-1.54762e-01	-7.41849e-02	-1.63721e-01
[1631]	-4.13570e-01	-2.85700e-01	-1.42934e-01	-3.96851e-02	-2.20996e-01
[1636]	-3.78902e-01	-4.26627e-01	-5.59050e-02	-2.50242e-01	-3.04358e-01
[1641]	-9.66276e-02	-2.05763e-01	-2.54709e-01	-1.73657e-01	-1.65305e-01
[1646]	-1.65962e-01	-2.79561e-01	-4.37516e-02	-5.28223e-02	1.82449e-02
[1651]	-4.54795e-01	-2.12803e-01	-1.51728e-01	-1.32994e-01	-1.52454e-01
[1656]	-5.21124e-01	-3.32544e-01	-1.78221e-01	-5.38228e-02	-4.54573e-01
[1661]	-2.13449e-01	-2.04123e-01	-1.87474e-01	-1.16024e-01	-1.98811e-01
[1666]	-2.82225e-01	-3.19604e-01	-2.15995e-01	-1.22056e-01	6.47902e-02
[1671]	-2.35450e-01	-1.38910e-01	2.55960e-02	-1.79630e-01	-1.07117e-01
[1676]	-3.75572e-01	-6.87866e-02	-4.62913e-01	-2.46592e-01	-5.76002e-02
[1681]	-2.90923e-01	-3.60421e-01	-1.88226e-01	-1.86604e-01	-3.79082e-01
[1686]	-5.02890e-02	-3.03065e-01	-2.76688e-01	-3.51655e-01	-2.68854e-01
[1691]	-1.43931e-01	-6.69970e-02	-8.47430e-02	2.62749e-02	-4.10190e-01
[1696]	-2.04260e-01	-2.58852e-01	-8.42569e-02	-2.44034e-01	1.08696e-01
[1701]	-7.57993e-02	-1.98765e-01	-2.70598e-01	-2.09726e-01	-2.94501e-01

[1706]	-2.29325e-01	-4.36038e-01	-1.89649e-01	3.80568e-02	6.85361e-02
[1711]	-1.75277e-01	-4.50639e-01	1.39797e-02	6.93395e-04	-4.45855e-01
[1716]	-1.28029e-01	-4.88826e-01	-1.96827e-01	3.74725e-02	8.01240e-03
[1721]	-4.53259e-01	-9.85353e-02	-5.36068e-01	-4.76442e-02	-1.43462e-01
[1726]	-1.49878e-01	-3.75966e-01	-3.17466e-01	-1.23926e-02	-2.37597e-02
[1731]	2.49897e-01	-9.77659e-02	-3.76350e-01	-2.82756e-01	-8.15445e-02
[1736]	-2.09948e-01	-1.17703e-01	-4.60172e-02	-1.96211e-01	-2.84516e-01
[1741]	-6.90463e-02	5.42693e-02	-3.36729e-01	-6.87647e-02	-1.72371e-01
[1746]	-9.12039e-02	-8.25473e-02	-4.69611e-01	-4.92828e-02	-3.45413e-01
[1751]	-3.89985e-01	-2.20517e-01	-2.27002e-01	-2.08168e-01	-2.69055e-01
[1756]	-1.60279e-01	-1.27936e-01	-3.58961e-01	-3.72960e-01	-1.60085e-01
[1761]	-2.78800e-01	-1.29484e-01	-2.28499e-01	-7.05090e-02	-2.45667e-01
[1766]	-2.57898e-01	-1.56812e-01	-2.05146e-01	-3.62864e-01	-1.93914e-01
[1771]	-2.56077e-02	-7.31469e-02	-3.55891e-01	-3.49050e-01	-1.14402e-01
[1776]	-6.31033e-02	-2.01923e-01	-3.02952e-01	-7.78503e-02	-4.45226e-01
[1781]	-3.55099e-01	-1.14442e-02	-2.53014e-01	-2.46085e-01	-6.64018e-01
[1786]	-2.62569e-01	-2.55162e-01	-9.65997e-02	-8.43032e-02	-3.05813e-01
[1791]	-1.59128e-01	-1.81556e-01	-2.49117e-01	-3.89799e-02	-2.26208e-01
[1796]	1.92885e-01	-1.91804e-01	-3.10540e-01	-4.97247e-01	-3.85762e-01
[1801]	-4.04535e-02	-6.68897e-02	-8.81375e-02	-1.90899e-01	-1.77231e-01
[1806]	-2.10192e-01	-2.36742e-01	-1.44588e-01	-2.98119e-01	-1.25774e-01
[1811]	-1.96246e-01	-3.03477e-01	-2.31176e-01	-3.30679e-01	-2.07528e-01
[1816]	-3.59084e-01	-4.49767e-01	-1.69373e-01	-3.75207e-01	-3.76425e-01
[1821]	-3.84140e-01	-6.24257e-01	-5.77303e-02	-1.45633e-01	-3.40139e-01
[1826]	-1.83143e-01	-2.32388e-01	-5.95697e-01	-3.84410e-01	-4.97342e-01
[1831]	-9.19650e-02	-2.00049e-01	-2.51323e-01	-3.82984e-01	-4.46500e-01
[1836]	-9.53920e-02	3.23442e-02	-3.70386e-02	-3.02435e-01	-4.06304e-01
[1841]	-1.72639e-01	5.32688e-02	1.76891e-02	-2.77221e-01	-3.59926e-01
[1846]	-1.89184e-01	-2.17806e-01	-2.17519e-01	-1.49635e-01	-9.96417e-02
[1851]	-1.06779e-01	-1.36291e-01	-1.10095e-01	-1.73733e-01	-1.88711e-01
[1856]	-5.43278e-02	-3.50396e-01	-7.91665e-02	-4.19465e-01	-2.42906e-01
[1861]	-1.65252e-01	-5.16445e-01	-1.19431e-01	-3.21930e-01	-1.92656e-01
[1866]	-6.05690e-02	-1.26009e-01	-2.03618e-01	-2.14207e-01	6.21574e-02
[1871]	-6.30105e-02	-1.40736e-01	-1.41422e-01	-1.47534e-01	-1.61483e-01
[1876]	-4.77871e-01	-3.13451e-01	-9.17849e-02	-6.57931e-02	-2.96039e-01
[1881]	-2.88765e-01	-8.08024e-02	-2.91658e-02	-2.17076e-01	-1.10888e-01
[1886]	-3.06321e-01	-4.74454e-01	-4.91984e-01	-1.65131e-01	-5.30792e-03
[1891]	-2.53813e-01	-1.50515e-01	-1.50054e-01	2.79647e-05	-3.62554e-01
[1896]	-2.39954e-01	-1.85751e-01	-4.18069e-01	-2.15534e-01	-2.45085e-01
[1901]	-1.45379e-01	-1.33136e-01	-4.35478e-01	-3.50999e-01	-2.51156e-01
[1906]	-2.06728e-01	-4.25744e-01	-3.92197e-01	-4.51160e-01	-2.33642e-02
[1911]	-1.23372e-01	-9.97448e-02	-6.29731e-02	-5.27343e-02	-8.49085e-02
[1916]	-7.51332e-02	-2.50627e-01	4.23634e-02	3.28652e-02	-1.12120e-01

```
[1921] -2.38904e-01 -1.32642e-01 -1.58383e-01 -1.74478e-01 -2.17698e-01
[1926] -6.45646e-02 -4.05984e-01 -1.33283e-01 -1.60329e-01 -4.47030e-01
[1931] -1.33841e-01 -1.44318e-01 -2.05262e-01 -6.36198e-02 1.22788e-02
[1936] -1.62921e-01 -3.87143e-01 -1.50322e-01 -5.04758e-01 -2.82829e-01
[1941] -3.72188e-01 -2.57449e-01 -1.71646e-01 -4.75980e-02 -1.17390e-02
[1946] -3.95734e-01 -3.54168e-01 -4.75139e-01 -1.00731e-01 -3.29236e-01
[1951] -2.34531e-01 -3.76212e-01 -8.75748e-02 -2.04454e-03 -4.37554e-01
[1956] -3.79320e-01 -1.04054e-01 -1.12676e-01 -8.79391e-02 -1.85449e-02
[1961] -5.07450e-01 -9.84385e-02 -3.60356e-01 -5.66678e-02 -2.83637e-01
[1966] -2.53144e-01 -2.26531e-01 -6.53249e-02 -2.32461e-01 -2.04347e-01
[1971] -8.82412e-02 -2.87504e-01 1.19440e-01 -5.01171e-01 7.89852e-02
[1976] -1.89273e-01 -2.64709e-01 -4.78704e-02 -3.27616e-01 -4.97026e-02
[1981] -2.92499e-01 -3.17740e-01 6.97201e-02 -7.09262e-02 -3.75979e-01
[1986] -2.16396e-01 -2.20399e-01 -2.42498e-01 -6.19238e-01 8.41162e-02
[1991] -3.19517e-01 -3.49050e-01 -5.12430e-01 -3.62675e-01 -3.35461e-01
[1996] 2.45077e-01 -3.23850e-01 -4.45263e-01 -2.22201e-01 -9.99990e-02
[2001] -1.32820e-01
```

```
post_MODU_G$beta$beta_temp_fruits
```

```
[1] -1.75674e-01 -1.87634e-01 -1.20222e-01 -1.86172e-01 -1.93513e-01
[6] -7.41158e-02 -2.54680e-01 -9.30569e-02 -2.14312e-01 1.12856e-01
[11] 2.25747e-01 2.72193e-02 -2.17763e-01 1.40656e-02 -3.26820e-01
[16] -3.33271e-01 -2.58640e-01 -3.78690e-01 -1.93660e-01 -1.38194e-01
[21] -1.03091e-01 9.77666e-02 -3.45110e-01 -3.64562e-01 -7.23454e-02
[26] -2.46815e-01 -2.30988e-01 -8.20353e-02 -9.52294e-02 -2.68177e-01
[31] -1.84090e-01 -1.90515e-01 -3.72623e-02 -1.64690e-01 1.18080e-01
[36] -1.95427e-01 -2.39598e-01 -3.23687e-01 -2.97511e-01 -2.87110e-01
[41] -4.61460e-02 7.81178e-02 3.23601e-03 -1.72142e-01 -1.85729e-01
[46] -4.73510e-02 4.37159e-02 -2.01661e-01 -4.37571e-01 7.58373e-02
[51] -3.34460e-01 -1.18832e-01 1.26393e-02 4.09478e-02 1.80857e-01
[56] -2.04298e-01 -1.23080e-01 -2.63553e-02 -8.06808e-02 1.82351e-03
[61] -3.24518e-01 -4.42292e-01 -1.45878e-01 -8.37702e-02 -4.46085e-01
[66] -2.53392e-01 -1.66203e-01 -2.07584e-01 -4.28449e-01 -2.68400e-01
[71] -3.24920e-01 -1.75551e-01 -2.50052e-01 -2.24910e-01 -2.42190e-01
[76] 7.70459e-02 -8.41773e-02 1.08836e-01 1.53930e-01 9.63862e-03
[81] -1.37012e-01 3.80881e-02 2.36669e-03 -2.34229e-01 7.78315e-02
[86] -1.15313e-01 -2.58933e-01 -1.06662e-01 -1.45925e-01 -7.19887e-02
[91] -4.39919e-01 -2.57786e-01 -2.15874e-01 -1.76600e-01 -8.71708e-02
[96] -1.75243e-01 1.45371e-01 -2.82592e-01 -1.34529e-01 -1.97668e-01
[101] -2.34920e-01 -2.05410e-01 -7.06126e-02 -3.61073e-01 -3.52419e-02
[106] -4.34276e-01 -2.12237e-01 -6.81587e-02 -3.03671e-01 -3.00213e-01
```

[111]	-4.01408e-01	-3.06993e-01	-7.02611e-02	-2.19168e-01	-5.92729e-02
[116]	-1.58129e-01	-1.02908e-01	-1.79646e-01	-9.42123e-02	-9.61505e-02
[121]	-1.46561e-01	-4.38683e-01	9.52427e-02	1.89671e-01	-3.19184e-01
[126]	-5.49459e-02	-1.38628e-01	-1.71052e-01	-2.87839e-01	-3.25572e-01
[131]	-2.13016e-01	-5.21563e-01	8.50784e-02	-2.72679e-01	-1.87128e-01
[136]	-2.56604e-01	-7.93753e-02	-1.35430e-01	-2.63654e-01	-1.60463e-01
[141]	-2.22134e-01	-7.61280e-02	-3.78853e-01	-2.87404e-01	-2.33796e-01
[146]	1.53271e-01	-6.45823e-03	-3.27419e-01	-2.02851e-01	-2.03138e-01
[151]	3.48988e-02	-8.44906e-02	-1.42713e-01	2.05916e-01	-5.30413e-01
[156]	-1.61644e-01	-1.96053e-01	-1.22900e-01	-2.29404e-01	5.29210e-02
[161]	-1.07947e-01	-1.30758e-01	-5.59992e-02	-1.70112e-01	-1.27259e-02
[166]	-2.96800e-01	4.80753e-02	-2.30384e-01	-1.60853e-01	-2.10293e-01
[171]	-1.09318e-01	-5.57135e-02	-4.91900e-01	-1.38620e-01	-2.03270e-01
[176]	-1.89185e-01	-2.04653e-01	-1.75023e-01	-2.02811e-01	-6.07705e-02
[181]	-1.77740e-01	-1.16398e-01	5.58373e-02	-1.14484e-01	-1.60851e-01
[186]	-2.33583e-01	-1.28907e-04	-3.50320e-01	-4.56204e-01	-3.00578e-01
[191]	-4.31040e-01	-1.51097e-01	-2.10693e-01	-2.05716e-01	-6.39089e-02
[196]	-2.34553e-01	-3.81480e-01	-1.38817e-01	-2.15572e-01	-2.58673e-01
[201]	-2.31991e-01	-1.31432e-01	-2.68410e-01	-1.39794e-01	-1.29642e-01
[206]	-8.36398e-02	-3.19996e-02	-2.15723e-01	-7.41249e-02	-1.68459e-01
[211]	-3.90441e-01	-9.37511e-02	-3.50062e-01	6.28821e-02	-3.75226e-01
[216]	-1.19429e-02	-6.01319e-02	-1.35080e-01	-1.00725e-01	-1.55159e-01
[221]	-1.55870e-02	-2.89535e-01	-1.95906e-01	-9.08730e-02	-2.21677e-01
[226]	-1.00471e-01	-1.92700e-01	-9.42253e-02	-1.99849e-01	-2.70654e-02
[231]	-2.53240e-01	-1.75269e-01	-1.42516e-01	-2.37642e-01	9.48817e-02
[236]	-2.85774e-01	-7.94611e-02	-1.34743e-01	2.31807e-02	-2.31707e-01
[241]	-2.95672e-01	-2.24449e-01	-5.22603e-02	-2.82721e-01	-3.28341e-01
[246]	-2.46992e-01	1.11060e-01	-4.39930e-02	-1.55122e-01	-2.98833e-01
[251]	-1.85694e-01	1.16549e-01	-1.26737e-01	-1.69849e-01	-2.67984e-01
[256]	-5.81163e-02	3.01096e-02	-3.43921e-01	-3.11181e-01	-1.58780e-01
[261]	-2.92067e-01	3.40072e-02	-3.13009e-01	-1.43560e-01	2.76160e-02
[266]	-2.18730e-01	-1.92183e-01	-1.33876e-01	-3.35170e-01	-2.92452e-01
[271]	2.89674e-01	-1.98420e-02	-2.56906e-01	2.86146e-02	-2.59362e-01
[276]	-2.62634e-01	-1.92896e-01	-1.98096e-01	-2.34733e-01	-2.80936e-01
[281]	-3.93199e-01	-2.68989e-02	-2.24993e-01	-2.02705e-01	-4.51882e-03
[286]	-1.03337e-01	-3.79143e-01	1.59748e-01	-2.67861e-01	8.39588e-02
[291]	-5.09966e-02	-2.13050e-01	-2.92433e-01	1.00544e-01	-1.74811e-01
[296]	-4.06177e-01	-3.38030e-01	-2.35332e-01	2.90375e-02	-1.81487e-01
[301]	-1.68162e-01	-6.52044e-02	-7.14975e-02	-2.59024e-01	8.58527e-02
[306]	-3.18734e-01	-1.48622e-01	-2.14255e-01	-2.09403e-01	-3.42819e-01
[311]	5.99007e-02	-2.67926e-01	-8.12435e-02	-2.27858e-01	-1.74209e-01
[316]	-1.14979e-01	-1.31338e-01	-1.05705e-01	-1.77432e-01	-2.34941e-01
[321]	-1.45856e-01	-3.34388e-01	-1.58538e-01	-2.71358e-01	-2.55221e-01

[326]	4.72328e-02	-3.99622e-01	-8.13835e-02	-1.86300e-01	-2.31810e-01
[331]	-9.58826e-02	-1.78191e-01	-9.10465e-02	-2.22215e-01	-3.59985e-01
[336]	-1.15304e-01	-3.26231e-01	-1.89230e-02	-1.83891e-01	-2.27718e-01
[341]	5.72550e-02	-1.13114e-01	-2.09528e-01	1.55228e-02	-1.59194e-01
[346]	5.30749e-02	7.45176e-02	-9.26246e-02	-2.10135e-01	4.45493e-02
[351]	1.54886e-01	-2.87126e-01	-2.87911e-02	-1.07019e-01	-2.81804e-01
[356]	-6.20875e-02	1.41565e-01	-3.08249e-01	5.80414e-02	2.82378e-01
[361]	-1.79785e-01	-1.96157e-01	-4.80178e-01	1.24052e-01	-1.76932e-01
[366]	1.86212e-02	9.29801e-02	2.71163e-02	-2.35396e-01	-2.74281e-01
[371]	-1.32918e-01	-2.51374e-01	-4.79378e-02	-3.98145e-01	-1.25577e-01
[376]	-6.29316e-02	-4.31308e-01	-2.36295e-01	-2.38698e-01	-2.91280e-01
[381]	-2.95061e-01	-2.53939e-03	-2.01898e-01	-1.80416e-01	-2.51789e-01
[386]	-2.96453e-02	-1.99078e-02	-3.24703e-01	3.10837e-02	-2.12199e-01
[391]	-1.46882e-01	-7.87896e-03	-3.87207e-01	-3.43659e-01	-8.66236e-02
[396]	-4.22061e-02	2.12548e-02	-2.14983e-01	-1.94624e-02	-6.67844e-02
[401]	-1.16949e-01	-3.95256e-02	1.78290e-03	1.84708e-02	-1.09482e-01
[406]	-3.41003e-01	-1.98384e-01	-9.98759e-03	-2.01402e-01	-2.41155e-01
[411]	8.80704e-02	-2.28305e-01	-8.41671e-02	-2.62180e-01	-8.80279e-02
[416]	2.16305e-02	4.47716e-02	-1.72367e-01	-2.53689e-01	-2.50978e-01
[421]	-2.18346e-01	-1.17525e-01	-1.08918e-01	-8.73380e-02	-2.62900e-01
[426]	-1.44400e-01	-1.06457e-01	-1.63221e-02	-1.16988e-01	-2.42942e-01
[431]	-9.41545e-02	8.38924e-02	-3.99090e-01	-2.59616e-01	3.32939e-02
[436]	1.53458e-01	-1.58756e-01	-1.52576e-01	-7.77786e-03	-3.79430e-01
[441]	-1.46252e-01	-2.29501e-01	-2.14188e-01	-8.75067e-02	-4.75528e-02
[446]	-2.02452e-01	-1.86175e-01	-6.03051e-02	-9.58897e-02	-2.12959e-01
[451]	-4.23178e-01	-1.64825e-01	-3.15164e-01	2.36085e-02	-3.73035e-01
[456]	-3.63571e-01	-4.34426e-01	-2.00210e-01	-7.00517e-02	-4.07159e-01
[461]	-3.41906e-01	-3.53156e-02	-3.12949e-01	-3.68720e-01	-1.29226e-01
[466]	-8.34563e-02	1.66710e-01	-4.56077e-01	3.51888e-02	-5.67241e-02
[471]	-1.68575e-01	-3.08327e-01	-7.61803e-02	-3.68800e-02	-1.55841e-01
[476]	-1.37932e-01	-1.86358e-01	-2.88389e-01	5.37174e-02	-7.13116e-02
[481]	-5.05115e-01	-3.28838e-01	-2.54720e-01	-9.17708e-02	-1.42338e-01
[486]	-1.84838e-01	-1.98239e-02	-3.68069e-01	-1.30653e-01	-2.29644e-01
[491]	-3.95664e-01	-1.56166e-01	2.53731e-02	-3.19359e-01	-2.83605e-03
[496]	-4.37146e-01	-4.08138e-03	-5.01153e-03	-2.21157e-01	-1.02313e-01
[501]	5.37088e-02	-1.42174e-01	-3.44107e-01	-1.23439e-02	-1.74547e-01
[506]	-1.81153e-01	-1.09738e-01	-1.27317e-01	-7.56273e-02	-8.93999e-02
[511]	-3.67370e-01	-1.02634e-01	-1.85783e-03	-3.77015e-01	-1.57307e-01
[516]	-1.76023e-01	-1.61515e-01	-1.57943e-01	-1.89822e-01	2.22311e-02
[521]	-2.17230e-01	-2.76274e-01	1.32697e-02	-2.67708e-01	-9.36176e-02
[526]	-2.03647e-01	-8.74408e-02	-3.25718e-01	-2.58543e-01	-2.22423e-01
[531]	-2.09462e-01	-4.17536e-01	-1.16045e-01	3.65052e-02	-2.06156e-01
[536]	-8.93245e-02	-1.17776e-01	-3.07285e-01	-5.86597e-02	-1.85998e-01

[541]	-2.20442e-01	-2.12414e-01	-1.88236e-01	-2.05084e-01	-2.87135e-01
[546]	-3.48615e-01	9.83290e-03	-4.55175e-02	-7.77071e-02	-2.95663e-01
[551]	-2.87371e-01	-1.97075e-01	-2.87917e-01	-7.71336e-02	-2.60899e-01
[556]	-1.67946e-01	-1.36673e-01	-4.88346e-01	-2.15489e-01	-1.48299e-01
[561]	-1.98193e-01	-1.37149e-01	-1.01913e-01	-2.16902e-01	-3.36489e-01
[566]	-3.72463e-01	-2.92814e-01	-1.53693e-01	6.53923e-03	-2.03102e-01
[571]	-1.76013e-01	-8.88203e-02	-1.74070e-01	-5.35048e-02	6.34544e-02
[576]	-2.19844e-01	-2.28139e-01	-3.93861e-01	-5.25503e-02	-1.63308e-01
[581]	-1.01233e-01	-6.89296e-02	-2.20598e-01	-6.64142e-02	-1.73733e-01
[586]	-9.05028e-02	-2.65160e-01	-1.71404e-01	-3.07704e-01	-6.35757e-02
[591]	-2.19683e-01	-7.50670e-02	-2.45891e-01	-2.65431e-02	-3.26743e-01
[596]	-8.17628e-02	-2.92429e-02	-2.30315e-01	-3.15283e-01	-2.88752e-02
[601]	-1.17816e-01	-2.85796e-01	-9.20525e-02	-2.38837e-01	-3.16586e-01
[606]	-1.52176e-01	-2.08323e-01	3.33090e-04	-6.56486e-02	-3.21704e-01
[611]	-1.07273e-01	-7.60329e-02	-1.73390e-01	-7.67258e-02	-6.89441e-02
[616]	-1.57324e-01	-2.81509e-02	-5.13557e-02	-2.85463e-02	-5.64530e-03
[621]	-1.50674e-01	-1.80936e-01	-3.75771e-01	-1.84544e-01	-1.96328e-01
[626]	-1.03324e-01	-8.66136e-02	-5.14333e-02	-1.85479e-01	-1.82590e-01
[631]	-8.70512e-02	-3.42419e-01	-2.86271e-01	-2.54950e-01	-1.12306e-01
[636]	-3.04077e-01	7.66101e-02	-8.02901e-02	1.62712e-01	-2.36354e-01
[641]	-2.97578e-01	-8.59981e-02	-1.64064e-01	-3.25276e-01	-9.41399e-02
[646]	-1.04171e-01	-7.40944e-02	-3.14553e-01	-6.97474e-02	-1.59473e-01
[651]	7.04731e-02	-2.86143e-01	-1.11792e-01	-3.78820e-02	-3.02945e-01
[656]	-3.45739e-01	-2.38770e-01	-1.07906e-01	-3.02529e-01	-1.45954e-01
[661]	-2.97098e-01	-3.70819e-01	-2.07770e-01	-4.12825e-02	-1.33793e-01
[666]	-3.09546e-01	-1.67543e-01	-1.51258e-01	-2.66162e-01	1.18001e-02
[671]	-1.73492e-01	-3.04930e-01	-1.29613e-01	-2.20418e-01	-7.27008e-02
[676]	-2.08020e-01	-1.87079e-01	-1.40158e-01	-3.02519e-01	5.40978e-03
[681]	-2.17472e-01	-1.40260e-01	-3.41092e-01	-2.69697e-01	-2.86300e-01
[686]	-3.29619e-01	-3.11685e-01	-2.73296e-01	-2.56097e-01	-2.50769e-01
[691]	-7.06822e-02	-1.84942e-01	-1.42875e-01	-1.76380e-01	-4.71990e-01
[696]	6.86820e-02	-6.01149e-01	-5.82856e-01	-2.52087e-01	5.76519e-02
[701]	-1.61048e-01	-8.11729e-02	-7.11615e-02	6.99141e-02	4.84708e-02
[706]	-3.47740e-01	-2.19986e-01	1.27991e-01	-1.99029e-01	-2.58880e-01
[711]	-2.78702e-01	-2.37371e-01	-9.75523e-02	-5.81101e-03	-4.11813e-02
[716]	-1.70859e-01	-3.30600e-01	-2.96069e-01	1.07783e-02	-5.08492e-02
[721]	-1.17991e-01	-1.26243e-02	-1.81111e-01	-1.03428e-01	-8.30311e-02
[726]	-3.38731e-01	-2.78015e-01	-2.13112e-01	-1.62194e-01	-2.84847e-01
[731]	-2.03715e-01	-6.31522e-02	-3.16127e-01	5.24133e-02	-3.78287e-02
[736]	-1.15951e-01	-1.92519e-01	-3.99063e-01	2.75331e-02	-9.72948e-02
[741]	-1.71942e-01	-1.79353e-01	-3.55576e-01	-8.95604e-02	-2.12400e-01
[746]	2.94533e-02	-2.38091e-01	-3.53919e-02	-1.39297e-01	-3.48536e-02
[751]	-1.66454e-01	-1.33411e-01	-5.43550e-02	-1.32540e-02	-7.59796e-02

[756]	-2.14541e-01	-2.93298e-01	-5.39990e-01	-2.17152e-01	-2.26525e-01
[761]	-4.38977e-01	-3.42482e-01	-4.39954e-02	-2.76307e-01	-1.98464e-01
[766]	-8.89744e-02	-2.33033e-01	-1.59066e-01	-1.88195e-01	-2.65077e-01
[771]	-2.52896e-01	-1.54221e-01	-1.73373e-01	-6.28253e-02	-3.63108e-01
[776]	-1.89781e-01	1.38416e-01	-1.93465e-01	-9.60859e-03	-1.42246e-01
[781]	2.25074e-03	-1.51812e-02	6.83297e-03	-5.21188e-02	9.55890e-02
[786]	1.41751e-01	-1.01654e-01	-2.34917e-01	-1.94909e-01	-5.84906e-02
[791]	-6.19807e-02	-1.18695e-02	-2.84526e-01	-1.54935e-02	-3.39569e-01
[796]	8.01848e-02	-1.83179e-01	-3.02674e-01	-1.52113e-01	-1.68237e-01
[801]	-3.04741e-01	-8.92510e-02	-1.27766e-01	-1.71718e-02	5.79733e-02
[806]	-1.30559e-01	-2.84772e-01	-5.09829e-01	-1.79067e-01	-3.09537e-01
[811]	-5.61413e-02	-4.75628e-01	-2.18004e-02	-1.51379e-01	-7.29359e-03
[816]	-1.29135e-03	-4.38800e-02	-2.69853e-01	-1.81694e-01	-2.38667e-01
[821]	-1.31724e-01	-4.63309e-02	-3.02283e-01	-3.13872e-01	-5.44939e-02
[826]	-3.01395e-01	-5.20327e-01	-2.23569e-01	-2.90762e-01	2.01371e-01
[831]	-5.27023e-02	-4.08158e-01	-1.72561e-01	-2.28395e-01	3.74880e-03
[836]	-1.14677e-01	-1.92365e-01	-1.85576e-01	3.73683e-02	-2.33856e-03
[841]	-8.85363e-02	-1.27462e-01	-1.38813e-01	-1.75999e-02	-2.52730e-01
[846]	-1.45519e-01	-6.87551e-02	-6.45625e-02	7.60345e-03	-1.06050e-01
[851]	-1.05058e-01	-1.63247e-01	7.73719e-02	-5.64765e-02	-2.35896e-01
[856]	-1.72752e-01	-5.27168e-01	-3.70601e-01	-1.11334e-02	-4.20545e-01
[861]	-1.95735e-01	-1.62032e-02	-1.92109e-01	-1.24348e-01	-2.93118e-01
[866]	-2.09163e-01	6.67217e-02	-3.85600e-01	-2.80430e-01	-3.89269e-01
[871]	-2.79720e-01	-1.10420e-01	-2.73407e-01	-3.65868e-01	-1.98690e-01
[876]	-2.41669e-01	-1.45964e-01	2.29032e-01	-4.26952e-01	-1.86502e-01
[881]	-1.21212e-01	-1.20471e-01	-8.85939e-02	-1.85636e-01	2.18375e-02
[886]	9.28184e-03	1.42447e-01	-4.18160e-02	-1.24036e-01	-3.98948e-01
[891]	-1.00928e-01	-6.66000e-02	-2.46162e-01	-2.18371e-01	-5.63431e-02
[896]	-2.94648e-01	-1.99275e-01	-1.92702e-01	4.08177e-02	-1.57748e-01
[901]	-1.25422e-01	-2.54388e-01	-3.26474e-01	1.00130e-01	-2.27337e-01
[906]	-2.06752e-01	6.81413e-02	4.49844e-02	-1.90488e-01	-1.34453e-01
[911]	4.77480e-02	3.82833e-02	-5.17727e-02	-2.23533e-01	-6.94370e-04
[916]	-9.93636e-02	-1.58381e-01	-2.50342e-01	-3.84552e-01	-1.07016e-01
[921]	-2.76088e-01	-1.40398e-01	-4.34259e-02	-2.55589e-01	1.94204e-02
[926]	6.50248e-02	-1.47232e-01	-9.03089e-02	-1.62739e-01	-1.83674e-01
[931]	-1.83569e-01	-1.03458e-01	-1.41242e-01	-1.98152e-01	-2.43220e-01
[936]	-2.06031e-01	-2.46348e-01	-4.20156e-02	-2.18943e-01	-3.36608e-01
[941]	-2.63403e-01	-1.83271e-01	-3.02784e-02	-3.40512e-01	-4.45017e-02
[946]	-2.75036e-01	-1.22601e-01	-2.85051e-01	-1.61354e-01	-2.33994e-01
[951]	-1.66832e-01	-2.35747e-01	-1.71413e-01	-3.05606e-02	-9.83553e-02
[956]	-3.10166e-02	-6.35011e-02	-3.81203e-02	-2.14351e-01	-2.95362e-01
[961]	1.95315e-01	-2.89896e-02	-1.43784e-01	-1.61412e-01	-5.80624e-02
[966]	-3.55407e-02	-1.32807e-01	-1.06910e-01	-2.62960e-01	-7.21262e-02

[971]	-5.44211e-01	-1.31669e-01	-4.27999e-02	-6.91206e-02	-2.05561e-01
[976]	-1.77539e-01	-2.14998e-01	-3.96780e-01	-1.71337e-01	-4.79557e-02
[981]	-3.65596e-02	-1.78494e-01	-2.75742e-01	-2.21081e-01	-2.97134e-01
[986]	-1.69998e-01	-1.15670e-01	-8.10051e-02	-2.53402e-01	-4.16825e-01
[991]	-7.22760e-02	-6.22349e-02	-3.70104e-01	-3.74733e-01	2.00647e-02
[996]	-8.20301e-02	-3.39819e-02	-2.17296e-01	-1.19688e-01	-3.64419e-01
[1001]	-1.05136e-01	-9.84589e-02	-1.76784e-01	-1.08592e-01	-4.29450e-01
[1006]	-3.49695e-01	-2.98459e-01	-2.93288e-01	-3.24422e-01	-4.23922e-01
[1011]	5.27660e-02	-1.37199e-01	-2.51104e-01	-1.50574e-01	-3.08126e-01
[1016]	-1.37229e-01	-2.47823e-01	-4.47494e-01	-2.06441e-01	-1.28329e-01
[1021]	-2.80221e-01	-4.52002e-02	-9.04939e-02	-2.05456e-01	-1.42806e-01
[1026]	-2.20921e-01	-4.09522e-02	-1.25864e-01	-1.33799e-01	-2.58173e-01
[1031]	-1.01787e-01	-7.88850e-02	-1.31929e-02	-1.07231e-01	-1.33841e-01
[1036]	-1.30968e-01	-1.27594e-01	-6.17959e-02	-2.38334e-01	-1.40422e-01
[1041]	1.76934e-01	-6.62153e-02	4.86347e-02	-2.88527e-01	-7.96487e-02
[1046]	-7.71002e-02	-1.44695e-01	-2.33484e-01	-1.13939e-01	-1.77378e-01
[1051]	-2.61440e-01	5.44300e-02	-1.15171e-01	-1.03887e-01	-2.04553e-01
[1056]	9.13367e-02	-1.87688e-01	-3.41188e-01	-5.32125e-02	-2.45121e-01
[1061]	-5.71700e-02	-7.08650e-02	4.94334e-02	-1.23049e-02	-2.99283e-01
[1066]	-2.78247e-01	-2.71482e-01	-2.08853e-01	-1.67184e-01	-2.79506e-01
[1071]	-1.64547e-01	-1.59104e-01	-3.76297e-01	4.48327e-02	-2.35104e-01
[1076]	-6.68845e-03	-1.32406e-01	-2.95315e-01	7.37567e-02	-3.38078e-02
[1081]	3.11628e-02	1.59585e-02	-1.33303e-01	-1.75836e-01	-2.63212e-01
[1086]	-1.65918e-01	-2.87011e-01	-5.15181e-01	-1.44079e-01	-1.35093e-01
[1091]	-2.63372e-01	-3.17532e-01	-2.16987e-01	-5.24846e-02	-1.31631e-01
[1096]	-1.55581e-02	-2.94829e-01	-5.17831e-02	-2.74926e-01	-1.65581e-01
[1101]	-2.10390e-01	-2.03228e-01	-1.76175e-01	4.71915e-02	-1.94036e-01
[1106]	-4.72092e-02	-1.55204e-01	-1.20314e-01	-1.28801e-01	-1.78503e-01
[1111]	4.89736e-02	-1.20854e-01	-2.58277e-01	-2.69832e-01	-9.77206e-02
[1116]	-5.34351e-02	-1.86749e-01	-4.69039e-01	-3.59320e-01	-3.18170e-01
[1121]	-1.33008e-01	-4.57524e-01	-4.32822e-01	-1.60943e-01	-8.20252e-02
[1126]	-1.14526e-01	-8.84217e-02	-1.90644e-01	-3.22051e-01	-2.65759e-01
[1131]	-1.52853e-01	4.43035e-02	-1.49298e-01	-3.51114e-01	-2.10496e-01
[1136]	-2.77559e-01	4.06911e-02	-2.11912e-01	-4.35604e-01	-3.32379e-01
[1141]	-1.03410e-01	-1.07605e-01	-2.30668e-01	-6.63071e-02	-1.72975e-01
[1146]	-3.75660e-01	-2.58461e-02	-3.27738e-01	-1.46460e-01	-1.86790e-01
[1151]	-2.52299e-01	-2.22910e-01	-8.32497e-02	-1.29061e-01	-2.44373e-02
[1156]	-8.71686e-02	-1.42864e-01	-1.93859e-01	-4.63709e-01	-3.43172e-01
[1161]	-1.83144e-01	-1.59883e-01	-7.26626e-02	-1.82593e-01	-4.24533e-01
[1166]	-3.34436e-01	6.15831e-02	-2.76733e-01	-2.65798e-01	-1.78635e-01
[1171]	1.37170e-01	-3.13931e-01	-1.78144e-02	-1.76542e-01	-1.72882e-01
[1176]	-3.67153e-01	-2.23705e-01	6.55462e-02	2.89571e-02	-6.22048e-03
[1181]	-1.94696e-01	6.29261e-02	-3.49939e-01	-4.90273e-02	7.09344e-02

[1186]	-3.09154e-01	-2.07401e-01	-1.46437e-01	-2.90237e-02	-1.48776e-02
[1191]	-3.22878e-01	-1.87128e-01	-1.45058e-02	1.62108e-01	-1.55544e-01
[1196]	3.84844e-02	-2.53137e-01	-1.13357e-01	-2.85067e-01	-7.52142e-02
[1201]	-2.20669e-01	-1.01145e-01	-2.71688e-01	-9.78212e-02	-1.04663e-01
[1206]	-3.38002e-01	-1.12630e-01	-1.54878e-01	-8.77727e-02	-4.51960e-02
[1211]	-2.09293e-01	2.31923e-02	-1.09249e-01	4.38146e-02	-2.79438e-01
[1216]	1.04778e-02	-3.16419e-01	-1.92573e-01	-4.31654e-02	-1.12739e-01
[1221]	1.54710e-02	-3.55737e-01	5.54785e-03	1.42006e-01	-3.02570e-01
[1226]	-1.40698e-01	-1.38466e-01	1.17300e-01	-2.35069e-01	-2.95996e-01
[1231]	-8.82083e-03	-1.45133e-01	-1.68471e-02	6.07634e-02	3.46268e-02
[1236]	-5.50464e-01	-2.32813e-01	-3.76173e-01	-3.74855e-01	-3.38900e-01
[1241]	-4.19795e-01	-1.55012e-01	-3.57309e-01	-3.50278e-01	-7.81175e-02
[1246]	-2.05319e-02	2.09120e-02	-3.66866e-01	-1.45326e-01	3.85529e-02
[1251]	-9.59112e-02	-4.09069e-01	-9.04093e-02	-1.42069e-01	-3.19941e-02
[1256]	-2.09240e-01	4.04884e-02	3.83285e-03	-3.13930e-01	-1.57083e-01
[1261]	-5.92735e-02	9.23291e-02	-1.86077e-01	4.06489e-02	-1.85944e-01
[1266]	-4.77808e-02	-1.16899e-01	-3.49829e-01	-1.63912e-01	-1.20680e-01
[1271]	-1.94227e-01	1.16429e-01	3.89615e-02	-1.94833e-01	-4.36772e-02
[1276]	-2.04641e-01	-2.11950e-01	-2.32487e-01	-9.90477e-02	-3.47599e-01
[1281]	-2.35611e-01	-7.61269e-02	-2.45937e-01	-1.65856e-01	-2.15646e-01
[1286]	-6.10016e-02	1.98622e-02	-6.82172e-02	-1.11124e-01	-2.21987e-01
[1291]	-7.78455e-02	-1.90926e-01	-4.02156e-01	-6.06272e-02	-1.52306e-01
[1296]	-8.11724e-02	-2.53344e-01	-8.15067e-02	-2.17494e-01	-3.69036e-01
[1301]	-2.20338e-01	5.02223e-02	1.53488e-02	-1.58166e-01	-1.26385e-01
[1306]	3.57203e-02	-5.66417e-01	-2.19144e-01	1.16538e-02	1.43269e-01
[1311]	-1.75939e-01	-2.37918e-01	-1.47754e-02	3.06157e-02	-5.12548e-03
[1316]	-5.09435e-02	-1.82963e-01	-5.30132e-02	-3.04695e-01	-1.92990e-01
[1321]	-6.19357e-01	-1.72850e-01	-9.19308e-02	-2.05761e-01	-2.40040e-01
[1326]	3.83872e-02	-8.14396e-02	-2.61203e-01	-2.60084e-02	-1.40967e-01
[1331]	-7.81595e-02	-9.88818e-02	-5.27720e-02	1.72817e-02	-2.55364e-01
[1336]	-3.85445e-02	4.08679e-02	-1.93022e-01	-1.64950e-01	-7.71153e-03
[1341]	-8.33814e-02	-3.82476e-01	-2.64828e-01	-1.28256e-01	-3.95656e-01
[1346]	-4.09150e-01	-3.90523e-01	-6.97453e-02	-4.28785e-01	-2.82325e-01
[1351]	-2.52732e-01	-2.11120e-01	-1.04987e-01	-1.76910e-02	-1.91480e-01
[1356]	-9.36254e-02	-7.38380e-02	-2.14988e-01	-1.35499e-01	-2.67687e-01
[1361]	-1.72463e-01	-1.95728e-01	-9.49537e-02	-1.46884e-01	-1.25719e-01
[1366]	-3.54637e-01	-2.45179e-01	-3.34200e-01	-1.16708e-01	-3.84156e-01
[1371]	-1.97271e-01	-1.20545e-01	-5.97869e-02	-2.10277e-01	-1.41962e-01
[1376]	1.11590e-01	-6.27493e-02	-1.21777e-01	-1.14281e-02	-2.02620e-01
[1381]	-2.23624e-01	-2.31092e-01	-2.69979e-01	1.18227e-02	-3.05849e-02
[1386]	-7.15395e-02	-8.74175e-02	-2.57294e-01	-3.41872e-01	-1.61730e-01
[1391]	3.81169e-02	-2.08269e-01	-1.43552e-01	-4.31431e-01	-9.80117e-02
[1396]	-2.85940e-01	-6.82817e-02	-2.02120e-01	-1.16666e-01	-1.00260e-01

[1401]	-1.46133e-01	-4.06092e-02	-2.99675e-02	-2.52357e-01	-1.34991e-01
[1406]	-1.55452e-01	-3.69848e-02	-7.10419e-02	-3.05505e-01	-1.04365e-01
[1411]	-3.17158e-02	-7.53299e-02	-1.44016e-01	4.15174e-02	-1.31924e-01
[1416]	-2.25595e-01	1.34403e-02	6.93531e-02	7.01249e-02	-5.15379e-02
[1421]	-2.02001e-01	-7.62981e-02	-1.92834e-01	-6.56669e-02	-1.03472e-01
[1426]	-1.60358e-01	-2.31816e-01	-9.53053e-02	-4.63089e-01	-3.07132e-01
[1431]	1.62773e-03	1.93173e-01	4.31709e-02	-2.82963e-01	-3.82308e-02
[1436]	-1.89820e-01	-4.82157e-01	-3.37121e-01	6.39459e-02	4.24533e-02
[1441]	-1.09005e-01	-2.09916e-01	-3.15678e-01	-9.35563e-02	-4.74106e-02
[1446]	-1.48288e-01	-1.16082e-01	-1.95938e-01	-3.30473e-01	-1.69478e-01
[1451]	-1.40244e-01	-3.90383e-01	-2.00264e-02	-2.86872e-01	-5.80898e-02
[1456]	-2.68140e-03	-2.63167e-01	-1.63034e-01	-2.28487e-01	-3.23986e-01
[1461]	-2.05000e-01	-2.77694e-01	-2.15906e-01	-8.04918e-02	-2.08748e-01
[1466]	-1.41008e-01	-2.89364e-01	-1.02418e-01	-1.22159e-01	-1.39988e-01
[1471]	-1.82461e-01	-3.07573e-02	-2.62226e-01	-1.21983e-01	-2.62095e-01
[1476]	-1.50507e-01	-2.48285e-01	5.53408e-02	-3.48919e-01	-9.94600e-02
[1481]	-1.58840e-01	-2.33956e-01	-2.47490e-01	-3.70044e-01	-2.40096e-01
[1486]	-1.78881e-01	-5.54348e-02	1.00380e-02	-2.12678e-01	-2.82728e-01
[1491]	-1.67118e-01	-1.57230e-01	-3.24903e-01	-2.24016e-01	-7.59042e-02
[1496]	2.73944e-01	5.22903e-02	-8.82647e-02	-2.30952e-01	-3.07543e-01
[1501]	-5.90408e-04	-1.20938e-01	-6.27102e-02	-7.85962e-02	4.83900e-02
[1506]	2.28027e-01	-1.69884e-01	-1.19431e-01	-8.57802e-02	-2.85951e-01
[1511]	-2.85931e-01	2.02741e-02	-1.21170e-01	-3.65860e-01	-3.30521e-01
[1516]	-1.80951e-01	-1.26376e-01	-2.08517e-01	1.65726e-02	9.16284e-02
[1521]	-2.43221e-01	-2.67800e-01	-1.90129e-01	1.10243e-01	-1.59222e-01
[1526]	-2.03300e-01	-1.76206e-01	-1.68852e-01	-1.59752e-01	-7.15727e-02
[1531]	-2.86140e-01	-1.89689e-01	-1.02797e-01	-2.06631e-01	1.33363e-01
[1536]	2.83286e-02	-1.06589e-01	-1.43790e-01	1.90487e-03	-1.47256e-01
[1541]	-6.69484e-02	-1.80161e-01	-2.54225e-02	-1.93554e-01	-7.20611e-02
[1546]	-2.30800e-01	-2.19332e-01	-1.93609e-01	-1.51684e-01	-3.48315e-01
[1551]	-8.55994e-02	-1.89575e-01	-1.98757e-01	-2.51766e-01	-1.19850e-01
[1556]	-4.23244e-01	-1.23734e-01	-1.46145e-01	7.34179e-02	-1.82090e-01
[1561]	-3.13725e-01	1.56701e-02	-7.63361e-02	-1.20866e-01	-1.21492e-01
[1566]	-3.40803e-01	-2.59656e-01	-9.11776e-02	-1.01406e-01	-1.98968e-01
[1571]	-1.57339e-01	-5.07886e-01	-1.70943e-02	-1.88328e-01	-2.29219e-01
[1576]	-5.69029e-02	-1.45943e-01	-1.48487e-01	-1.64668e-01	-1.97540e-01
[1581]	-1.23985e-01	-2.56297e-01	-4.33520e-01	-5.84649e-01	-2.54675e-01
[1586]	2.28445e-01	-2.08133e-01	-2.10298e-01	-3.64672e-01	2.26691e-02
[1591]	5.75313e-02	-5.32258e-02	-6.05797e-02	-1.69778e-01	-1.02393e-01
[1596]	2.17144e-02	2.52882e-02	-2.32765e-01	-2.51980e-02	-2.91522e-01
[1601]	9.96584e-02	-1.00557e-01	-1.27364e-01	-2.40407e-01	-8.45160e-02
[1606]	-8.88779e-02	-2.84708e-01	-1.13151e-01	-1.15805e-01	4.61827e-02
[1611]	-1.66646e-02	-1.63443e-01	-1.65779e-01	8.03731e-02	5.34348e-03

[1616]	-7.08215e-02	-4.57622e-02	-2.16507e-01	-1.16033e-01	7.56207e-02
[1621]	-1.95481e-01	-1.03117e-01	-3.06902e-01	-4.18101e-01	-9.16244e-02
[1626]	-4.53043e-01	-1.40894e-01	-1.67570e-01	-1.20932e-01	-2.58910e-01
[1631]	-3.72490e-01	-3.57164e-01	-8.08282e-02	-2.75067e-01	-3.97011e-01
[1636]	3.87130e-02	-2.45617e-01	-1.90643e-01	-1.66466e-01	-8.49713e-02
[1641]	-3.42758e-01	-2.12481e-01	-1.13593e-01	-3.27257e-01	-1.01319e-01
[1646]	-6.80811e-02	-2.81343e-01	-2.88480e-01	-3.21073e-01	-6.17031e-02
[1651]	-9.98247e-02	-1.09859e-01	-1.13286e-01	-1.44397e-01	-3.50810e-01
[1656]	-3.94310e-01	-1.81916e-01	-1.77394e-01	-2.24383e-01	-1.62657e-01
[1661]	-4.38015e-02	-5.21918e-02	-1.27121e-01	-1.02668e-01	-3.19829e-01
[1666]	-8.48697e-03	2.36797e-02	-2.09359e-01	1.47829e-02	-2.14941e-01
[1671]	3.87736e-02	7.88174e-02	-1.47642e-01	-1.89772e-01	-3.82741e-01
[1676]	-9.95129e-02	-7.96200e-02	5.29348e-02	-6.42003e-02	-2.47868e-01
[1681]	-7.30537e-02	-9.49274e-02	-1.45608e-01	3.48199e-02	-5.49173e-02
[1686]	8.59578e-02	-2.12324e-01	-2.14515e-01	-2.23642e-01	-2.24858e-01
[1691]	2.66736e-02	1.41769e-01	-3.01143e-01	-1.71890e-01	-1.01296e-01
[1696]	-9.37882e-02	7.93614e-02	-2.45844e-01	8.39236e-02	-3.75302e-02
[1701]	-1.49010e-01	-2.43955e-01	-1.81890e-01	-2.94125e-01	-9.48488e-02
[1706]	4.36057e-02	-3.25013e-02	-1.75867e-03	-1.55363e-01	-1.31609e-01
[1711]	-2.20343e-01	-4.27128e-02	-1.03750e-01	-2.33411e-01	-1.03446e-01
[1716]	-3.01020e-01	-2.32823e-01	-2.59660e-01	-3.26960e-01	-2.34122e-01
[1721]	-4.33262e-01	-3.30803e-01	-2.47378e-01	-3.24080e-01	-2.11883e-01
[1726]	-2.31020e-01	-1.50476e-01	-1.27410e-01	-1.17153e-01	-3.65471e-01
[1731]	-1.58651e-01	-1.15598e-01	-2.12639e-01	-2.02808e-01	-1.04577e-01
[1736]	-2.23436e-01	-1.86371e-01	-8.14642e-02	-8.75562e-02	-4.45622e-01
[1741]	-2.21187e-01	-1.90021e-01	-3.73906e-02	-2.99377e-01	2.00976e-02
[1746]	-2.03372e-01	-1.49166e-01	-2.68889e-01	-1.20698e-01	-2.66160e-01
[1751]	-9.47054e-02	-1.62310e-01	-4.76610e-02	-7.70711e-02	7.59749e-02
[1756]	-3.24411e-01	-1.01974e-01	9.87542e-02	-2.64375e-02	-6.68193e-02
[1761]	-7.02292e-02	-4.62010e-01	-2.68346e-01	-3.71926e-01	-3.09592e-01
[1766]	-3.26505e-01	-1.76320e-01	-4.99534e-02	-9.72301e-02	-1.85344e-01
[1771]	-1.80582e-01	-1.34985e-01	-7.87430e-02	9.94988e-02	-1.42825e-01
[1776]	1.73212e-02	-7.64372e-02	-1.70217e-02	-1.66025e-01	2.25283e-02
[1781]	-3.98061e-02	-4.96903e-02	-1.59408e-01	-1.96614e-01	-2.98290e-01
[1786]	-2.40629e-01	-3.23583e-01	-3.63939e-01	-1.46564e-01	1.23769e-01
[1791]	-8.02191e-02	-3.33055e-03	-1.50063e-01	-1.65635e-01	-3.37044e-01
[1796]	1.26847e-01	1.43204e-01	-7.85278e-03	-1.58699e-01	-2.60186e-01
[1801]	-1.40556e-01	1.47704e-02	-2.43180e-01	-2.13604e-01	-1.31147e-01
[1806]	-3.50374e-01	-3.25209e-01	-1.43582e-01	-2.03693e-01	-2.85266e-01
[1811]	-4.36035e-02	-9.28712e-02	-2.39265e-01	-1.49933e-01	-6.50257e-02
[1816]	1.22549e-02	-2.08878e-01	-3.04554e-01	3.48591e-02	6.27914e-02
[1821]	-8.40122e-02	-3.21171e-01	-2.02614e-01	-1.34715e-01	-3.57210e-01
[1826]	-2.08021e-01	-1.60762e-01	-1.22784e-01	-1.91310e-01	2.65820e-02

```
[1831] -1.90183e-01 -1.17191e-01  5.41842e-02 -1.50834e-01 -5.09774e-01
[1836] -1.41421e-01 -2.98688e-01  6.31351e-02  2.12670e-02 -2.14315e-01
[1841]  9.18870e-02 -1.14656e-01 -1.18761e-01 -9.55841e-03 -5.67140e-02
[1846] -1.64865e-01 -9.67768e-02 -1.40919e-01 -1.52619e-01 -2.32979e-01
[1851] -1.16720e-01 -2.17666e-01 -2.61565e-01 -9.08208e-02 -1.48438e-01
[1856] -5.18892e-02 -4.32610e-02 -2.10994e-01 -1.92226e-01 -3.24465e-01
[1861] -1.36051e-01 -1.76156e-01 -3.37232e-01 -1.36178e-01 -2.09116e-01
[1866] -2.34001e-01 -2.09757e-01 -1.30083e-01 -8.02852e-02 -2.93540e-01
[1871] -2.40333e-01 -1.62521e-01 -2.51232e-01 -2.36622e-06 -2.22499e-01
[1876] -3.29315e-01 -4.52831e-01 -7.72316e-02  3.82780e-02 -2.33184e-01
[1881] -1.72136e-01 -8.39479e-02 -2.14319e-01 -1.53087e-01 -5.79276e-02
[1886]  1.31112e-01  2.30229e-02 -1.56568e-02 -1.58253e-01 -1.31492e-01
[1891]  4.59315e-02 -1.07180e-01 -2.75719e-02  5.58737e-02 -3.44130e-01
[1896] -1.16576e-01 -2.64165e-02 -8.18250e-02 -3.07536e-01 -2.73162e-01
[1901] -2.17533e-01 -1.47231e-01 -1.73214e-01  1.41589e-01 -1.19422e-01
[1906] -2.60776e-01 -9.96826e-02  1.35603e-01 -2.06017e-01  6.93023e-02
[1911] -3.13642e-01 -2.49293e-01 -3.03552e-01  1.66271e-01  7.12680e-02
[1916] -1.60762e-01 -1.22695e-01 -4.99076e-01 -6.12712e-02 -2.13313e-01
[1921] -2.69600e-01 -1.88121e-01 -2.11059e-01 -1.44229e-01 -1.48338e-01
[1926] -4.72611e-01 -4.95691e-02 -1.57585e-01 -1.28406e-01 -2.27576e-01
[1931] -1.89271e-01 -3.22849e-01 -4.71996e-02 -2.84869e-01 -1.50686e-01
[1936] -2.64318e-02 -1.09284e-01 -1.94027e-01 -1.74826e-01 -1.35384e-01
[1941] -1.49765e-01 -3.75659e-01 -1.55324e-01 -2.45038e-01 -3.25081e-01
[1946] -2.92129e-01 -3.52391e-01 -1.77711e-01 -1.48486e-01 -7.43301e-02
[1951] -2.56823e-01 -5.47349e-02 -4.05587e-01 -1.53823e-01 -1.41424e-01
[1956]  3.34940e-02 -1.91412e-01 -3.15042e-01 -1.71803e-01 -6.63010e-02
[1961] -2.28490e-01  2.88846e-02 -2.07232e-01 -1.17195e-01 -7.77013e-02
[1966] -1.18020e-01 -9.60905e-03 -2.24141e-01  1.36911e-01 -7.71617e-02
[1971] -8.38892e-02 -1.01269e-01 -2.72148e-01 -2.37252e-01 -1.82429e-01
[1976] -9.48102e-02 -2.85401e-01 -1.22260e-01 -4.04780e-01 -1.89803e-01
[1981]  7.74606e-02 -3.57602e-01 -7.68102e-02 -2.02632e-01  7.03885e-02
[1986] -1.47621e-01 -2.92994e-01 -9.12661e-02 -1.21240e-01 -1.81724e-01
[1991] -3.53125e-02 -1.03905e-01  5.36960e-02 -3.32209e-01 -5.58959e-01
[1996] -2.35555e-01 -3.17154e-01 -1.37098e-01 -1.83541e-02 -1.29588e-01
[2001] -3.60167e-01
```

#### `betas_MODU_GFFECT`

```
# A tibble: 8 x 7
  var                  mu      li      ls p_pos p_neg effect
  <chr>              <dbl>  <dbl>  <dbl> <dbl> <dbl> <lgl>
1 beta_temp_MODU    -0.0462 -0.338  0.269  0.378  0.622 FALSE
```

2	beta_rain_MODU	0.163	-0.102	0.412	0.892	0.108	TRUE
3	beta_fruitAB_MODU	0.0238	-0.0782	0.124	0.678	0.322	FALSE
4	beta_fruitS_MODU	0.00532	-0.0971	0.101	0.546	0.454	FALSE
5	beta_nut_MODU	0.0441	-0.0530	0.144	0.793	0.207	TRUE
6	beta_mor_MODU	0.0128	-0.0833	0.108	0.600	0.400	FALSE
7	beta_PLAN_MODU	0.0380	-0.0623	0.140	0.767	0.233	TRUE
8	beta_BIRD_MODU	0.00148	-0.0914	0.0984	0.503	0.497	FALSE

```

estimated_MODU_G_effect <-
  lapply(1, FUN =
    function(x) {

      f_ <- 'f_MODU['

      mu_MODU <- mean(dat$modularity)
      sd_MODU <- sd(dat$modularity)

      s <- grep('(.*) (MODU) (.*)$', colnames(post_MODU_G$alpha))

      est <- lapply(1:12, FUN =
        function(j) {
          f_ <- paste0(f_, j, ']')

          m <-
            apply(post_MODU_G$alpha[, s], 1, mean) +
            post_MODU_G$f[[f_]] +
            post_MODU_G$beta$beta_temp_MODU * mean(dat$z_temperature) +
            post_MODU_G$beta$beta_rain_MODU * mean(dat$z_rainfall) +
            post_MODU_G$beta$beta_fruitAB_MODU * mean(dat$fruit_abun) +
            post_MODU_G$beta$beta_fruitS_MODU * mean(dat$fruit_S) +
            post_MODU_G$beta$beta_nut_MODU * mean(dat$HV_plants_nut) +
            post_MODU_G$beta$beta_mor_MODU * mean(dat$HV_plants_morfo) +
            post_MODU_G$beta$beta_PLAN_MODU * mean(dat$HV_plant) +
            post_MODU_G$beta$beta_BIRD_MODU * mean(dat$HV_bird)

          m <- inv_logit(m)
          set.seed(123)
          m <- sample(m, 1e3)

          tibble(est = (m - mu_MODU) / sd_MODU,
                 month = j,
                 class = 'MODU')
        })
      est <- do.call('rbind', est)

      est
    })
  }

estimated_MODU_G_effect <-
  do.call('rbind', estimated_MODU_G_effect)

month_abbr <- na.omit(ymd(paste0("2023-", estimated_MODU_G_effect$month, "-01"),
                           label = TRUE, abbr = TRUE))

```

Warning: 2 failed to parse.

```
estimated_MODU_G_effect$month1 <- month_abbr
```

### S3.3.5 Plotting posterior distribution

```
estimated_MODU_G_effect |>
  group_by(month1) |>
  transmute(mu = mean(est),
            li = quantile(est, 0.025),
            ls = quantile(est, 0.975)) |>
  unique() |>
  ggplot() +
  geom_ribbon(aes(month1, ymin = li, ymax = ls), alpha = 0.5) +
  geom_line(aes(month1, mu)) +
  scale_x_date(date_labels = '%b', date_breaks = '2 month')
```

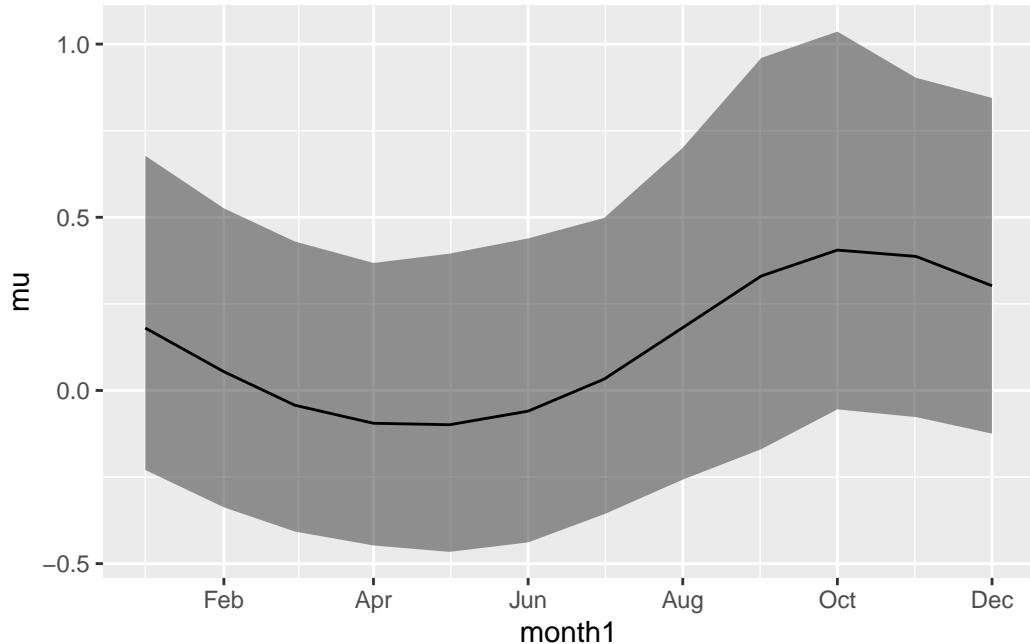


Figure S6: Monthly temporal pattern of modularity in seed dispersal networks

## S3.4 $H'_2$

### S3.4.1 Stan code

```

cat(file = 'H2_global.stan',
"

functions {
    vector GP_periodic(int period,           // periodicity
                        real gamma,          // smoothing term of the GP
                        real sigma,          // noise paramether
                        vector eta) {        // latent variable for each month

        int M = period;
        matrix[M, M] K;
        matrix[M, M] L_K;

        for (i in 1:(M - 1)) {
            for (j in (i+1):M) {
                real distance = abs(i - j);
                real periodic_distance = fmin(distance, period - distance);
                K[i, j] = sigma^2 * exp(-2 * square(sin(pi()*periodic_distance/period))/gamma^2);
                K[j, i] = K[i, j];   // filling the lower triangle
            }
            K[i, i] = sigma^2 + 1e-9; // small values to guarante stability
        }
        K[M, M] = sigma^2 + 1e-9; // small values to guarante stability
        return cholesky_decompose(K) * eta;
    }

    matrix GP_quadratic(matrix x,
                         real eta,
                         real rho,
                         real delta) {

        int N = dims(x)[1];
        matrix[N, N] K;
        matrix[N, N] L_K;

        for (i in 1:(N-1)) {
            K[i, i] = eta + delta;
            for (j in (i+1):N) {
                K[i, j] = square(eta) * exp(-rho * square(x[i, j]));
                K[j, i] = K[i, j];
            }
        }

        K[N, N] = eta + delta;
        L_K = cholesky_decompose(K);
        return L_K;
    }
}

data{
    int N;
    int N_sites;
    int N_month;
    array[N] int network_size;
    vector[N] modularity;
    vector[N] nestedness;
    vector[N] H2;
    array[N] int month;
    array[N] int site;
    vector[N] HV_network;
    vector[N] HV_plant;
    vector[N] HV_plants_morfo;
    vector[N] HV_plants_nut;
    vector[N] HV_bird;
    vector[N] z_temperature;
}

```

```

        vector[N] z_rainfall;
        vector[N] fruit_S;
        vector[N] fruit_abun;
        matrix[N_sites, N_sites] dist_sites;
    }

parameters {
    //////////////////////////////////////////////////////////////////// Fruit abundance ///////////////////////////////////////////////////////////////////
    // GP periodic
    real<lower = 0> gamma_fruitS;
    real<lower = 0> sigma_f_fruitS;
    vector[N_month] eta_fruitS;
    // GP quadratic
    vector[N_sites] z_sites_fruitS;
    real<lower = 0> eta_site_fruitS;
    real<lower = 0> rho_site_fruitS;

    // Pars linear model
    real beta_temp_fruitS;
    real beta_rain_fruitS;
    // noise for likelihood function
    real<lower = 0> sigma_fruitS;

    //////////////////////////////////////////////////////////////////// Fruit abundance ///////////////////////////////////////////////////////////////////
    // GP periodic
    real<lower = 0> gamma_fruitAB;
    real<lower = 0> sigma_f_fruitAB;
    vector[N_month] eta_fruitAB;
    // GP quadratic
    vector[N_sites] z_sites_fruitAB;
    real<lower = 0> eta_site_fruitAB;
    real<lower = 0> rho_site_fruitAB;

    // Pars linear model
    real beta_temp_fruitAB;
    real beta_rain_fruitAB;
    // noise for likelihood function
    real<lower = 0> sigma_fruitAB;

    //////////////////////////////////////////////////////////////////// Fru. Nut. HV ///////////////////////////////////////////////////////////////////
    // GP periodic
    real<lower = 0> gamma_nut_HV;
    real<lower = 0> sigma_f_nut_HV;
    vector[N_month] eta_nut_HV;
    // GP quadratic
    vector[N_sites] z_sites_nut_HV;
    real<lower = 0> eta_site_nut_hv;
    real<lower = 0> rho_site_nut_hv;

    // Pars linear model
    real beta_temp_nut_HV;
    real beta_rain_nut_HV;
    real beta_fruitAB_nut_HV;
    real beta_fruitS_nut_HV;
    real beta_BIRD_nut_HV;
    // noise for likelihood function
}

```

```

real<lower = 0> sigma_nut_HV;
//////////////////////////////////////////////////////////////// Fru. morpho. HV ///////////////////////////////
// GP periodic
real<lower = 0> gamma_mor_HV;
real<lower = 0> sigma_f_mor_HV;
vector[N_month] eta_mor_HV;
// GP quadratic
vector[N_sites] z_sites_mor_HV;
real<lower = 0> eta_site_mor_hv;
real<lower = 0> rho_site_mor_hv;

// Pars linear model

real beta_temp_mor_HV;
real beta_rain_mor_HV;
real beta_fruitAB_mor_HV;
real beta_fruits_mor_HV;
real beta_BIRD_mor_HV;
// noise for likelihood function
real<lower = 0> sigma_mor_HV;
real nu;

//////////////////////////////////////////////////////////////// Plants HV ///////////////////////////////
// GP periodic
real<lower = 0> gamma_PLAN_HV;
real<lower = 0> sigma_f_PLAN_HV;
vector[N_month] eta_PLAN_HV;
// GP quadratic
vector[N_sites] z_sites_PLAN_HV;
real<lower = 0> eta_site_PLAN_hv;
real<lower = 0> rho_site_PLAN_hv;

// Pars linear model
real beta_temp_PLAN_HV;
real beta_rain_PLAN_HV;
real beta_fruitAB_PLAN_HV;
real beta_fruits_PLAN_HV;
real beta_nut_PLAN_HV;
real beta_mor_PLAN_HV;
real beta_BIRD_PLAN_HV;
// noise for likelihood function
real<lower = 0> sigma_PLAN_HV;

//////////////////////////////////////////////////////////////// Birsd's HV ///////////////////////////////
// GP periodic
real<lower = 0> gamma_BIRD_HV;
real<lower = 0> sigma_f_BIRD_HV;
vector[N_month] eta_BIRD_HV;
// GP quadratic
vector[N_sites] z_sites_BIRD_HV;
real<lower = 0> eta_site_BIRD_hv;
real<lower = 0> rho_site_BIRD_hv;

// Pars linear model
real beta_temp_BIRD_HV;
real beta_rain_BIRD_HV;
real beta_fruitAB_BIRD_HV;
real beta_fruits_BIRD_HV;
// noise for likelihood function
real<lower = 0> sigma_BIRD_HV;

//////////////////////////////////////////////////////////////// Network metrics ///////////////////////////////

```

```

// ////////////////////////////// H2 //////////////////////////////
// GP periodic
real<lower = 0> gamma_H2;
real<lower = 0> sigma_f_H2;
vector[N_month] eta_H2;
// GP quadratic
vector[N_sites] z_sites_H2;
real<lower = 0> eta_site_H2;
real<lower = 0> rho_site_H2;

// Pars linear model
real beta_temp_H2;
real beta_rain_H2;
real beta_fruitAB_H2;
real beta_fruitS_H2;
real beta_nut_H2;
real beta_mor_H2;
real beta_PLAN_H2;
real beta_BIRD_H2;
// noise for likelihood function
real<lower = 0> sigma_H2;

}

transformed parameters {
    ////////////////////////////// Fruit abundance //////////////////////////////
    // GP periodic
    vector[N_month] f_fruitAB;

    f_fruitAB = GP_periodic(
        12,
        gamma_fruitAB,
        sigma_f_fruitAB,
        eta_fruitAB
    );

    // GP quadratic
    vector[N_sites] alpha_fruitAB;
    matrix[N_sites, N_sites] L_K_fruitAB;
    L_K_fruitAB = GP_quadratic(dist_sites,
        eta_site_fruitAB,
        rho_site_fruitAB, 0.001);
    alpha_fruitAB = L_K_fruitAB * z_sites_fruitAB;

    ////////////////////////////// Fruit richness //////////////////////////////
    // GP periodic
    vector[N_month] f_fruitS;

    f_fruitS = GP_periodic(
        12,
        gamma_fruitS,
        sigma_f_fruitS,
        eta_fruitS
    );

    // GP quadratic
    vector[N_sites] alpha_fruitS;
    matrix[N_sites, N_sites] L_K_fruitS;
    L_K_fruitS = GP_quadratic(dist_sites,
        eta_site_fruitS,
        rho_site_fruitS, 0.001);
    alpha_fruitS = L_K_fruitS * z_sites_fruitS;
}

```

```

////////// Fru. Nut. HV //////////
// GP periodic
vector[N_month] f_nut_HV;

f_nut_HV = GP_periodic(
    12,
    gamma_nut_HV,
    sigma_f_nut_HV,
    eta_nut_HV
);

// GP quadratic
vector[N_sites] alpha_nut_HV;
matrix[N_sites, N_sites] L_K_nut_HV;
L_K_nut_HV = GP_quadratic(dist_sites,
                           eta_site_nut_hv,
                           rho_site_nut_hv, 0.001);
alpha_nut_HV = L_K_nut_HV * z_sites_nut_HV;

// Fru. morpho. HV //////////
// GP periodic
vector[N_month] f_mor_HV;

f_mor_HV = GP_periodic(
    12,
    gamma_mor_HV,
    sigma_f_mor_HV,
    eta_mor_HV
);

// GP quadratic
vector[N_sites] alpha_mor_HV;
matrix[N_sites, N_sites] L_K_mor_HV;
L_K_mor_HV = GP_quadratic(dist_sites,
                           eta_site_mor_hv,
                           rho_site_mor_hv, 0.001);
alpha_mor_HV = L_K_mor_HV * z_sites_mor_HV;

// Plants HV //////////
// GP periodic
vector[N_month] f_PLAN_HV;

f_PLAN_HV = GP_periodic(
    12,
    gamma_PLAN_HV,
    sigma_f_PLAN_HV,
    eta_PLAN_HV
);

// GP quadratic
vector[N_sites] alpha_PLAN_HV;
matrix[N_sites, N_sites] L_K_PLAN_HV;
L_K_PLAN_HV = GP_quadratic(dist_sites,
                           eta_site_PLAN_hv,
                           rho_site_PLAN_hv, 0.001);
alpha_PLAN_HV = L_K_PLAN_HV * z_sites_PLAN_HV;

// Birds HV //////////
// GP periodic
vector[N_month] f_BIRD_HV;

f_BIRD_HV = GP_periodic(

```

```

    12,
    gamma_BIRD_HV,
    sigma_f_BIRD_HV,
    eta_BIRD_HV
);

// GP quadratic
vector[N_sites] alpha_BIRD_HV;
matrix[N_sites, N_sites] L_K_BIRD_HV;
L_K_BIRD_HV = GP_quadratic(dist_sites,
                           eta_site_BIRD_hv,
                           rho_site_BIRD_hv, 0.001);
alpha_BIRD_HV = L_K_BIRD_HV * z_sites_BIRD_HV;

// ////////////////////////////// H2 //////////////////////////////
// //////////////////////////////
// GP periodic
vector[N_month] f_H2;

f_H2 = GP_periodic(
    12,
    gamma_H2,
    sigma_f_H2,
    eta_H2);

// GP quadratic
vector[N_sites] alpha_H2;
matrix[N_sites, N_sites] L_K_H2;
L_K_H2 = GP_quadratic(dist_sites,
                       eta_site_H2,
                       rho_site_H2, 0.001);
alpha_H2 = L_K_H2 * z_sites_H2;

}

model {

    ////////////////////////////// Fruit abundance //////////////////////////////
    // priors for periodic GP
    eta_fruitAB ~ normal(0, 0.5);
    gamma_fruitAB ~ inv_gamma(5, 5);
    sigma_f_fruitAB ~ cauchy(0, 1);
    // GP quadratic
    z_sites_fruitAB ~ normal(0, 0.25);
    eta_site_fruitAB ~ exponential(4);
    rho_site_fruitAB ~ exponential(1);
    // Pars linear model
    beta_temp_fruitAB ~ normal(0, 0.5);
    beta_rain_fruitAB ~ normal(0, 0.5);
    // noise parameter of the likelihood
    sigma_fruitAB ~ exponential(1);

    ////////////////////////////// Fruit richness //////////////////////////////
    // priors for periodic GP
    eta_fruitS ~ normal(0, 0.5);
    gamma_fruitS ~ inv_gamma(5, 5);
    sigma_f_fruitS ~ cauchy(0, 1);
    // GP quadratic
    z_sites_fruitS ~ normal(0, 0.25);
    eta_site_fruitS ~ exponential(4);
    rho_site_fruitS ~ exponential(1);
    // Pars linear model
    beta_temp_fruitS ~ normal(0, 0.5);
    beta_rain_fruitS ~ normal(0, 0.5);
    // noise parameter of the likelihood
}

```

```

sigma_fruitS ~ exponential(1);

////////////////// Fru. Nut. HV ///////////////////////
// priors for periodic GP
eta_nut_HV ~ normal(0, 0.5);
gamma_nut_HV ~ inv_gamma(5, 5);
sigma_f_nut_HV ~ cauchy(0, 1);
// GP quadratic
z_sites_nut_HV ~ normal(0, 0.25);
eta_site_nut_hv ~ exponential(4);
rho_site_nut_hv ~ exponential(1);
// Pars linear model
beta_temp_nut_HV ~ normal(0, 0.5);
beta_rain_nut_HV ~ normal(0, 0.5);
beta_fruitAB_nut_HV ~ normal(0, 0.5);
beta_fruitS_nut_HV ~ normal(0, 0.5);
beta_BIRD_nut_HV ~ normal(0, 0.5);
// noise parameter of the likelihood
sigma_nut_HV ~ exponential(1);
//
////////////////// Fru. morpho. HV ///////////////////////
// priors for periodic GP
eta_mor_HV ~ normal(0, 0.5);
gamma_mor_HV ~ inv_gamma(5, 5);
sigma_f_mor_HV ~ cauchy(0, 1);
// GP quadratic
z_sites_mor_HV ~ normal(0, 0.25);
eta_site_mor_hv ~ exponential(4);
rho_site_mor_hv ~ exponential(1);
// Pars linear model
beta_temp_mor_HV ~ normal(0, 0.5);
beta_rain_mor_HV ~ normal(0, 0.5);
beta_fruitAB_mor_HV ~ normal(0, 0.5);
beta_fruitS_mor_HV ~ normal(0, 0.5);
beta_BIRD_mor_HV ~ normal(0, 0.5);
// noise parameter of the likelihood
sigma_mor_HV ~ exponential(1);

////////////////// Plants HV ///////////////////////
// priors for periodic GP
eta_PLAN_HV ~ normal(0, 1);
gamma_PLAN_HV ~ inv_gamma(5, 5);
sigma_f_PLAN_HV ~ cauchy(0, 1);
// GP quadratic
z_sites_PLAN_HV ~ normal(0, 0.5);
eta_site_PLAN_hv ~ exponential(4);
rho_site_PLAN_hv ~ exponential(1);
// Pars linear model
beta_temp_PLAN_HV ~ normal(0, 0.5);
beta_rain_PLAN_HV ~ normal(0, 0.5);
beta_fruitAB_PLAN_HV ~ normal(0, 0.5);
beta_fruitS_PLAN_HV ~ normal(0, 0.5);
beta_nut_PLAN_HV ~ normal(0, 0.5);
beta_mor_PLAN_HV ~ normal(0, 0.5);
beta_BIRD_PLAN_HV ~ normal(0, 0.5);
// noise parameter of the likelihood
sigma_PLAN_HV ~ exponential(1);

////////////////// Birds HV ///////////////////////
// priors for periodic GP
eta_BIRD_HV ~ normal(0, 0.5);
gamma_BIRD_HV ~ inv_gamma(5, 5);
sigma_f_BIRD_HV ~ cauchy(0, 1);

```

```

// GP quadratic
z_sites_BIRD_HV ~ normal(0, 0.5);
eta_site_BIRD_hv ~ exponential(4);
rho_site_BIRD_hv ~ exponential(1);
// Pars linear model
beta_temp_BIRD_HV ~ normal(0, 0.5);
beta_rain_BIRD_HV ~ normal(0, 0.5);
beta_fruitAB_BIRD_HV ~ normal(0, 0.5);
beta_fruitS_BIRD_HV ~ normal(0, 0.5);
// noise parameter of the likelihood
sigma_BIRD_HV ~ exponential(1);

// ////////////////////////////// H2 //////////////////////////////
vector[N] mu_H2;
vector[N] p1_H2;
vector[N] p2_H2;
// priors for periodic GP
eta_H2 ~ normal(0, 1);
gamma_H2 ~ inv_gamma(5, 5);
sigma_f_H2 ~ cauchy(0, 1);
// GP quadratic
z_sites_H2 ~ normal(0, 0.25);
eta_site_H2 ~ exponential(4);
rho_site_H2 ~ exponential(1);
// Pars linear model
beta_fruitAB_H2 ~ normal(0, 0.055);
beta_fruitS_H2 ~ normal(0, 0.055);
beta_nut_H2 ~ normal(0, 0.055);
beta_mor_H2 ~ normal(0, 0.055);
beta_PLAN_H2 ~ normal(0, 0.055);
beta_BIRD_H2 ~ normal(0, 0.055);
// beta_NW_H2 ~ normal(0, 0.055);
// noise parameter of the likelihood
sigma_H2 ~ exponential(2);

////////////////// fruit abundance ///////////////////
// for (i in 1:N) {
//   fruit_abun[i] ~ student_t(7, alpha_fruitAB[site[i]] + f_fruitAB[month[i]] +
//                             beta_temp_fruitAB * z_temperature[i] +
//                             beta_rain_fruitAB * z_rainfall[i],
//                             sigma_fruitAB);
// }

////////////////// fruit richness ///////////////////
// for (i in 1:N) {
//   fruit_S[i] ~ student_t(7, alpha_fruitS[site[i]] + f_fruitS[month[i]] +
//                           beta_temp_fruitS * z_temperature[i] +
//                           beta_rain_fruitS * z_rainfall[i],
//                           sigma_fruitS);
// }

////////////////// Fru. Nut. HV ///////////////////
// for (i in 1:N) {
//   HV_plants_nut[i] ~ student_t(7, alpha_nut_HV[site[i]] + f_nut_HV[month[i]] +
//                                 beta_temp_nut_HV * z_temperature[i] +
//                                 beta_rain_nut_HV * z_rainfall[i] +
//                                 beta_BIRD_nut_HV * HV_bird[i] +
//                                 beta_fruitAB_nut_HV * fruit_abun[i] +
//                                 beta_fruitS_nut_HV * fruit_S[i],
//                                 sigma_nut_HV);
// }

// //////////////////// Fru. morpho. HV ///////////////////

```

```

// /////////////////////////////////
for (i in 1:N) {
  HV_plants_morfo[i] ~ student_t(15, alpha_mor_HV[site[i]] + f_mor_HV[month[i]] +
    beta_temp_mor_HV * z_temperature[i] +
    beta_rain_mor_HV * z_rainfall[i] +
    beta_fruitAB_mor_HV * fruit_abun[i] +
    beta_fruitS_mor_HV * fruit_S[i] +
    beta_BIRD_mor_HV * HV_bird[i],
    sigma_mor_HV);
}

// /////////////////////////////// Plants HV ///////////////////////////////
// /////////////////////////////// for (i in 1:N) {
for (i in 1:N) {
  HV_plant[i] ~ student_t(7, alpha_PLAN_HV[site[i]] + f_PLAN_HV[month[i]] +
    beta_temp_PLAN_HV * z_temperature[i] +
    beta_rain_PLAN_HV * z_rainfall[i] +
    beta_fruitAB_PLAN_HV * fruit_abun[i] +
    beta_fruitS_PLAN_HV * fruit_S[i] +
    beta_nut_PLAN_HV * HV_plants_nut[i] +
    beta_mor_PLAN_HV * HV_plants_morfo[i] +
    beta_BIRD_PLAN_HV * HV_bird[i],
    sigma_PLAN_HV);
}

// /////////////////////////////// Birds HV ///////////////////////////////
// /////////////////////////////// for (i in 1:N) {
for (i in 1:N) {
  HV_bird[i] ~ student_t(7, alpha_BIRD_HV[site[i]] + f_BIRD_HV[month[i]] +
    beta_temp_BIRD_HV * z_temperature[i] +
    beta_rain_BIRD_HV * z_rainfall[i] +
    beta_fruitAB_BIRD_HV * fruit_abun[i] +
    beta_fruitS_BIRD_HV * fruit_S[i],
    sigma_BIRD_HV);
}

// /////////////////////////////// H2 ///////////////////////////////
// /////////////////////////////// for (i in 1:N) {
for (i in 1:N) {
  mu_H2[i] = inv_logit(alpha_H2[site[i]] + f_H2[month[i]] +
    beta_temp_H2 * z_temperature[i] +
    beta_rain_H2 * z_rainfall[i] +
    beta_fruitAB_H2 * fruit_abun[i] +
    beta_fruitS_H2 * fruit_S[i] +
    beta_nut_H2 * HV_plants_nut[i] +
    beta_mor_H2 * HV_plants_morfo[i] +
    beta_PLAN_H2 * HV_plant[i] +
    beta_BIRD_H2 * HV_bird[i]);
}

p1_H2 = mu_H2 * sigma_H2;
p2_H2 = (1 - mu_H2) * sigma_H2;

H2 ~ beta(p1_H2, p2_H2);

}

generated quantities {

// /////////////////////////////// Fruit abundance ///////////////////////////////
array[N] real ppcheck_fruitAB;
vector[N] mu_fruitAB;

for (i in 1:N) {
  mu_fruitAB[i] = alpha_fruitAB[site[i]] + f_fruitAB[month[i]] +
    beta_temp_fruitAB * z_temperature[i] +
    beta_rain_fruitAB * z_rainfall[i];
}

```

```

}

ppcheck_fruitAB = student_t_rng(7, mu_fruitAB, sigma_fruitAB);

////////////////// Fruit richness ///////////////////
array[N] real ppcheck_fruitS;
vector[N] mu_fruitS;

for (i in 1:N) {
    mu_fruitS[i] = alpha_fruitS[site[i]] + f_fruitS[month[i]] +
                    beta_temp_fruitS * z_temperature[i] +
                    beta_rain_fruitS * z_rainfall[i];
}
ppcheck_fruitS = student_t_rng(7, mu_fruitS, sigma_fruitS);

////////////////// Fru. Nut. HV ///////////////////
array[N] real ppcheck_nut_HV;
vector[N] mu_nut_HV;

for (i in 1:N) {
    mu_nut_HV[i] = alpha_nut_HV[site[i]] + f_nut_HV[month[i]] +
                    beta_temp_nut_HV * z_temperature[i] +
                    beta_rain_nut_HV * z_rainfall[i] +
                    beta_fruitAB_nut_HV * fruit_abun[i] +
                    beta_fruitS_nut_HV * fruit_S[i] +
                    beta_BIRD_nut_HV * HV_bird[i];
}
ppcheck_nut_HV = student_t_rng(7, mu_nut_HV, sigma_nut_HV);
//
// //////////////////// Fru. morpho. HV ///////////////////
array[N] real ppcheck_mor_HV;
vector[N] mu_mor_HV;

for (i in 1:N) {
    mu_mor_HV[i] = alpha_mor_HV[site[i]] + f_mor_HV[month[i]] +
                    beta_temp_mor_HV * z_temperature[i] +
                    beta_rain_mor_HV * z_rainfall[i] +
                    beta_fruitAB_mor_HV * fruit_abun[i] +
                    beta_fruitS_mor_HV * fruit_S[i] +
                    beta_BIRD_mor_HV * HV_bird[i];
}
ppcheck_mor_HV = student_t_rng(15, mu_mor_HV, sigma_mor_HV);

// //////////////////// Plants HV ///////////////////
array[N] real ppcheck_PLAN_HV;
vector[N] mu_PLAN_HV;

for (i in 1:N) {
    mu_PLAN_HV[i] = alpha_PLAN_HV[site[i]] + f_PLAN_HV[month[i]] +
                    beta_temp_PLAN_HV * z_temperature[i] +
                    beta_rain_PLAN_HV * z_rainfall[i] +
                    beta_fruitAB_PLAN_HV * fruit_abun[i] +
                    beta_fruitS_PLAN_HV * fruit_S[i] +
                    beta_nut_PLAN_HV * HV_plants_nut[i] +
                    beta_mor_PLAN_HV * HV_plants_morfo[i] +
                    beta_BIRD_PLAN_HV * HV_bird[i];
}

```

```

ppcheck_PLAN_HV = student_t_rng(7, mu_PLAN_HV, sigma_PLAN_HV);

////////////////////// Birds HV ///////////////////////////////
array[N] real ppcheck_BIRD_HV;
vector[N] mu_BIRD_HV;

for (i in 1:N) {
    mu_BIRD_HV[i] = alpha_BIRD_HV[site[i]] + f_BIRD_HV[month[i]] +
        beta_temp_BIRD_HV * z_temperature[i] +
        beta_rain_BIRD_HV * z_rainfall[i] +
        beta_fruitAB_BIRD_HV * fruit_abun[i] +
        beta_fruitS_BIRD_HV * fruit_S[i];
}

ppcheck_BIRD_HV = student_t_rng(7, mu_BIRD_HV, sigma_BIRD_HV);

// //////////////////// H2 ///////////////////////////////
array[N] real ppcheck_H2;
vector[N] mu_H2;
vector[N] p1_H2;
vector[N] p2_H2;
real variance_fixed;
real variance_GP_temp;
real variance_GP_space;
real variance_total;
real prop_GP_temp;
real prop_GP_space;
real prop_fixed;

for (i in 1:N) {
    mu_H2[i] = alpha_H2[site[i]] + f_H2[month[i]] +
        beta_temp_H2 * z_temperature[i] +
        beta_rain_H2 * z_rainfall[i] +
        beta_fruitAB_H2 * fruit_abun[i] +
        beta_fruitS_H2 * fruit_S[i] +
        beta_nut_H2 * HV_plants_nut[i] +
        beta_mor_H2 * HV_plants_morfo[i] +
        beta_PLAN_H2 * HV_plant[i] +
        beta_BIRD_H2 * HV_bird[i];
}

// variance decomposition

variance_fixed = variance(mu_H2);
variance_GP_space = eta_site_H2^2;
variance_GP_temp = sigma_f_H2^2;
variance_total = variance_fixed + variance_GP_temp +
    variance_GP_space + ((pi()^2)/3);
prop_GP_temp = variance_GP_temp / variance_total;
prop_GP_space = variance_GP_space / variance_total;
prop_fixed = variance_fixed / variance_total;

mu_H2 = inv_logit(mu_H2);
p1_H2 = mu_H2 * sigma_H2;
p2_H2 = (1 - mu_H2) * sigma_H2;

ppcheck_H2 = beta_rng(p1_H2, p2_H2);

}
"
)

```

### S3.4.2 Fitting the model

```
file <- paste0(getwd(), '/H2_global.stan')
fit_H2_G <- cmdstan_model(file, compile = T)

mod_H2_G <-
  fit_H2_G$sample(
    data = dat,
    iter_sampling = 2e3,
    iter_warmup = 500,
    thin = 3,
    chains = 3,
    parallel_chains = 3,
    seed = 5
  )
```

### S3.4.3 Sampling diagnostics

```
summary_H2_G <- mod_H2_G$summary()

par(mfrow = c(3, 3), mar = c(4, 4, 2, 1))

mod_diagnostics(mod_H2_G,
                 summary_H2_G[grep('fruitAB', summary_H2_G$variable), ])
title('Fruit abundance')
mod_diagnostics(mod_H2_G,
                 summary_H2_G[grep('fruits', summary_H2_G$variable), ])
title('Fruit richness')
mod_diagnostics(mod_H2_G,
                 summary_H2_G[grep('nut_HV', summary_H2_G$variable), ])
title('Frut. Nut. HV')
mod_diagnostics(mod_H2_G,
                 summary_H2_G[grep('mor_HV', summary_H2_G$variable), ])
title('Frut. Morpho. HV')
mod_diagnostics(mod_H2_G,
                 summary_H2_G[grep('PLAN_HV', summary_H2_G$variable), ])
title('Plants HV')
mod_diagnostics(mod_H2_G,
                 summary_H2_G[grep('BIRD_HV', summary_H2_G$variable), ])
title('Birds HV')
mod_diagnostics(mod_H2_G,
                 summary_H2_G[grep('H2', summary_H2_G$variable), ])
title('H2')
par(mfrow = c(1, 1))
```

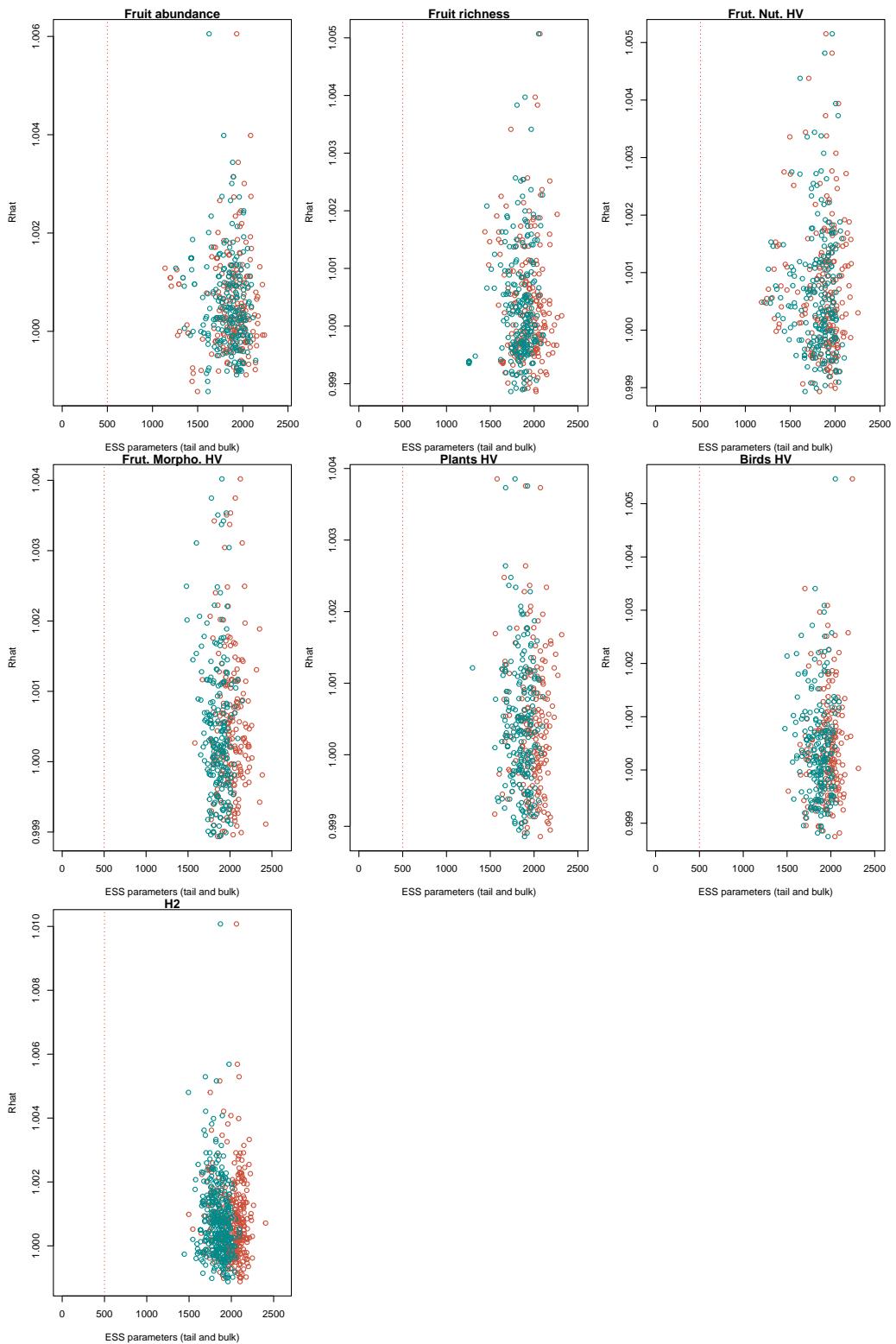


Figure S7: Rhat values vs effective sampling size (ess). Rhat  $< 1.05$  indicates that the Markov Chains converged to the same stationary distribution. ess must be at least 100 per chain in order to be reliable

```

par(mfrow = c(3, 3), mar = c(4, 4, 1, 1))
ppcheck_H2_G <- mod_H2_G$draws('ppcheck_fruitAB', format = 'matrix')
plot(density(dat$fruit_abun), main = '', xlab = 'Fruit abundance',
      ylim = c(0, 1))
for (i in 1:100) lines(density(ppcheck_H2_G[i, ]), lwd = 0.1)
lines(density(dat$fruit_abun), col = 'red', lwd = 2)

ppcheck_H2_G <- mod_H2_G$draws('ppcheck_fruitS', format = 'matrix')
plot(density(dat$fruit_S), main = '', xlab = 'Fruit richness',
      ylim = c(0, 1))
for (i in 1:100) lines(density(ppcheck_H2_G[i, ]), lwd = 0.1)
lines(density(dat$fruit_S), col = 'red', lwd = 2)
ppcheck_H2_G <- mod_H2_G$draws('ppcheck_nut_HV', format = 'matrix')
plot(density(dat$HV_plants_nut), main = '', xlab = 'Frut. Nut. HV',
      ylim = c(0, 1))
for (i in 1:100) lines(density(ppcheck_H2_G[i, ]), lwd = 0.1)
lines(density(dat$HV_plants_nut), col = 'red', lwd = 2)

ppcheck_H2_G <- mod_H2_G$draws('ppcheck_mor_HV', format = 'matrix')
plot(density(dat$HV_plants_morfo), main = '', xlab = 'Frut. Morpho. HV',
      ylim = c(0, 1))
for (i in 1:100) lines(density(ppcheck_H2_G[i, ]), lwd = 0.1)
lines(density(dat$HV_plants_morfo), col = 'red', lwd = 2)

ppcheck_H2_G <- mod_H2_G$draws('ppcheck_PLAN_HV', format = 'matrix')
plot(density(dat$HV_plant), main = '', xlab = 'Plants HV',
      ylim = c(0, 1))
for (i in 1:100) lines(density(ppcheck_H2_G[i, ]), lwd = 0.1)
lines(density(dat$HV_plant), col = 'red', lwd = 2)

ppcheck_H2_G <- mod_H2_G$draws('ppcheck_BIRD_HV', format = 'matrix')
plot(density(dat$HV_bird), main = '', xlab = 'Birds HV',
      ylim = c(0, 1))
for (i in 1:100) lines(density(ppcheck_H2_G[i, ]), lwd = 0.1)
lines(density(dat$HV_bird), col = 'red', lwd = 2)

ppcheck_H2_G <- mod_H2_G$draws('ppcheck_H2', format = 'matrix')
plot(density(dat$H2), main = '', xlab = 'H2',
      ylim = c(0, 2))
for (i in 1:100) lines(density(ppcheck_H2_G[i, ]), lwd = 0.1)
lines(density(dat$H2), col = 'red', lwd = 2)
par(mfrow = c(1, 1))

```

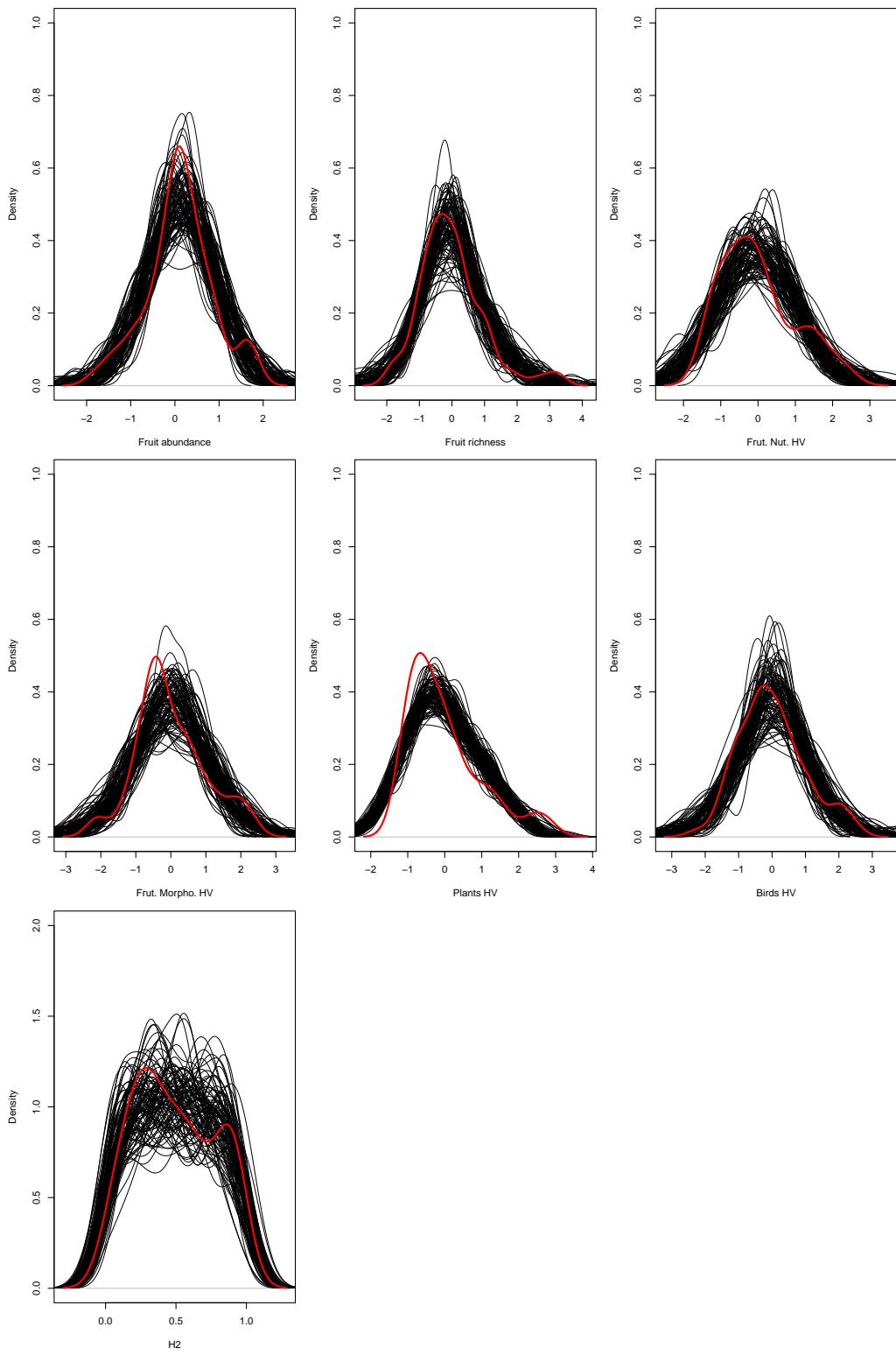


Figure S8: Posterior predictive checks of the of variables integrating the generative model explaining the H2 of seed seed dispersal networks

### S3.4.4 Extracting posterior draws

```

pars_H2_G <- summary_H2_G[grep('`beta', summary_H2_G$variable), ]$variable
betas_H2_G <- mod_H2_G$draws(pars_H2_G, format = 'df')
betas_H2_G <- betas_H2_G[, -c((ncol(betas_H2_G)-2):ncol(betas_H2_G))]

```

Warning: Dropping 'draws\_df' class as required metadata was removed.

```

betas_H2_G <-
lapply(1:ncol(betas_H2_G),
      function(i) {
        x <- betas_H2_G[[i]]
        var <- colnames(betas_H2_G)[[i]]

        df <-
        tibble(var = var,
               mu = mean(x),
               li = quantile(x, 0.025),
               ls = quantile(x, 0.975),
               p_pos = mean(x > 0),
               p_neg = mean(x < 0))

        df$effect <- df$p_pos > 0.7 | df$p_neg > 0.7
        df
      })

betas_H2_G <- do.call('rbind', betas_H2_G)

betas_H2_G_LABS <- betas_H2_G[c(grep('PLAN_H2', betas_H2_G$var),
                                 grep('mor_H2', betas_H2_G$var),
                                 grep('nut_H2', betas_H2_G$var)), ]

betas_H2_GEFFECT <- betas_H2_G[grep('H2', betas_H2_G$var), ]

```

```

betas_H2_G_z <- betas_H2_G

betas_H2_G_z$var <-
  gsub('`^(beta_)([A-Za-z]*)(_)(.*)$', '\\\\2 -> \\\\4', betas_H2_G_z$var)

colnames(betas_H2_G_z) <-
  c('Directional effect', 'Mean', '0.025%', '0.975%', 'P(effect > 0)',
    'P(effect < 0)', 'effect')

knitr::kable(betas_H2_G_z,
             digits = 2,
             caption = 'Average values, 95% CIs, and probabilities of directional effects for the relationships a'

```

Table S3: Average values, 95% CIs, and probabilities of directional effects for the relationships assessed in the generative model. An effect is marked as TRUE if  $P(\text{effect} > 0) > 0.7$  or  $P(\text{effect} < 0) > 0.7$  or.

Directional effect	Mean	0.025%	0.975%	$P(\text{effect} > 0)$	$P(\text{effect} < 0)$	effect
temp -> fruitS	-0.15	-0.42	0.12	0.13	0.87	TRUE
rain -> fruitS	0.45	0.16	0.73	1.00	0.00	TRUE
temp -> fruitAB	0.00	-0.26	0.24	0.49	0.51	FALSE
rain -> fruitAB	-0.09	-0.29	0.10	0.17	0.83	TRUE
temp -> nut_HV	0.36	0.06	0.66	0.99	0.01	TRUE
rain -> nut_HV	-0.03	-0.32	0.29	0.45	0.55	FALSE
fruitAB -> nut_HV	0.22	-0.15	0.61	0.87	0.13	TRUE
fruitS -> nut_HV	-0.03	-0.35	0.27	0.42	0.58	FALSE
BIRD -> nut_HV	0.02	-0.17	0.20	0.61	0.39	FALSE
temp -> mor_HV	0.03	-0.25	0.32	0.58	0.42	FALSE
rain -> mor_HV	0.06	-0.26	0.38	0.65	0.35	FALSE
fruitAB -> mor_HV	0.21	-0.15	0.60	0.87	0.13	TRUE
fruitS -> mor_HV	-0.03	-0.33	0.29	0.45	0.55	FALSE
BIRD -> mor_HV	0.04	-0.21	0.29	0.63	0.37	FALSE
temp -> PLAN_HV	-0.01	-0.14	0.11	0.42	0.58	FALSE
rain -> PLAN_HV	0.07	-0.06	0.20	0.85	0.15	TRUE
fruitAB -> PLAN_HV	-0.14	-0.29	0.00	0.03	0.97	TRUE
nut -> PLAN_HV	0.59	0.48	0.70	1.00	0.00	TRUE
mor -> PLAN_HV	0.47	0.38	0.56	1.00	0.00	TRUE
BIRD -> PLAN_HV	0.03	-0.06	0.12	0.74	0.26	TRUE
temp -> BIRD_HV	0.19	-0.13	0.50	0.89	0.11	TRUE
rain -> BIRD_HV	-0.13	-0.48	0.19	0.22	0.78	TRUE
fruitAB -> BIRD_HV	0.24	-0.12	0.59	0.91	0.09	TRUE
fruitS -> BIRD_HV	-0.05	-0.37	0.27	0.39	0.61	FALSE
temp -> H2	-0.09	-0.39	0.20	0.27	0.73	TRUE
rain -> H2	0.21	-0.07	0.52	0.93	0.07	TRUE
fruitAB -> H2	0.01	-0.10	0.11	0.56	0.44	FALSE
fruitS -> H2	0.00	-0.10	0.10	0.49	0.51	FALSE

Directional effect	Mean	0.025%	0.975%	P(effect > 0)	P(effect < 0)	effect
nut -> H2	0.01	-0.10	0.10	0.55	0.45	FALSE
mor -> H2	0.00	-0.10	0.10	0.46	0.54	FALSE
PLAN -> H2	0.02	-0.09	0.12	0.62	0.38	FALSE
BIRD -> H2	0.01	-0.10	0.10	0.56	0.44	FALSE

```

pars_H2_G <-
  summary_H2_G[c(grep('`beta', summary_H2_G$variable),
                 grep('`alpha', summary_H2_G$variable),
                 grep('`f', summary_H2_G$variable),
                 grep('`sigma', summary_H2_G$variable)), ]$variable

post_H2_G <- mod_H2_G$draws(pars_H2_G, format = 'df')

post_H2_G <-
  lapply(c(`beta`, `alpha`, `f`, `sigma`), FUN =
    function(x) {
      post_H2_G[, grep(x, colnames(post_H2_G))]
    })
  
```

Warning: Dropping 'draws\_df' class as required metadata was removed.  
 Warning: Dropping 'draws\_df' class as required metadata was removed.  
 Warning: Dropping 'draws\_df' class as required metadata was removed.  
 Warning: Dropping 'draws\_df' class as required metadata was removed.

```

names(post_H2_G) <- c('beta', 'alpha', 'f', 'sigma')

estimated_H2_G_effect <-
  lapply(1, FUN =
    function(x) {

      f_ <- 'f_H2['
      s <- grep('alpha_H2', colnames(post_H2_G$alpha))
      mu_H2_G <- mean(dat$H2)
      sd_H2_G <- sd(dat$H2)

      est <- lapply(1:12, FUN =
        function(j) {
          f_ <- paste0(f_, j, ']')
          m <-
            apply(post_H2_G$alpha[, s], 1, mean) +
            post_H2_G$f[[f_]] +
            post_H2_G$beta$beta_temp_H2 * mean(dat$z_temperature) +
            post_H2_G$beta$beta_rain_H2 * mean(dat$z_rainfall) +
            post_H2_G$beta$beta_fruitAB_H2 * mean(dat$fruit_abun) +
            post_H2_G$beta$beta_fruitS_H2 * mean(dat$fruit_S) +
            post_H2_G$beta$beta_nut_H2 * mean(dat$HV_plants_nut) +
            post_H2_G$beta$beta_mor_H2 * mean(dat$HV_plants_morfo) +
            post_H2_G$beta$beta_PLAN_H2 * mean(dat$HV_plant) +
            post_H2_G$beta$beta_BIRD_H2 * mean(dat$HV_bird)
        }
      )
    }
  )
  
```

```

    m <- inv_logit(m)

    set.seed(123)
    m <- sample(m, 1e3)

    tibble(est = (m - mu_H2_G) / sd_H2_G,
           month = j,
           class = 'H2')

  })
est <- do.call('rbind', est)

est

})

estimated_H2_G_effect <-
  do.call('rbind', estimated_H2_G_effect)

month_abbr <- na.omit(ymd(paste0("2023-", estimated_H2_G_effect$month, "-01"),
                           label = TRUE, abbr = TRUE))

```

Warning: 2 failed to parse.

```
estimated_H2_G_effect$month1 <- month_abbr
```

### S3.4.5 Plotting posterior distribution

```

estimated_H2_G_effect |>
  group_by(month1) |>
  transmute(mu = mean(est),
            li = quantile(est, 0.025),
            ls = quantile(est, 0.975)) |>
  unique() |>
  ggplot() +
  geom_ribbon(aes(month1, ymin = li, ymax = ls), alpha = 0.5) +
  geom_line(aes(month1, mu)) +
  scale_x_date(date_labels = '%b', date_breaks = '2 month')

```

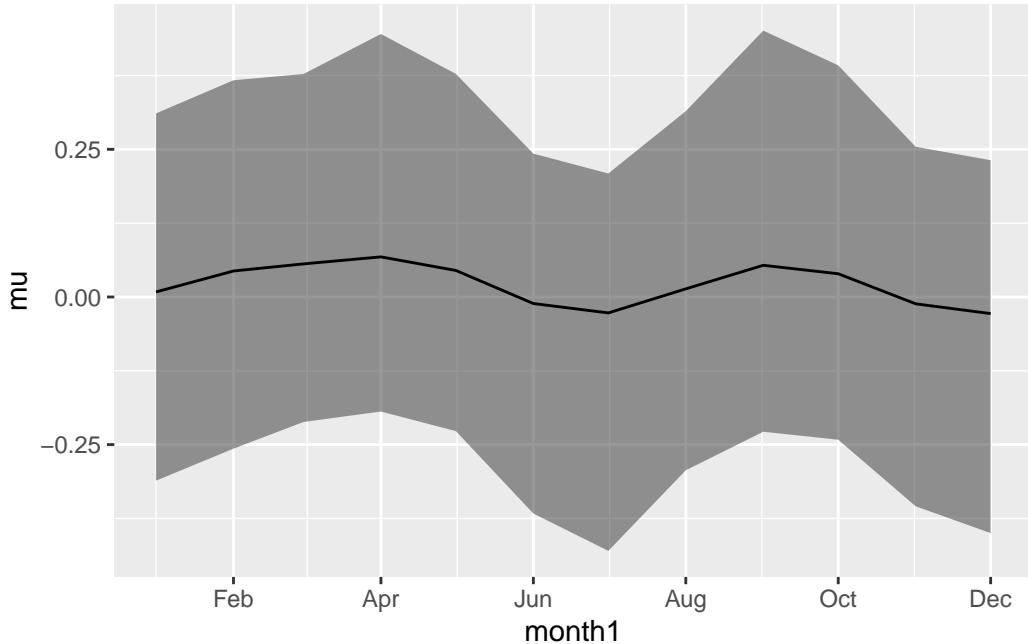


Figure S9: Monthly temporal pattern of H2 in seed dispersal networks

## S4 Final results

### S4.1 Climatic predictors

```

global_climate <- readRDS('climate_for_plot.rds')

global_climate$class <- rep(c('Rainfall', 'Temperature'), each = 12)

global_climate$month1 <- ymd(paste0('2025-',
                                      global_climate$month,
                                      '-01')))

plot_global_climate <-
  global_climate |>
  mutate(type = 'Climate') |>
  ggplot(aes(month1, mu, ymin = li_mu, ymax = ls_mu)) +
  geom_line(aes(color = class), linewidth = 0.7, alpha = 0.5) +
  # geom_point(data =
  #             rbind(global_climate[global_climate$class == 'Temperature', ] |>
  #                   filter(mu == min(mu) | mu == max(mu)),
  #                   global_climate[global_climate$class == 'Rainfall', ] |>
  #                   filter(mu == min(mu) | mu == max(mu))),
  #             mapping = aes(month1, mu, color = class)) +
  geom_ribbon(aes(fill = class), alpha = 0.3) +
  geom_hline(yintercept = 0, linetype = 3, linewidth = 0.25) +
  geom_vline(xintercept = c(ymd(c('2025-04-01', '2025-11-01'))),
             linetype = 2, linewidth = 0.3) +

```

```

annotate(geom = 'text', label = 'Rainy season',
        x = ymd('2025-07-15'), y = -2) +
scale_x_date(date_labels = '%b', date_breaks = '2 month') +
labs(y = 'z-scores', x = NULL) +
theme_classic() +
facet_wrap(~ type) +
scale_color_manual(values = c("steelblue", "tan1")) +
scale_fill_manual(values = c("steelblue", "tan1")) +
theme(panel.grid = element_blank(),
      legend.title = element_blank(),
      legend.position = c(0.5, 0.95),
      legend.direction = 'horizontal',
      strip.background = element_blank(),
      axis.line = element_line(linewidth = 0.25),
      axis.ticks = element_line(linewidth = 0.25),
      #text = element_text(family = 'Times New Roman'),
      legend.background = element_blank(),
      #axis.text.x = element_text(angle = 45, hjust = 1, vjust = 1),
      axis.text = element_text(size = 8.5),
      #title = element_blank(),
      legend.margin = margin(c(0, 0, 0, 0)))

```

Warning: A numeric `legend.position` argument in `theme()` was deprecated in ggplot2 3.5.0.  
 i Please use the `legend.position.inside` argument of `theme()` instead.

plot\_global\_climate

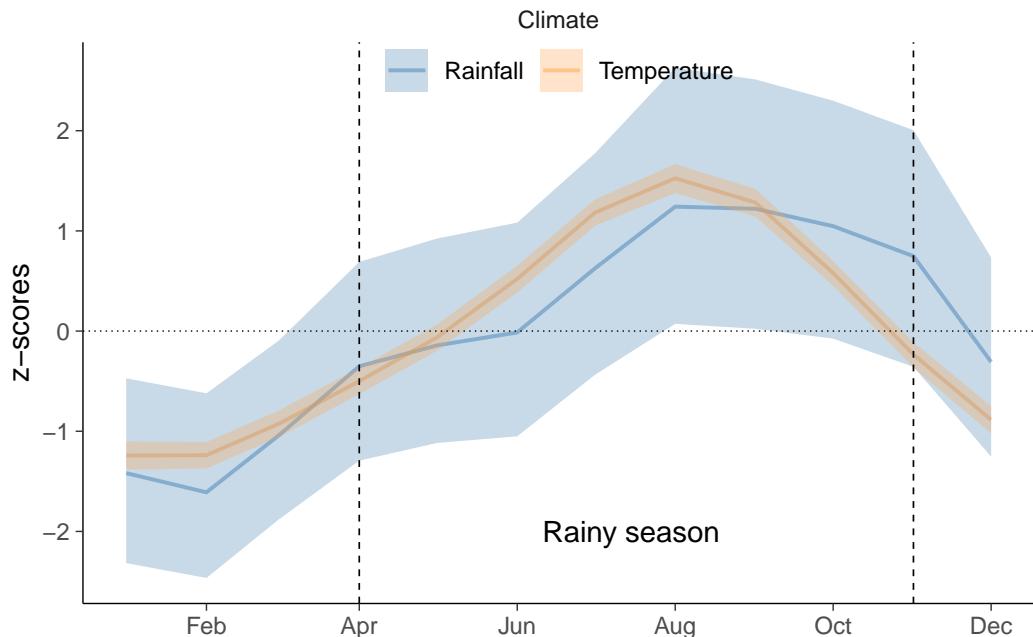


Figure S10: Yearly climatic pattern in Oahu island, Hawaii

## S4.2 Phenology

```
global_phenology <-
  lapply(1:2, FUN =
    function(x) {

      f_ <- c('f_fruitAB[', 'f_fruitS[')
      alpha <- c('alpha_fruitAB', 'alpha_fruitS')
      s <- grep(alpha[x], colnames(post_H2_G$alpha))
      est <- lapply(1:12, FUN =
        function(j) {
          class = c('Fruit production',
                    'Fruit richness')
          f_ <- paste0(f_[x], j, ']')
          m <-
            apply(post_H2_G$alpha[, s], 1, mean) +
            post_H2_G$f[[f_]] +
            post_H2_G$beta$beta_temp_fruitAB * mean(dat$z_temperature) +
            post_H2_G$beta$beta_rain_fruitAB * mean(dat$z_rainfall)
            # post_H2_G$beta$beta_fruitAB_H2 * mean(dat$fruit_abun) +
            # post_H2_G$beta$beta_fruits_S_H2 * mean(dat$fruit_S) +
            # post_H2_G$beta$beta_nut_H2 * mean(dat$HV_plants_nut) +
            # post_H2_G$beta$beta_mor_H2 * mean(dat$HV_plants_morfo) +
            # post_H2_G$beta$beta_PLAN_H2 * mean(dat$HV_plant) +
            # post_H2_G$beta$beta_BIRD_H2 * mean(dat$HV_bird)
            set.seed(123)
            m <- sample(m, 1e3)
            tibble(est = m,
                   month = j,
                   class = class[x])
        })
      est <- do.call('rbind', est)
    }
  )
  est
}

global_phenology <-
  do.call('rbind', global_phenology)

month_abbr <- na.omit(ymd(paste0("2023-", global_phenology$month, "-01"),
                           label = TRUE, abbr = TRUE))
```

Warning: 2 failed to parse.

```
global_phenology$month1 <- month_abbr
global_phenology |>
  group_by(class, month1) |>
  transmute(mu = mean(est),
```

```

    li = quantile(est, 0.1),
    ls = quantile(est, 0.9),
    type = 'Fruiting phenology',
    month = month) |>
unique() |>
filter(class == 'Fruit richness')

```

```

# A tibble: 12 x 7
# Groups:   class, month1 [12]
  class      month1       mu      li      ls type     month
  <chr>     <date>     <dbl>    <dbl>    <dbl> <chr>    <int>
1 Fruit richness 2023-01-01 -0.0437   -0.211   0.0993 Fruiting phenology 1
2 Fruit richness 2023-02-01 -0.0237   -0.188   0.126  Fruiting phenology 2
3 Fruit richness 2023-03-01  0.0000547 -0.139   0.140  Fruiting phenology 3
4 Fruit richness 2023-04-01  0.0241   -0.110   0.168  Fruiting phenology 4
5 Fruit richness 2023-05-01  0.0524   -0.0792  0.198  Fruiting phenology 5
6 Fruit richness 2023-06-01  0.0907   -0.0567  0.266  Fruiting phenology 6
7 Fruit richness 2023-07-01  0.122    -0.0479  0.333  Fruiting phenology 7
8 Fruit richness 2023-08-01  0.119    -0.0530  0.343  Fruiting phenology 8
9 Fruit richness 2023-09-01  0.0765   -0.0746  0.268  Fruiting phenology 9
10 Fruit richness 2023-10-01 0.0166   -0.125   0.158  Fruiting phenology 10
11 Fruit richness 2023-11-01 -0.0288  -0.178   0.108  Fruiting phenology 11
12 Fruit richness 2023-12-01 -0.0446  -0.201   0.0947 Fruiting phenology 12

```

```

dat_mod_pheno <- readRDS('data_phenology_model.rds')

plot_global_pheno <-
  global_pheno |>
  group_by(class, month1) |>
  transmute(mu = mean(est),
            li = quantile(est, 0.1),
            ls = quantile(est, 0.9),
            type = 'Fruiting phenology') |>
unique() |>
ggplot(aes(month1, mu, ymin = li, ymax = ls)) +
  geom_line(aes(color = class), linewidth = 0.7, alpha = 0.5) +
  geom_ribbon(aes(fill = class), alpha = 0.3) +
  geom_hline(yintercept = 0, linetype = 3, linewidth = 0.25) +
  geom_vline(xintercept = c(ymd(c('2023-04-01', '2023-11-01'))),
             linetype = 2, linewidth = 0.3) +
  scale_x_date(date_labels = '%b', date_breaks = '2 month') +
  labs(y = 'z-scores', x = NULL) +
  theme_classic() +
  facet_wrap(~ type) +
  scale_color_manual(values = c("tan1", "steelblue")) +
  scale_fill_manual(values = c("tan1", "steelblue")) +
  theme(panel.grid = element_blank(),
        legend.title = element_blank(),
        legend.position = c(0.5, 0.95),
        legend.direction = 'horizontal',
        strip.background = element_blank(),
        axis.line = element_line(linewidth = 0.25),
        axis.ticks = element_line(linewidth = 0.25),
        #text = element_text(family = 'Times New Roman'),

```

```

#axis.text.x = element_text(angle = 45, hjust = 1, vjust = 1),
#axis.text = element_text(size = 8.5),
#title = element_blank(),
legend.margin = margin(c(0, 0, 0, 0)))

```

plot\_global\_phenology

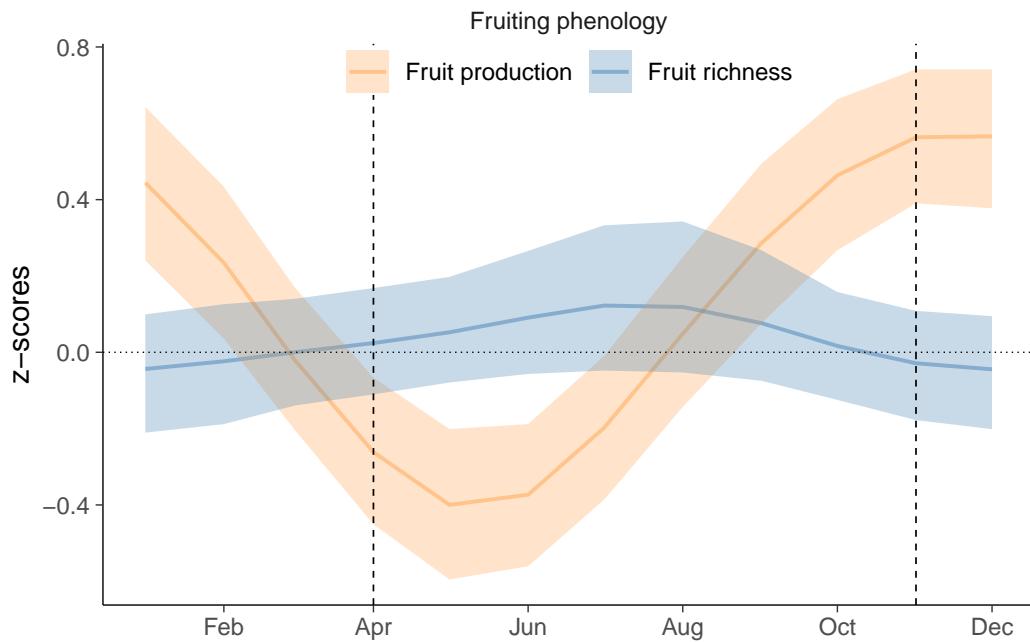


Figure S11: Temporal dynamics of plants fruiting phenology in Oahu island, Hawaii

#### S4.2.1 Statistical differences

##### S4.2.1.1 Species richness

Contrast between months with minimum and maximum values across the year.

```

## species richness

# maximum species number
max_phenologyS_std <-
  global_phenology |>
    filter(class == 'Fruit richness' &
           month >= 7 & month <= 8) # July and August

max_phenologyS <-
  max_phenologyS_std$est + (mean(d$fruit_S) * sd(d$fruit_S))

median(max_phenologyS) # average maximum

```

```
[1] 13.24518
```

```
quantile(max_phenologyS, 0.025) # low boundary CI
```

2.5%  
13.02998

```
quantile(max_phenologyS, 0.975) # upper boundary CI
```

97.5%  
13.64132

```
min_phenologyS_std <-  
  global_phenology |>  
  filter(month >= 2 & month <=3) |>  
  filter(class == 'Fruit richness')  
  
min_phenologyS <-  
  min_phenologyS_std$est + mean(d$fruit_S) * sd(d$fruit_S)  
  
median(min_phenologyS) # average maximum
```

```
[1] 13.14543
```

```
quantile(min_phenologyS, 0.025) # low boundary CI
```

2.5%  
12.8577

```
quantile(min_phenologyS, 0.975) # upper boundary CI
```

97.5%  
13.36183

```
set.seed(1)  
mean(max_phenologyS > # P(max > high)  
  sample(min_phenologyS, length(max_phenologyS)))
```

```
[1] 0.745
```

#### S4.2.1.2 Fruit production

Contrast between months with minimum and maximum values across the year.

```

# maximum fruit production
max_phenologyABU_std <-
  global_phenology |>
    filter(class == 'Fruit production' &
      month >= 11 & month <= 12) # November and December

max_phenologyABU <-
  exp(max_phenologyABU_std$est +
    (mean(dat_mod_pheno$abun) * sd(dat_mod_pheno$abun)))

median(max_phenologyABU) # average maximum

```

[1] 1261.782

```
quantile(max_phenologyABU, 0.025) # lower boundary CI
```

2.5%  
946.8521

```
quantile(max_phenologyABU, 0.975) # upper boundary CI
```

97.5%  
1675.593

```

min_phenologyABU_std <-
  global_phenology |>
    filter(month >= 5 & month <= 6) |>
    filter(class == 'Fruit production')

min_phenologyABU <-
  exp(min_phenologyABU_std$est +
    (mean(dat_mod_pheno$abun) * sd(dat_mod_pheno$abun)))

median(min_phenologyABU) # Average minimum

```

[1] 487.4656

```
quantile(min_phenologyABU, 0.025) # lower boundary CI
```

2.5%  
357.8064

```
quantile(min_phenologyABU, 0.975) # upper boundary CI
```

97.5%  
653.9503

```
tot_fru_increase <- 100 - (min_phenologyABU * 100) / max_phenologyABU  
median(tot_fru_increase) # Average contrast
```

[1] 61.3746

```
quantile(tot_fru_increase, 0.025) # lower boundary CI
```

2.5%  
39.99767

```
quantile(tot_fru_increase, 0.975) # upper boundary CI
```

97.5%  
74.86027

```
mean(max_phenologyABU > min_phenologyABU) # P(higher > lower)
```

[1] 1

### S4.3 Interaction functional space

```
global_hypervolumes <-  
  lapply(1:4, FUN =  
    function(x) {  
  
      class <- c("Fruits' nutr. hypervolume",  
              "Fruits' morph. hypervolume",  
              "Fruits' total hypervolume",  
              "Birds' hypervolume")  
  
      f_ <- c('f_nut_HV[', 'f_mor_HV[', 'f_PLAN_HV[', 'f_BIRD_HV[')  
  
      alpha <- c('alpha_nut_HV', 'alpha_mor_HV',  
                'alpha_PLAN_HV', 'alpha_BIRD_HV')  
  
      s <- grep(alpha[x], colnames(post_H2_G$alpha))  
  
      if (f_[x] == 'f_nut_HV[') {  
  
        est <- lapply(1:12, FUN =  
          function(j) {  
  
            f_ <- paste0(f_[x], j, ']')  
  
            m <-  
              apply(post_H2_G$alpha[, s], 1, mean) +  
              post_H2_G$f[[f_]] +  
              post_H2_G$alpha[s, x]  
          })  
  
        global_hypervolumes[[x]] <- est  
      }  
    })
```

```

post_H2_G$beta$beta_temp_nut_HV * mean(dat$z_temperature) +
post_H2_G$beta$beta_rain_nut_HV * mean(dat$z_rainfall) +
post_H2_G$beta$beta_fruitAB_nut_HV * mean(dat$fruit_abun) +
post_H2_G$beta$beta_fruitS_nut_HV * mean(dat$fruit_S) +
#post_H2_G$beta$beta_nut_H2 * mean(dat$HV_plants_nut) +
# post_H2_G$beta$beta_mor_H2 * mean(dat$HV_plants_morfo) +
# post_H2_G$beta$beta_PLAN_H2 * mean(dat$HV_plant) +
post_H2_G$beta$beta_BIRD_nut_HV * mean(dat$HV_bird)
set.seed(123)
m <- sample(m, 1e3)

tibble(est = m,
      month = j,
      class = class[x])

})
est <- do.call('rbind', est)

est
} else if (f_[x] == 'f_mor_HV[') {

est <- lapply(1:12, FUN =
  function(j) {

  f_ <- paste0(f_[x], j, ']')

  m <-
    apply(post_H2_G$alpha[, s], 1, mean) +
    post_H2_G$f[[f_]] +
    post_H2_G$beta$beta_temp_mor_HV * mean(dat$z_temperature) +
    post_H2_G$beta$beta_rain_mor_HV * mean(dat$z_rainfall) +
    post_H2_G$beta$beta_fruitAB_mor_HV * mean(dat$fruit_abun) +
    post_H2_G$beta$beta_fruitS_mor_HV * mean(dat$fruit_S) +
    #post_H2_G$beta$beta_nut_H2 * mean(dat$HV_plants_nut) +
    # post_H2_G$beta$beta_mor_H2 * mean(dat$HV_plants_morfo) +
    # post_H2_G$beta$beta_PLAN_H2 * mean(dat$HV_plant) +
    post_H2_G$beta$beta_BIRD_mor_HV * mean(dat$HV_bird)
  set.seed(123)
  m <- sample(m, 1e3)

  tibble(est = m,
        month = j,
        class = class[x])

})
est <- do.call('rbind', est)

est
} else if (f_[x] == 'f_PLAN_HV[') {

est <- lapply(1:12, FUN =
  function(j) {

  f_ <- paste0(f_[x], j, ']')

  m <-
    apply(post_H2_G$alpha[, s], 1, mean) +
    post_H2_G$f[[f_]] +
    post_H2_G$beta$beta_temp_PLAN_HV * mean(dat$z_temperature) +
    post_H2_G$beta$beta_rain_PLAN_HV * mean(dat$z_rainfall) +
    post_H2_G$beta$beta_fruitAB_PLAN_HV * mean(dat$fruit_abun) +
    post_H2_G$beta$beta_fruitS_PLAN_HV * mean(dat$fruit_S) +
    post_H2_G$beta$beta_nut_PLAN_HV * mean(dat$HV_plants_nut) +
    post_H2_G$beta$beta_mor_PLAN_HV * mean(dat$HV_plants_morfo) +
    #post_H2_G$beta$beta_PLAN_HV * mean(dat$HV_plant) +
    post_H2_G$beta$beta_BIRD_PLAN_HV * mean(dat$HV_bird)
  set.seed(123)
}

```

```

m <- sample(m, 1e3)

tibble(est = m,
       month = j,
       class = class[x])

})

est <- do.call('rbind', est)

est
} else if (f_[x] == 'f_BIRD_HV[') {

est <- lapply(1:12, FUN =
             function(j) {

f_ <- paste0(f_[x], j, ']')

m <-
  apply(post_H2_G$alpha[, s], 1, mean) +
  post_H2_G$f[[f_]] +
  post_H2_G$beta$temp_BIRD_HV * mean(dat$z_temperature) +
  post_H2_G$beta$beta_rain_BIRD_HV * mean(dat$z_rainfall) +
  post_H2_G$beta$beta_fruitAB_BIRD_HV * mean(dat$fruit_abun) +
  post_H2_G$beta$beta_fruitS_BIRD_HV * mean(dat$fruit_S)
# post_H2_G$beta$beta_nut_PLAN_HV * mean(dat$HV_plants_nut) +
# post_H2_G$beta$beta_mor_PLAN_HV * mean(dat$HV_plants_morfo) +
#post_H2_G$beta$beta_PLAN_HV * mean(dat$HV_plant) +
#post_H2_G$beta$beta_BIRD_PLAN_HV * mean(dat$HV_bird)
set.seed(123)
m <- sample(m, 1e3)

tibble(est = m,
       month = j,
       class = class[x])

})
est <- do.call('rbind', est)

est
}

global_hypervolumes <-
do.call('rbind', global_hypervolumes)

month_abbr <- na.omit(ymd(paste0("2023-", global_hypervolumes$month, "-01"),
                           label = TRUE, abbr = TRUE))

```

Warning: 2 failed to parse.

```

global_hypervolumes$month1 <- month_abbr

plot_global_hypervolume <-
lapply(unique(global_hypervolumes$class), FUN =
      function(x) {
        p <-
          global_hypervolumes |>
            filter(class == x) |>
            group_by(class, month1) |>
            transmute(mu = mean(est),
                      li = quantile(est, 0.1),

```

```

      ls = quantile(est, 0.9)) |>
unique() |>
ggplot(aes(month1, mu, ymin = li, ymax = ls)) +
geom_line(aes(), linewidth = 0.7, alpha = 0.5, color = 'lightblue3') +
geom_ribbon(aes(), alpha = 0.3, fill = 'lightblue3') +
geom_hline(yintercept = 0, linetype = 3, linewidth = 0.25) +
geom_vline(xintercept = c(ymd(c('2023-04-01', '2023-11-01'))),
linetype = 2, linewidth = 0.3) +
scale_x_date(date_labels = '%b', date_breaks = '2 month') +
labs(y = 'z-scores', x = NULL) +
facet_wrap(~class) +
#lims(y = c(-0.8, 1)) +
theme_classic() +
#scale_color_manual(values = c("tomato", "steelblue", "goldenrod")) +
#scale_fill_manual(values = c("tomato", "steelblue", "goldenrod")) +
theme(panel.grid = element_blank(),
legend.title = element_blank(),
legend.position = 'top',
strip.background = element_blank(),
axis.line = element_line(linewidth = 0.25),
axis.ticks = element_line(linewidth = 0.25),
#text = element_text(family = 'Times New Roman'),
axis.text.x = element_text(angle = 45, hjust = 1, vjust = 1),
axis.text = element_text(size = 8.5),
#title = element_blank(),
legend.margin = margin(c(0, 0, 0, 0)))
)
^p

plot_global_hypervolume <-
plot_grid(plot_global_hypervolume[[1]] +
  lims(y = c(-0.75, 0.95)),
plot_global_hypervolume[[2]] +
  theme(axis.title.y = element_blank()),
plot_global_hypervolume[[3]] +
  lims(y = c(-0.15, 0.19)) +
  theme(axis.title.y = element_blank()),
plot_global_hypervolume[[4]] +
  lims(y = c(-0.5, 0.48)) +
  theme(axis.title.y = element_blank()),
nrow = 1,
rel_widths = c(1, rep(0.95, 3)),
labels = c('i', 'ii', 'iii', 'iv'),
#label_fontfamily = 'Times New Roman',
label_fontface = 'plain',
label_size = 12,
label_y = 0.825,
label_x = c(0.3, 0.25, 0.23, 0.25))

global_hypervolumes <- split(global_hypervolumes, global_hypervolumes$class)

```

```
plot_global_hypervolume
```

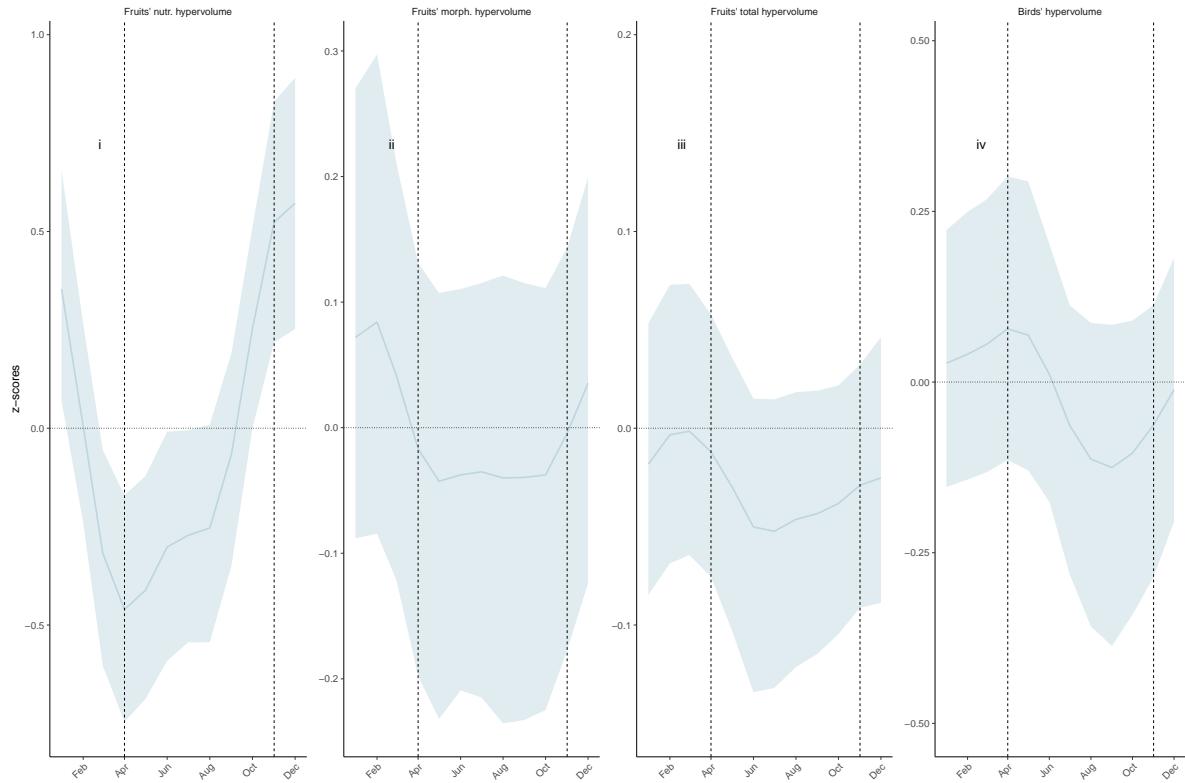


Figure S12: Temporal dynamics of different aspects of the interaction functional space of seed dispersal networks in Oahu island, Hawaii

#### S4.3.1 Statistical differences

##### S4.3.1.1 Fruits' nutritional functional space

Contrast between months with minimum and maximum values across the year.

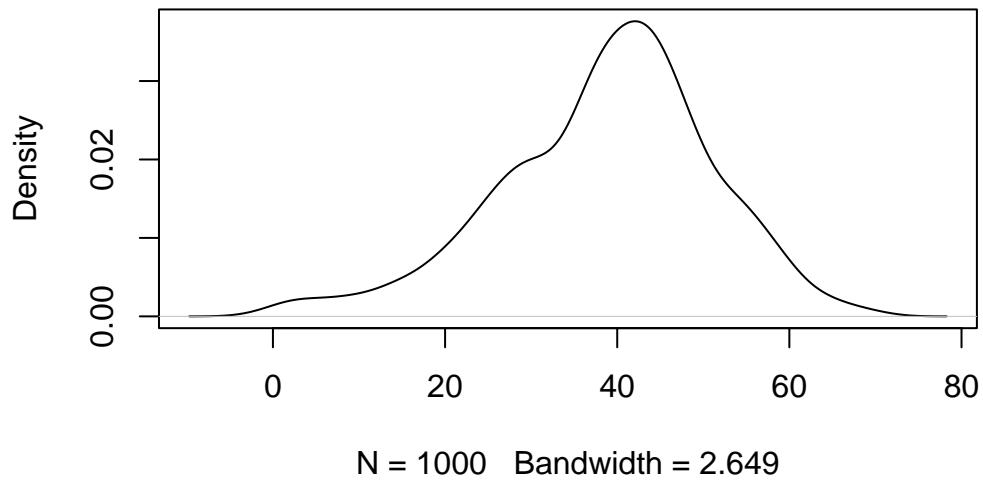
```
max_nut_HV <-
  global_hypervolumes$`Fruits' nutr. hypervolume` |>
  filter(month1 == '2023-12-01') |>
  mutate(est = mean(d$HV_plants_nut) + est * sd(d$HV_plants_nut))

min_nut_HV <-
  global_hypervolumes$`Fruits' nutr. hypervolume` |>
  filter(month1 == '2023-04-01') |>
  mutate(est = mean(d$HV_plants_nut) + est * sd(d$HV_plants_nut))

cont_nut_MAX_MIN <- 100 - (min_nut_HV$est * 100) / max_nut_HV$est

plot(density(cont_nut_MAX_MIN))
```

**density(x = cont\_nut\_MAX\_MIN)**



```
mean(cont_nut_MAX_MIN) # mean contrast
```

```
[1] 38.69988
```

```
quantile(cont_nut_MAX_MIN, 0.025) # lower boundary CI
```

```
2.5%  
9.728039
```

```
quantile(cont_nut_MAX_MIN, 0.975) # lower boundary CI
```

```
97.5%  
59.99575
```

#### S4.3.1.2 Fruits' morphological functional space

Contrast between months with minimum and maximum values across the year.

```

max_mor_HV <-
  global_hypervolumes$`Fruits' morph. hypervolume` |>
  filter(month1 >= '2023-01-01' & month1 <= '2023-03-01' |
    month1 == '2023-12-01') |>
  mutate(est = mean(sqrt(d$HV_plants_morfo)) +
    est * sd(sqrt(d$HV_plants_morfo)))

min_mor_HV <-
  global_hypervolumes$`Fruits' morph. hypervolume` |>
  filter(month1 >= '2023-04-01' & month1 <= '2023-11-01') |>
  mutate(est = mean(sqrt(d$HV_plants_morfo)) +
    est * sd(sqrt(d$HV_plants_morfo)))

cont_mor_MAX_MIN <- 100 - (min_mor_HV$est * 100) / max_mor_HV$est
mean(cont_mor_MAX_MIN) # mean contrast

```

[1] 1.923834

```
quantile(cont_mor_MAX_MIN, 0.025) # lower boundary CI
```

2.5%  
-5.414152

```
quantile(cont_mor_MAX_MIN, 0.975) # upper boundary CI
```

97.5%  
13.53737

```
mean(cont_mor_MAX_MIN > 0) # P(higher > lower)
```

[1] 0.665875

#### S4.3.1.3 Fruits' functional space (total)

Contrast between months with minimum and maximum values across the year.

```

max_PLAN_HV <-
  global_hypervolumes$`Fruits' total hypervolume` |>
  filter(month1 >= '2023-02-01' & month1 <= '2023-03-01') |>
  mutate(est = mean(d$HV_plants_morfo) +
    est * sd(d$HV_plants_morfo))

min_PLAN_HV <-
  global_hypervolumes$`Fruits' total hypervolume` |>
  filter(month1 >= '2023-06-01' & month1 <= '2023-07-01') |>
  mutate(est = mean(d$HV_plants_morfo) +
    est * sd(d$HV_plants_morfo))

cont_PLAN_MAX_MIN <- 100 - (min_PLAN_HV$est * 100) / max_PLAN_HV$est
mean(cont_PLAN_MAX_MIN) # mean contrast

```

```
[1] 2.172868
```

```
quantile(cont_PLAN_MAX_MIN, 0.025) # lower boundary CI
```

```
2.5%
-3.933065
```

```
quantile(cont_PLAN_MAX_MIN, 0.975) # upper boundary CI
```

```
97.5%
12.16609
```

```
mean(cont_PLAN_MAX_MIN > 0) # P(higher > lower)
```

```
[1] 0.704
```

#### S4.3.1.4 Birds' functional space

```
max_BIRD_HV <-
  global_hypervolumes`Birds' hypervolume` |>
  filter(month1 == '2023-04-01') |>
  mutate(est = mean(d$HV_plants_morfo) +
    est * sd(d$HV_plants_morfo))

min_BIRD_HV <-
  global_hypervolumes`Birds' hypervolume` |>
  filter(month1 >= '2023-08-01' & month1 <= '2023-09-01') |>
  mutate(est = mean(d$HV_plants_morfo) +
    est * sd(d$HV_plants_morfo))

cont_BIRD_MAX_MIN <- 100 - (min_BIRD_HV$est * 100) / max_BIRD_HV$est
mean(cont_BIRD_MAX_MIN) # mean contrast
```

```
[1] 8.127236
```

```
quantile(cont_BIRD_MAX_MIN, 0.025) # lower boundary CI
```

```
2.5%
-9.113396
```

```
quantile(cont_BIRD_MAX_MIN, 0.975) # upper boundary CI
```

```
97.5%
35.69932
```

```
mean(cont_BIRD_MAX_MIN > 0) # P(higher > lower)
```

```
[1] 0.7695
```

## S4.4 Network dynamics

```
colnames(estimated_nestednessG_effect)[4] <- 'month1'  
global_time_NM <-  
  rbind(estimated_MODU_G_effect,  
        estimated_nestednessG_effect,  
        estimated_H2_G_effect)  
global_time_NM$class <- as.factor(global_time_NM$class)  
levels(global_time_NM$class)
```

```
[1] "H2"      "mNSSu"   "MODU"
```

```
global_time_NM$class <- factor(global_time_NM$class,  
                                labels = c("H2", "Nestedness", "Modularity"))  
  
##### plot network metrics #####  
plot_global_NM <-  
  lapply(levels(global_time_NM$class), FUN =  
         function(x) {  
  
    p <-  
      global_time_NM |>  
        filter(class == x) |>  
        group_by(class, month1) |>  
        transmute(mu = mean(est),  
                  li = quantile(est, 0.1),  
                  ls = quantile(est, 0.9)) |>  
        unique() |>  
        ggplot(aes(month1, mu, ymin = li, ymax = ls)) +  
          geom_line(aes(), linewidth = 0.7, alpha = 0.5, color = 'lightblue3') +  
          geom_ribbon(aes(), alpha = 0.3, fill = 'lightblue3') +  
          geom_hline(yintercept = 0, linetype = 3, linewidth = 0.25) +  
          geom_vline(xintercept = c(ymd(c('2023-04-01', '2023-11-01'))),  
                     linetype = 2, linewidth = 0.3) +  
          scale_x_date(date_labels = '%b', date_breaks = '2 month') +  
          labs(y = 'z-scores', x = NULL) +  
          facet_wrap(~class) +  
          theme_classic() +  
          #scale_color_manual(values = c("tomato", "steelblue", "goldenrod")) +  
          #scale_fill_manual(values = c("tomato", "steelblue", "goldenrod")) +  
          theme(panel.grid = element_blank(),  
                legend.title = element_blank(),  
                legend.position = 'top',  
                strip.background = element_blank(),  
                axis.line = element_line(linewidth = 0.25),  
                axis.ticks = element_line(linewidth = 0.25),  
                #text = element_text(family = 'Times New Roman'),  
                #axis.text.x = element_text(angle = 45, hjust = 1, vjust = 1),
```

```

        axis.text = element_text(size = 8.5),
        #title = element_blank(),
        legend.margin = margin(c(0, 0, 0, 0)))
    }) p

plot_global_network_metrics <-
  plot_grid(plot_global_NM[[1]],
            plot_global_NM[[2]] +
              theme(axis.title.y = element_blank()),
            plot_global_NM[[3]] +
              theme(axis.title.y = element_blank()) +
              lims(y = c(-0.65, 0.8)),
            nrow = 1,
            rel_widths = c(1, 0.95, 0.95),
            labels = c('i', 'ii', 'iii'),
            #label_fontfamily = 'Times New Roman',
            label_fontface = 'plain',
            label_size = 12,
            label_y = 0.825,
            label_x = c(0.21, 0.185, 0.185))

global_time_NM <- split(global_time_NM, global_time_NM$class)

```

```
plot_global_network_metrics
```

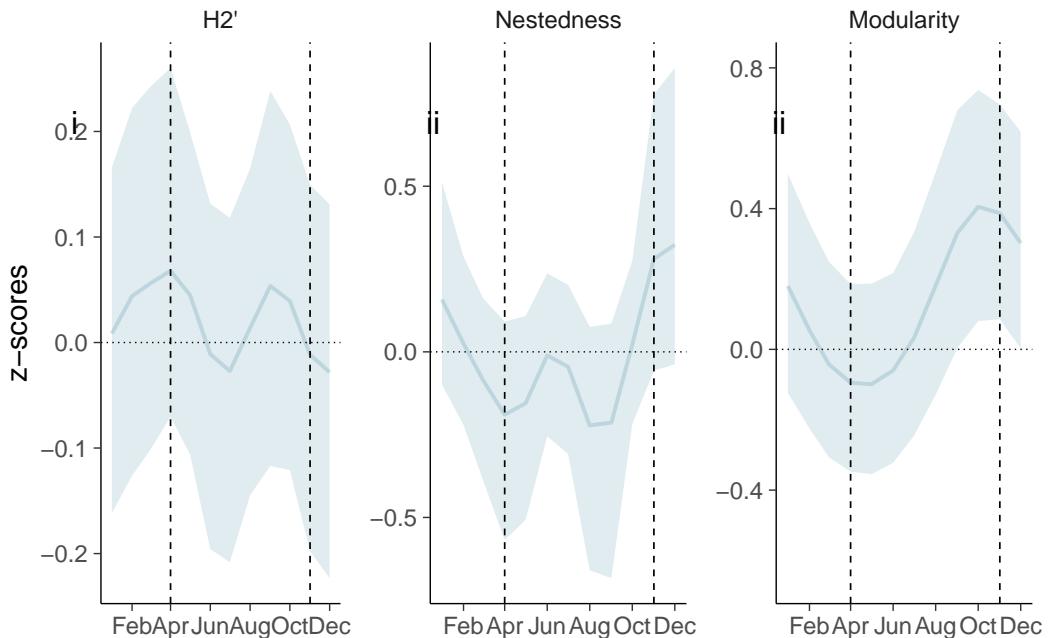


Figure S13: Temporal dynamics of the structure of seed dispersal networks in Oahu island, Hawaii

## S4.4.1 Statistical differences

### S4.4.1.1 Nestedness

Contrast between months with minimum and maximum values across the year.

```
max_NSS <-
  global_time_NM$Nestedness |>
  filter(month == 12) |> # December
  mutate(est = mean(d$nestedness) +
    est * sd(d$nestedness))

min_NSS <-
  global_time_NM$Nestedness |>
  filter(month >= 8 & month <= 9) |> # August and September
  mutate(est = mean(d$nestedness) +
    est * sd(d$nestedness))

max_NSS <-
  global_time_NM$Nestedness |>
  filter(month == 12) |>
  mutate(est = mean(d$nestedness) +
    est * sd(d$nestedness))

min_NSS <-
  global_time_NM$Nestedness |>
  filter(month >= 8 & month <= 9) |>
  mutate(est = mean(d$nestedness) +
    est * sd(d$nestedness))

cont_NSS_MIN_MAX <-
  100 - ((min_NSS$est * 100) / max_NSS$est)

mean(cont_NSS_MIN_MAX) # average difference
```

[1] 23.4137

```
quantile(cont_NSS_MIN_MAX, 0.025) # low boundary of CI
```

2.5%  
-7.655357

```
quantile(cont_NSS_MIN_MAX, 0.975) # upper boundary of CIU
```

97.5%  
71.97891

```
mean(cont_NSS_MIN_MAX > 0) # P(higher month > lower month)
```

[1] 0.8235

#### S4.4.1.2 Modularity

Contrast between months with minimum and maximum values across the year.

```
max_MODU <-
  global_time_NM$Modularity |>
  filter(month >= 10 & month <= 11) |>
  mutate(est = mean(d$modularity) +
    est * sd(d$modularity))

min_MODU <-
  global_time_NM$Modularity |>
  filter(month >= 4 & month <= 5) |>
  mutate(est = mean(d$modularity) +
    est * sd(d$modularity))

unique(min_MODU$month)
```

[1] 4 5

```
cont_MODU_MIN_MAX <- max_MODU$est - min_MODU$est
#100 - ((min_MODU$est * 100) / max_MODU$est)

mean(cont_MODU_MIN_MAX) * 100 # average difference
```

[1] 6.057846

```
quantile(cont_MODU_MIN_MAX, 0.025) * 100 # low boundary of CI
```

2.5%
-1.441087

```
quantile(cont_MODU_MIN_MAX, 0.975) * 100 # upper boundary of CIU
```

97.5%
14.03739

```
mean(cont_MODU_MIN_MAX > 0) # P(higher month > lower month)
```

[1] 0.9385

#### S4.4.1.3 $H'_2$

Contrast between months with minimum and maximum values across the year.

```

max_h2 <-
  global_time_NM$`H2` `|>
  filter(month == 4 | month == 9) `|>
  mutate(est = mean(d$H2) +
    est * sd(d$H2))

min_h2 <-
  global_time_NM$`H2` `|>
  filter(month == 12 | month == 7) `|>
  mutate(est = mean(d$H2) +
    est * sd(d$H2))

cont_h2_MIN_MAX <- max_h2$est - min_h2$est
#100 - ((min_h2$est * 100) / max_h2$est)

mean(cont_h2_MIN_MAX) * 100 # average difference

```

[1] 2.640817

```
quantile(cont_h2_MIN_MAX, 0.025) * 100 # low boundary of CI
```

2.5%  
-7.321328

```
quantile(cont_h2_MIN_MAX, 0.975) * 100 # upper boundary of CIU
```

97.5%  
21.16966

```
mean(cont_h2_MIN_MAX > 0) # P(higher month > lower month)
```

[1] 0.6245

## S4.5 Figure 2

```

plot_grid(plot_grid(NULL,
  plot_global_climate +
    theme(legend.background = element_blank()),
  plot_global_phenology +
    theme(axis.title.y = element_blank(),
      legend.background = element_blank()),
  NULL, nrow = 1,
  rel_widths = c(0.2, 1, 0.975, 0.2),
  labels = c(' ', 'i', 'ii', ''),
  #label_fontfamily = 'Times New Roman',
  label_fontface = 'plain',
  label_size = 12,
  label_y = 0.85,

```

```
    label_x = c(0, 0.15, 0.13, 0)),
plot_global_hypervolume,
plot_global_network_metrics,
ncol = 1,
labels = paste0('(', letters, ')'),
#label_fontfamily = 'Times New Roman',
label_fontface = 'plain',
label_size = 12,
label_x = c(0, 0, 0))
```

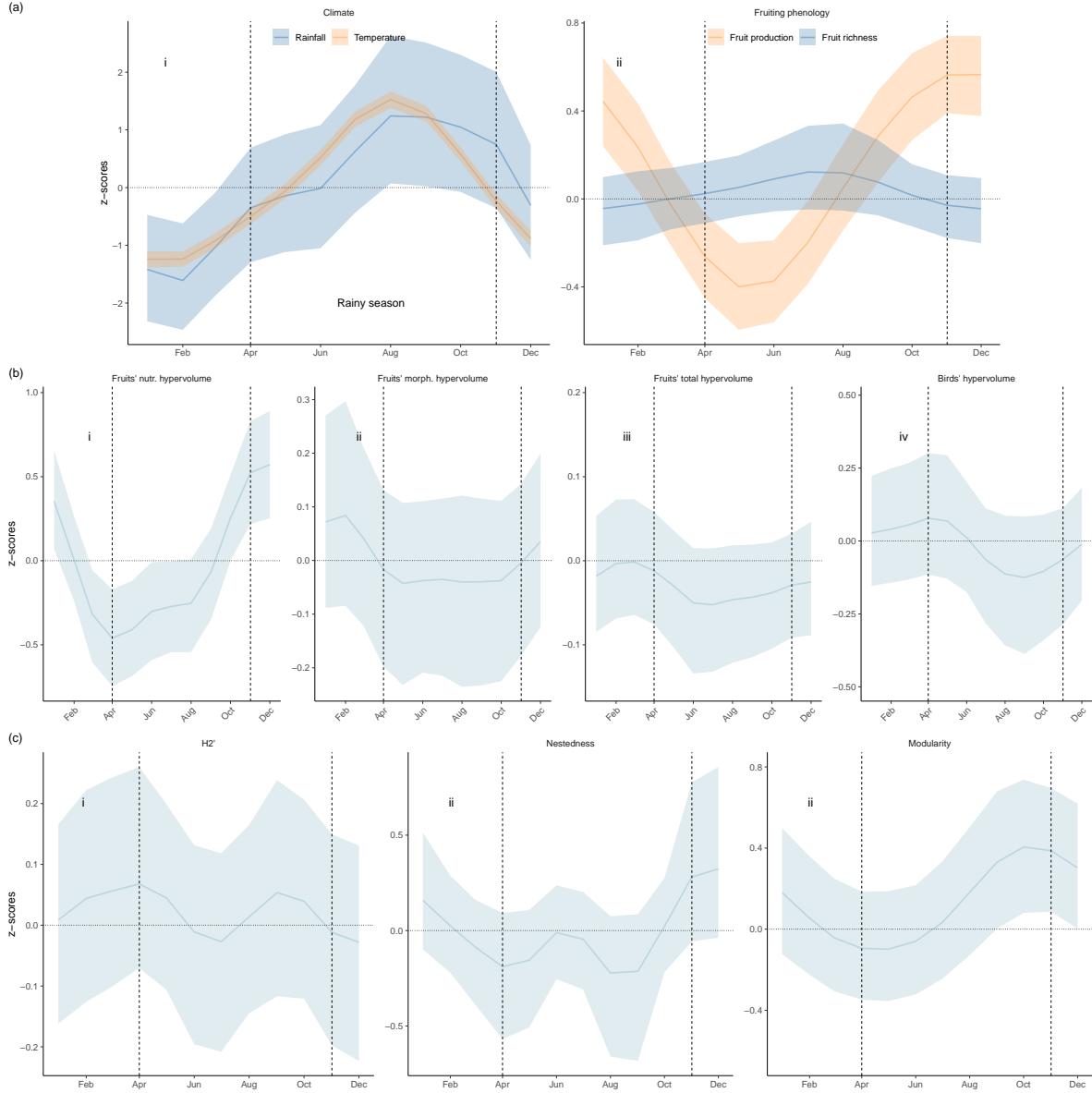


Figure S14: Monthly variation of the structure of seed dispersal networks in O'ahu island (Hawai'i), and biotic and abiotic factors that explain it. Each panel shows the temporal trend of the variable conditioned to average values of their continuous predictors and marginalizing site variation. (a-i) Rainfall and temperature patterns after marginalizing year variation. (a-ii) Fruiting phenology in terms of plant richness and fruit production (i.e., fruit abundance). (b-i to -iii) Nutritional, morphological and total (i.e., nutritional plus morphological) functional hypervolumes of fruiting plants integrating the seed dispersal networks. (b-iv) Functional hypervolume of frugivorous birds in the seed dispersal networks. (c-i to -iii) network metrics denoting the structure of the seed dispersal interactions. Vertical dashed lines denote the expected beginning and end of the rainy season (Chu and Chen 2005).

## S4.6 Temporal patterns: Expanded results

The lowest temperature and precipitation were recorded in January and February, respectively, whereas both peaked around August (Figure 2a-i). This pattern implied that, on average between 2014 and 2017, the wettest months at our sampling sites occurred during the summer season —from the beginning of April to the beginning of November. Therefore, hereafter, we will refer to the periods April–Octover and November–March as the wet and dry seasons, respectively (Figure 2).

The richness of fruit-producing plants remained stable throughout the year (Figure 2a-ii), with negligible differences ( $P(\text{highest} > \text{lowest}) = 0.74$ ) between peak (July–August: 13.2 [13.0–13.6] species) and lowest values (December–January: 13.1 [12.8–13.3] species). In contrast, fruit production exhibited a pronounced seasonal pattern, increasing after the beginning of the wet season in April and peaking between the end of the wet and beginning of the dry seasons (November) (Figure 2 a-ii). This was followed by a steep decline, reaching its lowest levels after the beginning of the wet season (May–June, Figure 2 a-ii). This pattern indicates that during peak fruit production, the forest produced 60% more fruit (60.8% [39.3–75.6]; 1268.5 [940.8–1707.1] fruits per 4.2 m<sup>2</sup>) compared to the lowest-production months (492.1 [358.9–654.7] fruits per 4.2 m<sup>2</sup>) (Appendix S5: Section 4.2.1).

The nutritional interaction functional space (IFS) in the seed dispersal network increased gradually during the wet season and showed a sharp decline during the dry season (from January to April, Figure 2b-i). Specifically, compared to the peak value in December, the nutritional IFS utilized by birds was ~40% smaller (39.1% [12.7–58.6]) by April, marking the end of the dry season. In contrast, the morphological IFS of fruits remained relatively stable throughout the year, with a modest increase during the dry season and the lowest values observed during the wet season (from April to November, Figure 2b-ii). It was 2% smaller (1.9% [−5.7–14.13]) during low periods compared to peak values, but these differences lacked statistical support ( $P(\text{highest} > \text{lowest}) = 0.64$ ). Similarly, the total IFS of fruits exhibited minimal annual variation (Figure 2b-iii), with a slight decrease in June–July and a peak in February–March, reflecting a 2% reduction (2.3% [−3.5–12.3]) in IFS size ( $P(\text{higher} > \text{lower}) = 0.71$ ). Birds' IFS followed a comparable trend, reaching its highest values at the onset of the wet season and its lowest in August–September (Figure 2b-iv). This represented an ~8% reduction (7.7% [−8.4–32.5]) in IFS size ( $P(\text{higher} > \text{lower}) = 0.75$ ) (Appendix S5: Section 4.3.1).

Compared to modularity, network specialization (H2') and nestedness exhibited less pronounced seasonal oscillations (Figure 2c). Specialization showed two minor peaks at the beginning and end of the wet season, with declines in July and December (Figure 2c-i), resulting in networks that were 3% less specialized (2.8% [−6.2–22.6]). However, this effect was highly variable and lacked strong statistical support ( $P(\text{higher} > \text{lower}) = 0.64$ ). Network nestedness peaked in December (early dry season) and reached its lowest values in August–September (Figure 2c-ii). These differences indicated that networks during low-nestedness periods were

23% less nested (23.4% [-7.6–71.9];  $P(\text{highest} > \text{lowest}) = 0.82$ ). The modularity of seed dispersal networks gradually increased throughout the wet season, peaking in October–November and reaching its lowest point in April–May (Figure 2c-iii). This pattern indicates that networks were 6% more modular (6.1% [-1.5–14.4]) during peak periods ( $P(\text{highest} > \text{lowest}) = 0.93$ ) (Appendix S5: Section 4.4.1).

## S4.7 Effects among variables: expanded results

### S4.7.1 Effects of climate and fruiting phenology on IFS

The interaction functional space (IFS) of birds within seed dispersal networks was modulated by changes in rainfall, temperature, and fruit production (Figure 3). Fruit production had a positive effect on birds' IFS ( $\beta = 0.24$  [-0.10–0.60],  $P(\beta < 0) = 0.91$ ), being 24.9% and 73.1% more important than the positive effect of temperature ( $\beta = 0.19$  [-0.11–0.50],  $P(\beta < 0) = 0.89$ ) and the negative effect of rainfall ( $\beta = -0.14$  [-0.48–0.20],  $P(\beta < 0) = 0.79$ ), respectively. The direct causal effects of rainfall, temperature and fruit production implied changes of birds' IFS by -8% (-7.8% [-26.8–11.3],  $P(\text{causal effect} > 0) = 0.79$ ), 10% (10.1% [-6.1–26.9],  $P(\text{causal effect} > 0) = 0.89$ ) and 12% (12.1% [-5.2–30.14],  $P(\text{causal effect} > 0) = 0.91$ ), respectively. Rainfall also exhibited an indirect negative effect mediated through fruit production, but this causal effect was negligible (Appendix S5: Section 4.6.2.4).

The nutritional IFS was directly affected by fruit production and temperature, and indirectly by rainfall (Figure 3). Fruit production showed a positive effect on fruits' nutritional IFS ( $\beta = 0.21$  [-0.16–0.60],  $P(\beta > 0) = 0.86$ ), though its relative importance was 61.5% lower than temperature's positive effect ( $\beta = 0.34$  [0.05–0.65],  $P(\beta > 0) = 0.99$ ). The causal effect translated to a 40% increase (40.2% [-29.2–110.2];  $P(\text{causal effect} > 0) = 0.86$ ) from fruit production and a 69% increase (68.8% [10.2–128.1];  $P(\text{causal effect} > 0) = 0.99$ ) from temperature in nutritional IFS. In contrast, the indirect effect of rainfall mediated through fruit production was negligible (causal effect < 1%) (Appendix S5: Section 4.6.3).

The fruits' morphological IFS was directly affected by fruit production and indirectly influenced by rainfall (Figure 3). Fruit production showed a positive effect ( $\beta = 0.20$  [-0.17–0.57],  $P(\beta > 0) = 0.85$ ), corresponding to a 34% increase (34% [-29.8–98.7];  $P(\text{causal effect} > 0) = 0.85$ ) in morphological IFS. In contrast, rainfall had a negative indirect effect mediated through fruit abundance, resulting in a 2% reduction (-2.3% [-20.2–6.8];  $P(\text{causal effect} < 0) = 0.73$ ) (Appendix S5: Section 4.6.4).

The total fruits' IFS was affected directly and indirectly by climatic, phenological and IFS factors (Figure 3). The main factors explaining fruits' IFS were their nutritional and morphological IFS (Appendix S5: section 4.6.5), which is natural due to the procedures to estimate total fruits IFS (see Methods section). Among the other variables, precipitation ( $\beta = 0.07$  [-0.06–0.2],  $P(\beta > 0) = 0.86$ ), richness of produced fruits ( $\beta = 0.07$  [-0.05–0.19],  $P(\beta > 0) = 0.89$ ), and birds' IFS ( $\beta = 0.03$  [-0.06–0.12],  $P(\beta > 0) = 0.75$ ) had positive effects, whereas

fruit production had a negative effect ( $\beta = -0.15$  [-0.29–0.001],  $P(\beta < 0) = 0.97$ ). On average, the effect of fruit production was 200 % ( $203.7 \pm 7.4$  %) more important compared to precipitation, richness of produced fruits and birds' IFS. The causal effect of fruit production implied a reduction of 45 % (-45.6 [-91.5–0.23] %) on total fruits IFS ( $P(\text{causal effect} < 0) = 0.97$ ), whilst the indirect effect of precipitation mediated by fruit production increased it by 2 % (2.2 [-3.1–11.3] %;  $P(\text{causal effect} > 0) = 0.81$ ) (Appendix S5: section 4.6.5).

#### S4.7.2 Biotic and abiotic predictors explaining network structure

The specialization (H2') and modularity of seed dispersal networks were primarily influenced by direct climatic effects (Figure 3). In contrast, nestedness was determined by more complex pathways involving direct and indirect effects of climate, fruiting phenology, and the IFS of both birds and fruits (Figure 3).

Network specialization showed a positive response to precipitation ( $\beta = 0.21$  [-0.07–0.50],  $P(\beta > 0) = 0.92$ ) but a negative to temperature ( $\beta = -0.10$  [-0.41–0.19],  $P(\beta < 0) = 0.75$ ). Precipitation's effect was 113.2% stronger than that of temperature. The causal analysis revealed that precipitation increased network specialization by 42.5% ([−16.5–89.2];  $P(\text{causal effect} > 0) = 0.92$ ), while temperature reduced it by 17.8% ([−77.2–37.5];  $P(\text{causal effect} < 0) = 0.75$ ) (Appendix S5: Section 4.6.9).

Network modularity showed positive responses to rainfall ( $\beta = 0.15$  [-0.12–0.39],  $P(\beta > 0) = 0.88$ ), nutritional IFS ( $\beta = 0.04$  [-0.05–0.14],  $P(\beta > 0) = 0.81$ ), and total fruit IFS ( $\beta = 0.04$  [-0.06–0.14],  $P(\beta > 0) = 0.78$ ) (Figure 3). However, rainfall's relative importance was 260% ( $261.8 \pm 3.9$ %) greater than other factors, with its direct causal effect potentially increasing modularity by 74.3% ([−46.7–222.7];  $P(\text{causal effect} > 0) = 0.89$ ). Additionally, nutritional and total fruit IFS showed smaller but significant effects, increasing modularity by 17.1% ([−20.7–55.5];  $P(\text{causal effect} > 0) = 0.82$ ) and 14.9% ([−21.9–53.8];  $P(\text{causal effect} > 0) = 0.78$ ), respectively (Appendix S5: Section 4.6.8).

Network nestedness showed negative responses to precipitation ( $\beta = -0.17$  [-0.50–0.16],  $P(\beta < 0) = 0.84$ ), fruit production ( $\beta = -0.22$  [-0.64–0.20],  $P(\beta < 0) = 0.85$ ), fruit IFS ( $\beta = -0.29$  [-0.78–0.19],  $P(\beta < 0) = 0.87$ ), and bird IFS ( $\beta = -0.08$  [-0.32–0.17],  $P(\beta < 0) = 0.74$ ), but positive responses to temperature ( $\beta = 0.18$  [-0.15–0.54],  $P(\beta > 0) = 0.84$ ) and morphological IFS ( $\beta = 0.22$  [-0.13–0.57],  $P(\beta > 0) = 0.74$ ) (Figure 3; Appendix S5: Section 4.6.7). The effect of fruit IFS was on average 92% stronger than those of climatic, phenological, and other IFS factors. Causal analysis revealed that fruit IFS and fruit production decreased network nestedness by 67.3% ([−185.1–46.3];  $P(\text{effect} < 0) = 0.88$ ) and 50.7% ([−141.7–44.7];  $P(\text{effect} < 0) = 0.85$ ), respectively, while morphological IFS increased it by 59.5% ([−35.4–154.3];  $P(\text{effect} > 0) = 0.89$ ) (Appendix S5: Section 4.6.7).

## S4.8 Paths of climatic effects

This section use the effects with  $P(effect) > 0.7$  to indicate all possible paths through which climatic bottom-up effects can affect temporal dynamics of seed dispersal networks

```

betas_MODU_G$p <-
  sapply(1:nrow(betas_MODU_G), FUN =
    function(x) {
      pos <- betas_MODU_G$p_pos[x]
      neg <- betas_MODU_G$p_neg[x]

      if (pos > neg) pos
      else neg
    })

betas_MODU_G$factor <- gsub('^(beta_)([a-zA-Z]*)(_)(.*)$', '\\\\2',
  betas_MODU_G$var)

betas_MODU_G$response <- gsub('^(beta_)([a-zA-Z]*)(_)(.*)$', '\\\\4',
  betas_MODU_G$var)

betas_H2_G$p <-
  sapply(1:nrow(betas_H2_G), FUN =
    function(x) {
      pos <- betas_H2_G$p_pos[x]
      neg <- betas_H2_G$p_neg[x]

      if (pos > neg) pos
      else neg
    })

betas_H2_G$factor <- gsub('^(beta_)([a-zA-Z]*)(_)(.*)$', '\\\\2',
  betas_H2_G$var)

betas_H2_G$response <- gsub('^(beta_)([a-zA-Z]*)(_)(.*)$', '\\\\4',
  betas_H2_G$var)

betas_nestednessG$p <-
  sapply(1:nrow(betas_nestednessG), FUN =
    function(x) {
      pos <- betas_nestednessG$p_pos[x]
      neg <- betas_nestednessG$p_neg[x]

      if (pos > neg) pos
      else neg
    })

betas_nestednessG$factor <- gsub('^(beta_)([a-zA-Z]*)(_)(.*)$', '\\\\2',
  betas_nestednessG$var)

betas_nestednessG$response <- gsub('^(beta_)([a-zA-Z]*)(_)(.*)$', '\\\\4',
  betas_nestednessG$var)

effects_net <-
  rbind(betas_H2_G[betas_H2_G$p >= 0.7,],
    betas_MODU_G[grep('MODU$', betas_MODU_G$var),],
    betas_nestednessG[grep('NSS$', betas_nestednessG$var),])
  
effects_net <- effects_net[, c("factor", "response", 'p')]

```

```

colnames(effects_net) <- c('from', 'to', 'weight')

effects_net$from[grep('^nut$', effects_net$from)] <- 'nut_HV'
effects_net$from[grep('^mor$', effects_net$from)] <- 'mor_HV'
effects_net$from[grep('^BIRD$', effects_net$from)] <- 'BIRD_HV'
effects_net$from[grep('^PLAN$', effects_net$from)] <- 'PLAN_HV'

vertex_df <-
  tibble(var = unique(c(effects_net$from, effects_net$to)),
         type = c('Climate', 'Climate', 'Phenology', 'Phenology',
                  'Plants', 'Plants', 'Birds', 'Plants', 'Network',
                  'Network', 'Network'),
         color = c('lightblue3', 'lightblue3', 'seagreen', 'seagreen',
                  'tomato3', 'tomato3', 'tan1', 'tomato3', 'cyan4',
                  'cyan4', 'cyan4'))
#effects_net <- effects_net[effects_net$weight >= 0.8, ]

net_plot <-
  graph_from_data_frame(effects_net, directed = T, vertices = vertex_df)

```

#### S4.8.1 Nestedness

Direct effects

```

sp <- lapply(1:2, FUN =
  function(x) {
    shortest_paths(net_plot, from = x, to = 11,
                   weights = E(net_plot)$weight)
  })

(shortest_path <- lapply(1:2, FUN =
  function(x) {
    sp[[x]]$vpath[[1]]
  }))

```

```

[[1]]
+ 2/11 vertices, named, from f0b35d9:
[1] temp NSS

[[2]]
+ 2/11 vertices, named, from f0b35d9:
[1] rain NSS

```

Bottom-up effects

```

all_paths_nestednes <- lapply(1:2, FUN =
  function(x) {
    all_simple_paths(net_plot, from = x, to = 11)
  })

path_lengths_nestednes <-
  lapply(all_paths_nestednes, FUN =

```

```

        function(i) {
            sapply(i, function(path) {
                sum(E(net_plot, path = path)$weight)
            })
    }
all_paths_nestednes

```

```

[[1]]
[[1]][[1]]
+ 4/11 vertices, named, from f0b35d9:
[1] temp      fruitS  PLAN_HV NSS

[[1]][[2]]
+ 4/11 vertices, named, from f0b35d9:
[1] temp      nut_HV  PLAN_HV NSS

[[1]][[3]]
+ 4/11 vertices, named, from f0b35d9:
[1] temp      BIRD_HV PLAN_HV NSS

[[1]][[4]]
+ 3/11 vertices, named, from f0b35d9:
[1] temp      BIRD_HV NSS

[[1]][[5]]
+ 2/11 vertices, named, from f0b35d9:
[1] temp NSS

[[2]]
[[2]][[1]]
+ 5/11 vertices, named, from f0b35d9:
[1] rain      fruitAB nut_HV  PLAN_HV NSS

[[2]][[2]]
+ 5/11 vertices, named, from f0b35d9:
[1] rain      fruitAB mor_HV  PLAN_HV NSS

[[2]][[3]]
+ 4/11 vertices, named, from f0b35d9:
[1] rain      fruitAB mor_HV  NSS

[[2]][[4]]
+ 5/11 vertices, named, from f0b35d9:

```

```

[1] rain      fruitAB BIRD_HV PLAN_HV NSS

[[2]][[5]]
+ 4/11 vertices, named, from f0b35d9:
[1] rain      fruitAB BIRD_HV NSS

[[2]][[6]]
+ 4/11 vertices, named, from f0b35d9:
[1] rain      fruitAB PLAN_HV NSS

[[2]][[7]]
+ 3/11 vertices, named, from f0b35d9:
[1] rain      fruitAB NSS

[[2]][[8]]
+ 4/11 vertices, named, from f0b35d9:
[1] rain      fruits  PLAN_HV NSS

[[2]][[9]]
+ 4/11 vertices, named, from f0b35d9:
[1] rain      BIRD_HV PLAN_HV NSS

[[2]][[10]]
+ 3/11 vertices, named, from f0b35d9:
[1] rain      BIRD_HV NSS

[[2]][[11]]
+ 3/11 vertices, named, from f0b35d9:
[1] rain      PLAN_HV NSS

[[2]][[12]]
+ 2/11 vertices, named, from f0b35d9:
[1] rain NSS

nestedness_paths <-
lapply(all_paths_nestednes, FUN =
  function(z) {
    d <-
      lapply(z, FUN =
        function(x) {
          x <- as.numeric(x)

edges <- c('Temperature', 'Rainfall', 'Fr. Production', 'Fr. S.',
          'Nutr. ifs', 'Morph. ifs', 'Birds. ifs',
          'Fr. Tot. ifs', "H2'", 'Modularity', 'Nestedness')

```

```

x <- edges[x]
v <- matrix(NA, nrow = 1, ncol = 5)
for (i in seq_along(x)) {
  v[1, i] <- x[i]
}

})
d <- as_tibble(do.call('rbind', d))
colnames(d) <- paste0('Stage ', 1:5)
d
})

```

Warning: The `x` argument of `as\_tibble.matrix()` must have unique column names if `.name\_repair` is omitted as of tibble 2.0.0.  
i Using compatibility `.name\_repair`.

```

nestedness_paths <- do.call('rbind', nestedness_paths)

#nestedness_paths <-
nestedness_paths |>
  make_long(`Stage 1`, `Stage 2`, `Stage 3`, `Stage 4`, `Stage 5`) |>
  ggplot(aes(
    x = x,
    next_x = next_x,
    node = node,
    next_node = next_node,
    fill = node,
    label = node
  )) +
  geom_sankey(flow.alpha = 0.6, node.color = "gray30") +
  geom_sankey_label(size = 3, color = "black", fill = "white") +
  theme_sankey(base_size = 14) +
  #ggtitle("Nestedness") +
  #scale_fill_brewer(palette = "Set2") +
  theme(axis.title = element_blank(),
        legend.position = 'none')

```

Warning: Removed 3 rows containing missing values or values outside the scale range (`geom\_label()`).

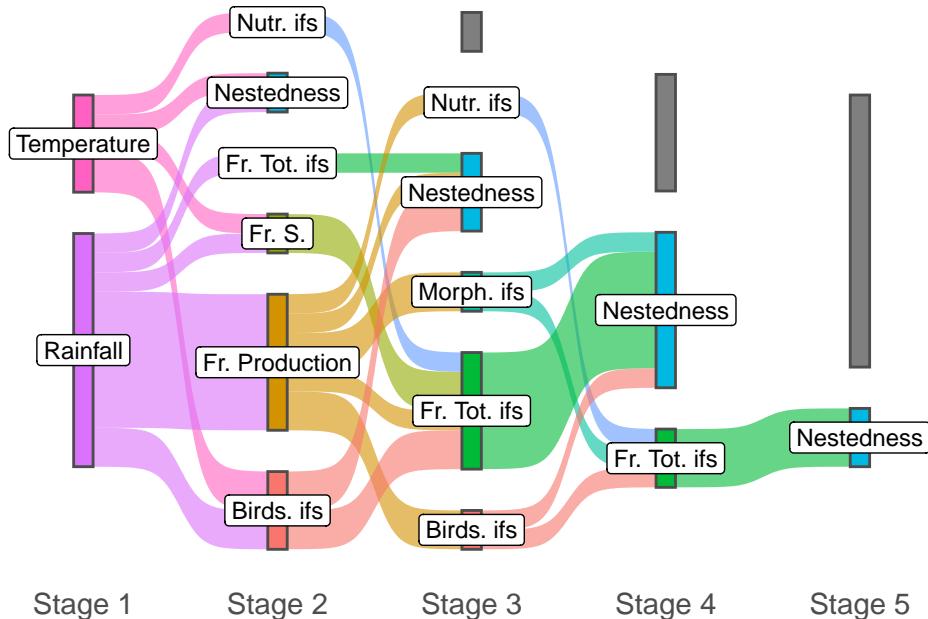


Figure S15: Sankey plot showing direct and climatic-driven bottom-up effects explaining temporal dynamics of nestedness in seed dispersal networks

#### S4.8.2 Modularity

Direct effects

```
sp <- lapply(1:2, FUN =
  function(x) {
    shortest_paths(net_plot, from = x, to = 10,
      weights = E(net_plot)$weight)
  })

(shortest_path_MODU <- lapply(1:2, FUN =
  function(x) {
    sp[[x]]$vpath[[1]]
  }))
```

```
[[1]]
+ 3/11 vertices, named, from f0b35d9:
[1] temp   nut_HV MODU

[[2]]
+ 2/11 vertices, named, from f0b35d9:
[1] rain MODU
```

## Bottom-up effects

```
all_paths_MODU <- lapply(1:2, FUN =  
  function(x) {  
    all_simple_paths(net_plot, from = x, to = 10)  
  })
```

```
all_paths_MODU
```

```
[[1]]  
[[1]][[1]]  
+ 4/11 vertices, named, from f0b35d9:  
[1] temp      fruits  PLAN_HV MODU
```

```
[[1]][[2]]  
+ 4/11 vertices, named, from f0b35d9:  
[1] temp      nut_HV  PLAN_HV MODU
```

```
[[1]][[3]]  
+ 3/11 vertices, named, from f0b35d9:  
[1] temp      nut_HV MODU
```

```
[[1]][[4]]  
+ 4/11 vertices, named, from f0b35d9:  
[1] temp      BIRD_HV PLAN_HV MODU
```

```
[[2]]  
[[2]][[1]]  
+ 5/11 vertices, named, from f0b35d9:  
[1] rain      fruitAB nut_HV  PLAN_HV MODU
```

```
[[2]][[2]]  
+ 4/11 vertices, named, from f0b35d9:  
[1] rain      fruitAB nut_HV  MODU
```

```
[[2]][[3]]  
+ 5/11 vertices, named, from f0b35d9:  
[1] rain      fruitAB mor_HV  PLAN_HV MODU
```

```
[[2]][[4]]  
+ 5/11 vertices, named, from f0b35d9:  
[1] rain      fruitAB BIRD_HV PLAN_HV MODU
```

```

[[2]][[5]]
+ 4/11 vertices, named, from f0b35d9:
[1] rain      fruitAB PLAN_HV MODU

[[2]][[6]]
+ 4/11 vertices, named, from f0b35d9:
[1] rain      fruitS   PLAN_HV MODU

[[2]][[7]]
+ 4/11 vertices, named, from f0b35d9:
[1] rain      BIRD_HV PLAN_HV MODU

[[2]][[8]]
+ 3/11 vertices, named, from f0b35d9:
[1] rain      PLAN_HV MODU

[[2]][[9]]
+ 2/11 vertices, named, from f0b35d9:
[1] rain MODU

MODU_paths <-
  lapply(all_paths_MODU, FUN =
    function(z) {
      d <-
        lapply(z, FUN =
          function(x) {
            x <- as.numeric(x)

            edges <- c('Temperature', 'Rainfall', 'Fr. Production', 'Fr. S.',
                      'Nutr. ifs', 'Morph. ifs', 'Birds. ifs',
                      'Fr. Tot. ifs', "H2'", 'Modularity', 'Nestedness')

            x <- edges[x]

            v <- matrix(NA, nrow = 1, ncol = 5)

            for (i in seq_along(x)) {
              v[1, i] <- x[i]
            }

            v
          })
      d <- as_tibble(do.call('rbind', d))
      colnames(d) <- paste0('Stage ', 1:5)
      d
    })
}

MODU_paths <- do.call('rbind', MODU_paths)

#MODU_paths <-
#  MODU_paths |>
#  make_long(`Stage 1`, `Stage 2`, `Stage 3`, `Stage 4`, `Stage 5`) |>

```

```

    ggpplot(aes(
      x = x,
      next_x = next_x,
      node = node,
      next_node = next_node,
      fill = node,
      label = node
    )) +
      geom_sankey(flow.alpha = 0.6, node.color = "gray30") +
      geom_sankey_label(size = 3, color = "black", fill = "white") +
      theme_sankey(base_size = 14) +
      #ggtitle("Modularity") +
      #scale_fill_brewer(palette = "Set2") +
      theme(axis.title = element_blank(),
            legend.position = 'none')

```

Warning: Removed 3 rows containing missing values or values outside the scale range  
`geom\_label()`).

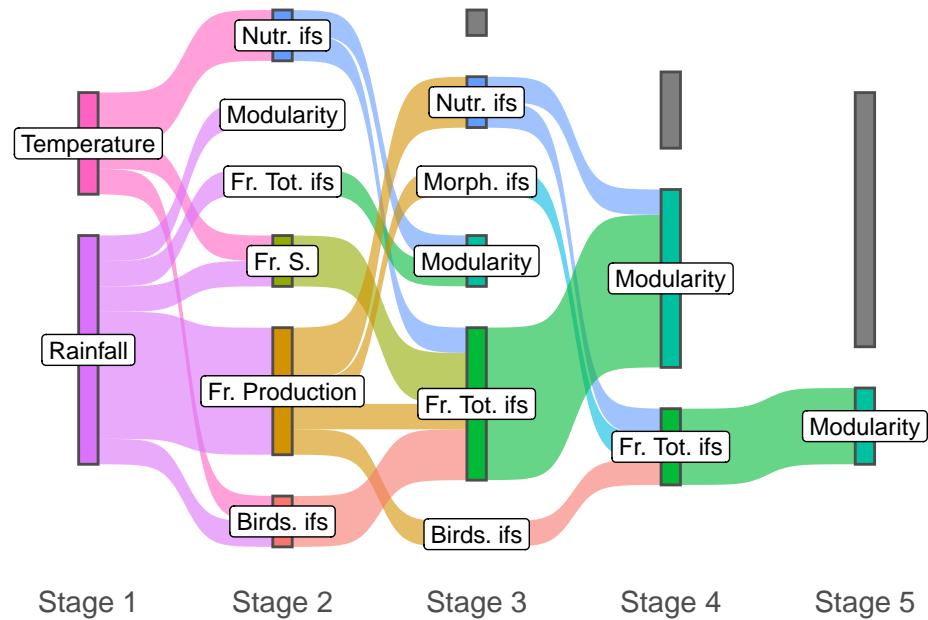


Figure S16: Sankey plot showing direct and climatic-driven bottom-up effects explaining temporal dynamics of modularity in seed dispersal networks

#### S4.8.3 $H'_2$

Direct effects

```

sp <- lapply(1:2, FUN =
             function(x) {
               shortest_paths(net_plot, from = x, to = 9,
                             weights = E(net_plot)$weight)
             })

(shortest_path_H2 <- lapply(1:2, FUN =
                            function(x) {
                              sp[[x]]$vpath[[1]]
                            }))

```

```

[[1]]
+ 2/11 vertices, named, from f0b35d9:
[1] temp H2

[[2]]
+ 2/11 vertices, named, from f0b35d9:
[1] rain H2

```

```
#
```

Bottom-up effects

```

all_paths_H2 <- lapply(1:2, FUN =
                        function(x) {
                          all_simple_paths(net_plot, from = x, to = 9)
                        })

all_paths_H2

```

```

[[1]]
[[1]][[1]]
+ 2/11 vertices, named, from f0b35d9:
[1] temp H2

```

```

[[2]]
[[2]][[1]]
+ 2/11 vertices, named, from f0b35d9:
[1] rain H2

```

```

H2_paths <-
  lapply(all_paths_H2, FUN =
        function(z) {
          d <-
            lapply(z, FUN =
                  function(x) {

```

```

x <- as.numeric(x)

edges <- c('Temperature', 'Rainfall', 'Fr. Production', 'Fr. S.',
          'Nutr. ifs', 'Morph. ifs', 'Birds. ifs',
          'Fr. Tot. ifs', "H2'", 'Modularity', 'Nestedness')

x <- edges[x]

v <- matrix(NA, nrow = 1, ncol = 5)

for (i in seq_along(x)) {
  v[1, i] <- x[i]
}

v
})

d <- as_tibble(do.call('rbind', d))
colnames(d) <- paste0('Stage ', 1:5)
d
})

H2_paths <- do.call('rbind', H2_paths)

total_paths <- rbind(nestedness_paths, MODU_paths, H2_paths)

total_paths <-
  total_paths |>
  make_long(`Stage 1`, `Stage 2`, `Stage 3`, `Stage 4`, `Stage 5`)

# total_paths$alpha <- 0
# total_paths$alpha[which(sum(is.na(total_paths$node)) > 0)] <- 0.2

H2_paths <-
  H2_paths |>
  make_long(`Stage 1`, `Stage 2`)

#total_paths_plot <-
#  ggplot(H2_paths, aes(
#    x = x,
#    next_x = next_x,
#    node = node,
#    next_node = next_node,
#    fill = node,
#    label = node
#  )) +
#  geom_sankey(flow.alpha = 0.6, node.color = "gray30") +
#  geom_sankey_label(size = 3, color = "black", fill = "white",) +
#  theme_sankey(base_size = 18) +
#  ggtitle("H2") +
#  scale_fill_brewer(palette = "Set2") +
#  theme(axis.title = element_blank(),
#        legend.position = 'none')

```

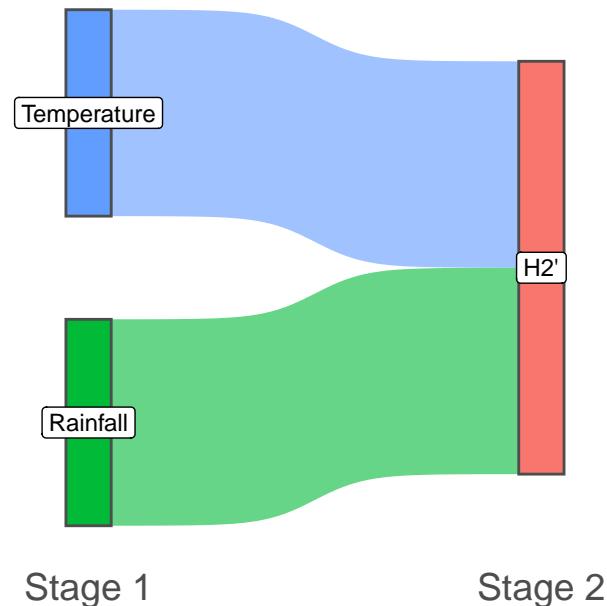


Figure S17: Sankey plot showing direct and climatic-driven bottom-up effects explaining temporal dynamics of specialization in seed dispersal networks

```
ggplot(total_paths, aes(
  x = x,
  next_x = next_x,
  node = node,
  next_node = next_node,
  fill = node,
  label = node
)) +
  geom_sankey(flow.alpha = 0.6, node.color = "gray30") +
  geom_sankey_label(size = 3, color = "black", fill = "white",) +
  theme_sankey(base_size = 18) +
  scale_fill_manual(values = RColorBrewer::brewer.pal(11, "Spectral")) +
  #ggtitle("H2") +
  #scale_fill_brewer(palette = "Set2") +
  theme(axis.title = element_blank(),
        legend.position = 'none')
```

Warning: Removed 3 rows containing missing values or values outside the scale range  
(`geom\_label()`).

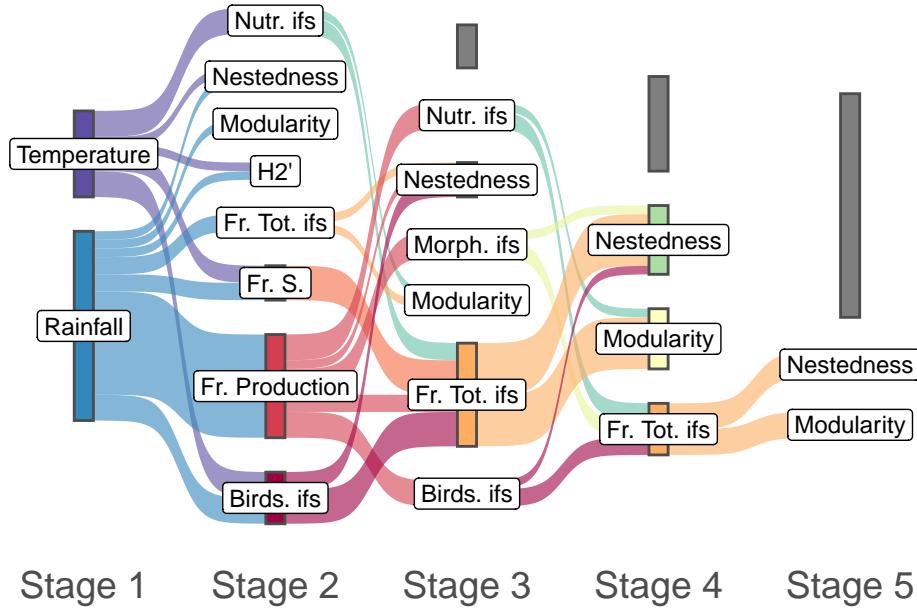


Figure S18: Sankey plot showing direct and climatic-driven bottom-up effects explaining temporal dynamics seed dispersal networks (all metrics)

#### S4.9 Causal effects

We used the posterior distributions of the generative model to simulate two types of scenarios:

- $Y_{\text{scenario 1}} = X_{\min} \rightarrow Y$
- $Y_{\text{scenario 2}} = X_{\max} \rightarrow Y$

That is, the causal effect of  $X$  on  $Y$  result from the contrast of both scenarios (i.e., causal effect  $X \rightarrow Y = |Y_{\text{scenario 2}} - Y_{\text{scenario 1}}|$ ). The function `causal_function()` was programmed to conduct the simulations.

```
causal_effects <-
  function(pars = post_H2_G, # object with posterior distributions of the BN model
          alpha_1, f_1, # site and GP parameters for equation 1
          alpha_2, f_2, # site and GP parameters for equation 2
          alpha_3, f_3, # site and GP parameters for equation 3
          alpha_4, f_4, # site and GP parameters for equation 4
          beta_1, beta_2, beta_3, beta_4, # effects to assess at equation i
          var, # exposure or predictor variable to assess
          y_obs, # outcome variable in its natural scale
          level_1 = T, # direct effect (x -> y)
          level_2 = F, # indirect effect (x -> z -> y)
          level_3 = F, # indirect effect (x -> z -> k -> y)
```

```

    level_4 = F # indirect effect (x -> z -> k -> l -> y)
) {

if (level_1) {

# minimum and maximum values of predictor variable
x_1 <- quantile(var, c(0, 1))

# indexing site and GP parameters EQ 1
a <- grep(alpha_1, colnames(pars$alpha))
f_1 <- grep(f_1, colnames(pars$f))

effect <- # simulating the intervention
sapply(x_1, FUN =
       function(x) {

         with(pars, {
           # EQ 1
           apply(alpha[, a], 1, mean) +
             apply(f[, f_1], 1, mean) +
               beta[[beta_1]] * x
         })
       })

colnames(effect) <- c('min', 'max')

# going from z-scores to the scale of the variable
if ((grepl('MODU', beta_1) +
     grepl('H2', beta_1)) == 0) {
  effect <- apply(effect, 2, function(z) mean(y_obs) + z * sd(y_obs))
} else {
  effect <- apply(effect, 2, function(z) inv_logit(z))
}

effect <- as_tibble(effect)
#contrast
effect$contrast <- effect %$% (max - min)
# p(effect)
effect$p_effect <- mean(effect$contrast > 0)
# % change
effect$relative_change <- ((effect$contrast * 100) / mean(y_obs))
effect$effect <- paste0(beta_1)
})

if (level_2) {

# minimum and maximum values of predictor variable
x_1 <- quantile(var, c(0, 1))

# indexing site and GP parameters EQ 1
a_1 <- grep(alpha_1, colnames(pars$alpha))
f_1 <- grep(f_1, colnames(pars$f))
# indexing site and GP parameters EQ 2
a_2 <- grep(alpha_2, colnames(pars$alpha))
f_2 <- grep(f_2, colnames(pars$f))

effect <- # simulating the intervention
sapply(x_1, FUN =
       function(x) {
         with(pars, {
           # EQ 1
           effect1 <-
             apply(alpha[, a_1], 1, mean) +
               apply(f[, f_1], 1, mean) +
                 beta[[beta_1]] * x
           # EQ 2
           effect2 <-
             apply(alpha[, a_2], 1, mean) +
               apply(f[, f_2], 1, mean) +
                 beta[[beta_2]] * x
         })
       })

colnames(effect) <- c('min', 'max')

# going from z-scores to the scale of the variable
if ((grepl('MODU', beta_1) +
     grepl('H2', beta_1)) == 0) {
  effect <- apply(effect, 2, function(z) mean(y_obs) + z * sd(y_obs))
} else {
  effect <- apply(effect, 2, function(z) inv_logit(z))
}

effect <- as_tibble(effect)
#contrast
effect$contrast <- effect %$% (max - min)
# p(effect)
effect$p_effect <- mean(effect$contrast > 0)
# % change
effect$relative_change <- ((effect$contrast * 100) / mean(y_obs))
effect$effect <- paste0(beta_1)
})
}
}

```

```

apply(alpha[, a_2], 1, mean) +
apply(f[, f_2], 1, mean) +
beta[[beta_2]] * effect1

effect2
})
})

colnames(effect) <- c('min', 'max')

# going from z-scores to the scale of the variable
if ((grepl('MODU', beta_2) +
grepl('H2', beta_2)) == 0) {
  effect <- apply(effect, 2, function(z) mean(y_obs) + z * sd(y_obs))
} else {
  effect <- apply(effect, 2, function(z) inv_logit(z))
}
effect <- as_tibble(effect)
#contrast
effect$contrast <- effect %$% (max - min)
# p(effect)
effect$p_effect <- mean(effect$contrast > 0)
# % change
effect$relative_change <- ((effect$contrast * 100) / mean(y_obs))
effect$effect <- paste0(beta_1, '->', beta_2)
}

if (level_3) {
  # minimum and maximum values of predictor variable
  x_1 <- quantile(var, c(0, 1))

  # indexing site and GP parameters EQ 1
  a_1 <- grep(alpha_1, colnames(pars$alpha))
  f_1 <- grep(f_1, colnames(pars$f))
  # indexing site and GP parameters EQ 2
  a_2 <- grep(alpha_2, colnames(pars$alpha))
  f_2 <- grep(f_2, colnames(pars$f))
  # indexing site and GP parameters EQ 3
  a_3 <- grep(alpha_3, colnames(pars$alpha))
  f_3 <- grep(f_3, colnames(pars$f))

  effect <- # simulating the intervention
  sapply(x_1, FUN =
    function(x) {
      with(pars, {
        # EQ 1
        effect1 <-
          apply(alpha[, a_1], 1, mean) +
          apply(f[, f_1], 1, mean) +
          beta[[beta_1]] * x

        # EQ 2
        effect2 <-
          apply(alpha[, a_2], 1, mean) +
          apply(f[, f_2], 1, mean) +
          beta[[beta_2]] * effect1

        # EQ 3
        effect3 <-
          apply(alpha[, a_3], 1, mean) +
          apply(f[, f_3], 1, mean) +
          beta[[beta_3]] * effect2

        effect3
      })
    })
}

```

```

colnames(effect) <- c('min', 'max')

# going from z-scores to the scale of the variable
if ((grepl('MODU', beta_3) +
     grepl('H2', beta_3)) == 0) {
  effect <- apply(effect, 2, function(z) mean(y_obs) + z * sd(y_obs))
} else {
  effect <- apply(effect, 2, function(z) inv_logit(z))
}
effect <- as_tibble(effect)
#contrast
effect$contrast <- effect %$% (max - min)
# p(effect)
effect$p_effect <- mean(effect$contrast > 0)
# % change
effect$relative_change <- ((effect$contrast * 100) / mean(y_obs))
effect$effect <- paste0(beta_1, '>', beta_2, '>', beta_3)

}

if (level_4) {
  # minimum and maximum values of predictor variable
  x_1 <- quantile(var, c(0, 1))

  # indexing site and GP parameters EQ 1
  a_1 <- grep(alpha_1, colnames(pars$alpha))
  f_1 <- grep(f_1, colnames(pars$f))
  # indexing site and GP parameters EQ 2
  a_2 <- grep(alpha_2, colnames(pars$alpha))
  f_2 <- grep(f_2, colnames(pars$f))
  # indexing site and GP parameters EQ 3
  a_3 <- grep(alpha_3, colnames(pars$alpha))
  f_3 <- grep(f_3, colnames(pars$f))
  # indexing site and GP parameters EQ 4
  a_4 <- grep(alpha_4, colnames(pars$alpha))
  f_4 <- grep(f_4, colnames(pars$f))

  effect <- # simulating the intervention
  sapply(x_1, FUN =
    function(x) {
      with(pars, {
        # EQ 1
        effect1 <-
          apply(alpha[, a_1], 1, mean) +
          apply(f[, f_1], 1, mean) +
          beta[[beta_1]] * x

        # EQ 2
        effect2 <-
          apply(alpha[, a_2], 1, mean) +
          apply(f[, f_2], 1, mean) +
          beta[[beta_2]] * effect1

        # EQ 3
        effect3 <-
          apply(alpha[, a_3], 1, mean) +
          apply(f[, f_3], 1, mean) +
          beta[[beta_3]] * effect2

        # EQ 4
        effect4 <-
          apply(alpha[, a_4], 1, mean) +
          apply(f[, f_4], 1, mean) +
          beta[[beta_4]] * effect2
      })
      effect4
    })
}

```

```

        })

colnames(effect) <- c('min', 'max')

# going from z-scores to the scale of the variable
if ((grepl('MODU', beta_4) +
     grepl('H2', beta_4)) == 0) {
  effect <- apply(effect, 2, function(z) mean(y_obs) + z * sd(y_obs))
} else {
  effect <- apply(effect, 2, function(z) inv_logit(z))
}
effect <- as_tibble(effect)
#contrast
effect$contrast <- effect %$% (max - min)
# p(effect)
effect$p_effect <- mean(effect$contrast > 0)
# % change
effect$relative_change <- ((effect$contrast * 100) / mean(y_obs))
effect$effect <- paste0(beta_1, '>', beta_2, '>', beta_3, '>', beta_4)
}

effect
}

```

#### S4.9.1 Functional space

#### S4.9.2 $X \rightarrow$ Birds' functional space

```

knitr::kable(betas_H2_G_z[grep('BIRD_HV$', betas_H2_G_z[[1]]), ],
             digits = 2,
             caption = "Factors affecting birds' functional space. Average values, 95% CIs, and probabilities of

```

Table S4: Factors affecting birds' functional space. Average values, 95% CIs, and probabilities of directional effects for the relationships assessed in the generative model. An effect is marked as TRUE if  $P(\text{effect} > 0) > 0.7$  or  $P(\text{effect} < 0) > 0.7$  or.

Directional effect	Mean	0.025%	0.975%	$P(\text{effect} > 0)$	$P(\text{effect} < 0)$	effect
temp -> BIRD_HV	0.19	-0.13	0.50	0.89	0.11	TRUE
rain -> BIRD_HV	-0.13	-0.48	0.19	0.22	0.78	TRUE
fruitAB -> BIRD_HV	0.24	-0.12	0.59	0.91	0.09	TRUE
fruitS -> BIRD_HV	-0.05	-0.37	0.27	0.39	0.61	FALSE

Relative importance of the most important exposure variable affecting birds' functional space.

```
# fruit production - temperature
((betas_H2_G[grep('BIRD_HV$', betas_H2_G$var), ]$mu[3] -
  abs(betas_H2_G[grep('BIRD_HV$', betas_H2_G$var), ]$mu[1])) / 
  abs(betas_H2_G[grep('BIRD_HV$', betas_H2_G$var), ]$mu[1])) * 100
```

[1] 25.69718

```
# fruit production - rainfall
((abs(betas_H2_G[grep('BIRD_HV$', betas_H2_G$var), ]$mu[3])) -
  abs(betas_H2_G[grep('BIRD_HV$', betas_H2_G$var), ]$mu[2])) / 
  abs(betas_H2_G[grep('BIRD_HV$', betas_H2_G$var), ]$mu[2])) * 100
```

[1] 78.80812

#### S4.9.2.1 Fruit production → Birds' functional space

```
effect_fruitAB_bird <-
  causal_effects(
    pars = post_H2_G,
    alpha_1 = 'alpha_BIRD_HV',
    f_1 = 'f_BIRD_HV',
    beta_1 = 'beta_fruitAB_BIRD_HV',
    var = dat$fruit_abun,
    y_obs = d$HV_bird,
    level_1 = T
  )
```

Percentage increase of intervening *fruit abundance*: average value and 95% CI.

```
quantile(effect_fruitAB_bird$relative_change, c(0.025, 0.5, 0.975))
```

2.5%	50%	97.5%
-5.923772	12.000343	29.611029

Probability of the causal effect being positive:

```
unique(effect_fruitAB_bird$p_effect)
```

[1] 0.9115442

#### S4.9.2.2 Rainfall → Birds' functional space

```

effect_rain_bird <-
causal_effects(
  pars = post_H2_G,
  alpha_1 = 'alpha_BIRD_HV',
  f_1 = 'f_BIRD_HV',
  beta_1 = 'beta_rain_BIRD_HV',
  var = dat$z_rainfall,
  y_obs = d$HV_bird,
  level_1 = T
)

```

Percentage increase of intervening *rainfall*: average value and 95% CI.

```
quantile(effect_rain_bird$relative_change, c(0.025, 0.5, 0.975))
```

2.5%	50%	97.5%
-26.726020	-7.493846	10.870469

Probability of the causal effect being positive:

```
unique(effect_rain_bird$p_effect)
```

[1] 0.2163918

#### S4.9.2.3 Temperature → Birds' functional space

```

effect_temp_bird <-
causal_effects(
  pars = post_H2_G,
  alpha_1 = 'alpha_BIRD_HV',
  f_1 = 'f_BIRD_HV',
  beta_1 = 'beta_temp_BIRD_HV',
  var = dat$z_rainfall,
  y_obs = d$HV_bird,
  level_1 = T
)

```

Percentage increase of intervening *temperature*: average value and 95% CI.

```
quantile(effect_temp_bird$relative_change, c(0.025, 0.5, 0.975))
```

2.5%	50%	97.5%
-7.32774	11.08906	28.11336

Probability of the causal effect being positive:

```
unique(effect_temp_bird$p_effect)
```

```
[1] 0.8890555
```

#### S4.9.2.4 Rainfall → fruitproduction → Birds' functional space

```
effect_rain_fruitAB_bird <-  
  causal_effects(  
    pars = post_H2_G,  
    alpha_1 = 'alpha_fruitAB',  
    f_1 = 'f_fruitAB',  
    beta_1 = 'beta_rain_fruitAB',  
    alpha_2 = 'alpha_BIRD_HV',  
    f_2 = 'f_BIRD_HV',  
    beta_2 = 'beta_fruitAB_BIRD_HV',  
    var = dat$z_rainfall,  
    y_obs = d$HV_bird,  
    level_1 = F,  
    level_2 = T  
)
```

Percentage increase of intervening *rainfal* (*mediation path*): average value and 95% CI.

```
quantile(effect_rain_fruitAB_bird$relative_change, c(0.025, 0.5, 0.975))
```

```
2.5%           50%         97.5%  
-6.1315783 -0.8743122  1.7864030
```

Probability of the causal effect being positive:

```
unique(effect_rain_fruitAB_bird$p_effect)
```

```
[1] 0.2343828
```

#### S4.9.2.5 Plot

```
df_causal_BIRD <-  
  rbind(effect_fruitAB_bird,  
        effect_temp_bird,  
        effect_rain_fruitAB_bird) |>  
  group_by(effect) |>  
  transmute(mu = median(relative_change),  
            li = quantile(relative_change, 0.025),  
            ls = quantile(relative_change, 0.975)) |>  
  unique() |> ungroup() |>  
  mutate(x =  
    c('Fruit production',  
      'Temperature',
```

```

      '*Rainfall      \n (fruit production)'),
y = "Birds' hypervolume")

causal_plot_BIRD_HV <-
  rbind(effect_fruitAB_bird,
    effect_temp_bird,
    effect_rain_fruitAB_bird) |>
  group_by(effect) |>
  transmute(mu = median(relative_change),
            li = quantile(relative_change, 0.025),
            ls = quantile(relative_change, 0.975)) |>
  unique() |> ungroup() |>
  mutate(x =
    c('Fruit production',
      'Temperature',
      '*Rainfall      \n (fruit production)'),
    y = "Birds' Hypervolume") |>
  ggplot() +
  geom_hline(yintercept = 0, linetype = 3, linewidth = 0.25) +
  geom_errorbar(aes(x, ymin = li, ymax = ls),
                width = 0, linewidth = 1.25, alpha = 0.5,
                color = '#1e81b0') +
  geom_point(aes(x, mu), color = '#e28743', size = 2) +
  labs(y = 'Relative causal effect (%)', x = '') +
  facet_wrap(~y) +
  theme_classic() +
  theme(panel.grid = element_blank(),
        strip.background = element_blank(),
        axis.line = element_line(linewidth = 0.25),
        axis.ticks = element_line(linewidth = 0.25),
        axis.title.x = element_blank(),
        #text = element_text(family = 'Times New Roman'),
        axis.text.x = element_text(angle = 45, hjust = 1, vjust = 1),
        axis.text = element_text(size = 8.5))

```

```
causal_plot_BIRD_HV
```

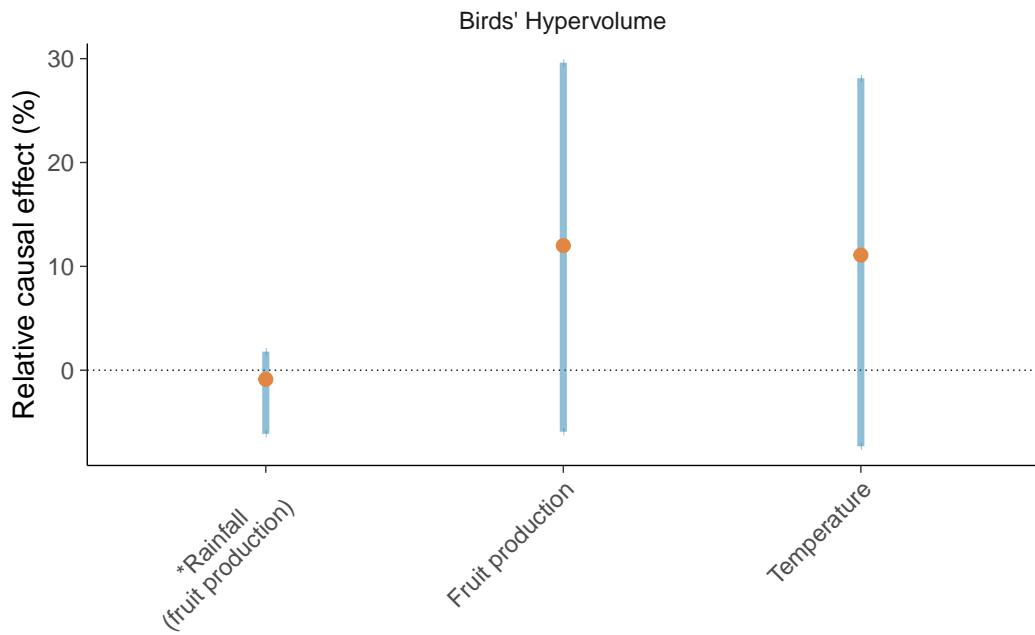


Figure S19: Causal effects of exposure variables on birds' functional space. An asterisk indicates effects mediated through the variable in parentheses

#### S4.9.3 $X \rightarrow$ fruits' nutritional functional space

```
knitr::kable(betas_H2_G_z[grep('nut_HV$', betas_H2_G_z[[1]]), ],
             digits = 2,
             caption = "Factors affecting fruits' nutritional functional space. Average values, 95% CIs, and probabilities of directional effects for the relationships assessed in the generative model. An effect is marked as TRUE if P(effect > 0) > 0.7 or P(effect < 0) > 0.7 or.")
```

Table S5: Factors affecting fruits' nutritional functional space. Average values, 95% CIs, and probabilities of directional effects for the relationships assessed in the generative model. An effect is marked as TRUE if  $P(\text{effect} > 0) > 0.7$  or  $P(\text{effect} < 0) > 0.7$  or.

Directional effect	Mean	0.025%	0.975%	$P(\text{effect} > 0)$	$P(\text{effect} < 0)$	effect
temp -> nut_HV	0.36	0.06	0.66	0.99	0.01	TRUE
rain -> nut_HV	-0.03	-0.32	0.29	0.45	0.55	FALSE
fruitAB -> nut_HV	0.22	-0.15	0.61	0.87	0.13	TRUE
fruitS -> nut_HV	-0.03	-0.35	0.27	0.42	0.58	FALSE
BIRD -> nut_HV	0.02	-0.17	0.20	0.61	0.39	FALSE

Relative importance of temperature explaining fruits' nutritional functional space.

```
# temperature - fruitAB
((abs(betas_H2_G[grep('nut_HV$', betas_H2_G$var), ]$mu[1]) -
  abs(betas_H2_G[grep('nut_HV$', betas_H2_G$var), ]$mu[3])) /
  abs(betas_H2_G[grep('nut_HV$', betas_H2_G$var), ]$mu[3])) * 100
```

```
[1] 65.59931
```

#### S4.9.3.1 Temperature → fruits' nutritional functional space

```
effect_temp_nut <-
  causal_effects(
    pars = post_H2_G,
    alpha_1 = 'alpha_nut_HV',
    f_1 = 'f_nut_HV',
    beta_1 = 'beta_temp_nut_HV',
    var = dat$z_temperature,
    y_obs = d$HV_plants_nut,
    level_1 = T
  )
```

Percentage increase of intervening *temperature*: average value and 95% CI.

```
quantile(effect_temp_nut$relative_change,
         c(0.025, 0.5, 0.975))
```

```
2.5%      50%      97.5%
11.95830  70.27279 129.98457
```

Probability of the causal effect being positive:

```
unique(effect_temp_nut$p_effect)
```

```
[1] 0.990005
```

#### S4.9.3.2 Fruit production → fruits' nutritional functional space

```
effect_fruitAB_nut <-
  causal_effects(
    pars = post_H2_G,
    alpha_1 = 'alpha_nut_HV',
    f_1 = 'f_nut_HV',
    beta_1 = 'beta_fruitAB_nut_HV',
    var = dat$fruit_abun,
    y_obs = d$HV_plants_nut,
    level_1 = T
  )
```

Percentage increase of intervening *fruit production*: average value and 95% CI.

```
quantile(effect_fruitAB_nut$relative_change,
         c(0.025, 0.5, 0.975))
```

```
2.5%      50%     97.5%
-27.38643 40.30687 111.01082
```

Probability of the causal effect being positive:

```
unique(effect_fruitAB_nut$p_effect)
```

```
[1] 0.870065
```

#### S4.9.3.3 Plot

```
df_causal_nut <-
  rbind(effect_temp_nut,
        effect_fruitAB_nut) |>
  group_by(effect) |>
  transmute(mu = median(relative_change),
            li = quantile(relative_change, 0.025),
            ls = quantile(relative_change, 0.975)) |>
  unique() |> ungroup() |>
  mutate(x =
    c('Temperature',
      'Fruit production'),
    y = "Fruits' nutr. hypervolume")

causal_plot_nut_HV <-
  rbind(effect_temp_nut,
        effect_fruitAB_nut) |>
  group_by(effect) |>
  transmute(mu = median(relative_change),
            li = quantile(relative_change, 0.025),
            ls = quantile(relative_change, 0.975)) |>
  unique() |> ungroup() |>
  mutate(x =
    c('Temperature',
      'Fruit production'),
    y = "Fruits' nutr. Hypervolume") |>
  ggplot() +
  geom_hline(yintercept = 0, linetype = 3, linewidth = 0.25) +
  geom_errorbar(aes(x, ymin = li, ymax = ls),
                width = 0, linewidth = 1.25, alpha = 0.5,
                color = '#1e81b0') +
  geom_point(aes(x, mu), color = '#e28743', size = 2) +
  labs(y = 'Relative causal effect (%)', x = '') +
  facet_wrap(~y) +
  theme_classic() +
  theme(panel.grid = element_blank(),
        strip.background = element_blank(),
        axis.line = element_line(linewidth = 0.25),
        axis.ticks = element_line(linewidth = 0.25),
        #text = element_text(family = 'Times New Roman'),
        axis.text.x = element_text(angle = 45, hjust = 1, vjust = 1),
        axis.text = element_text(size = 8.5))
```

`causal_plot_nut_HV`

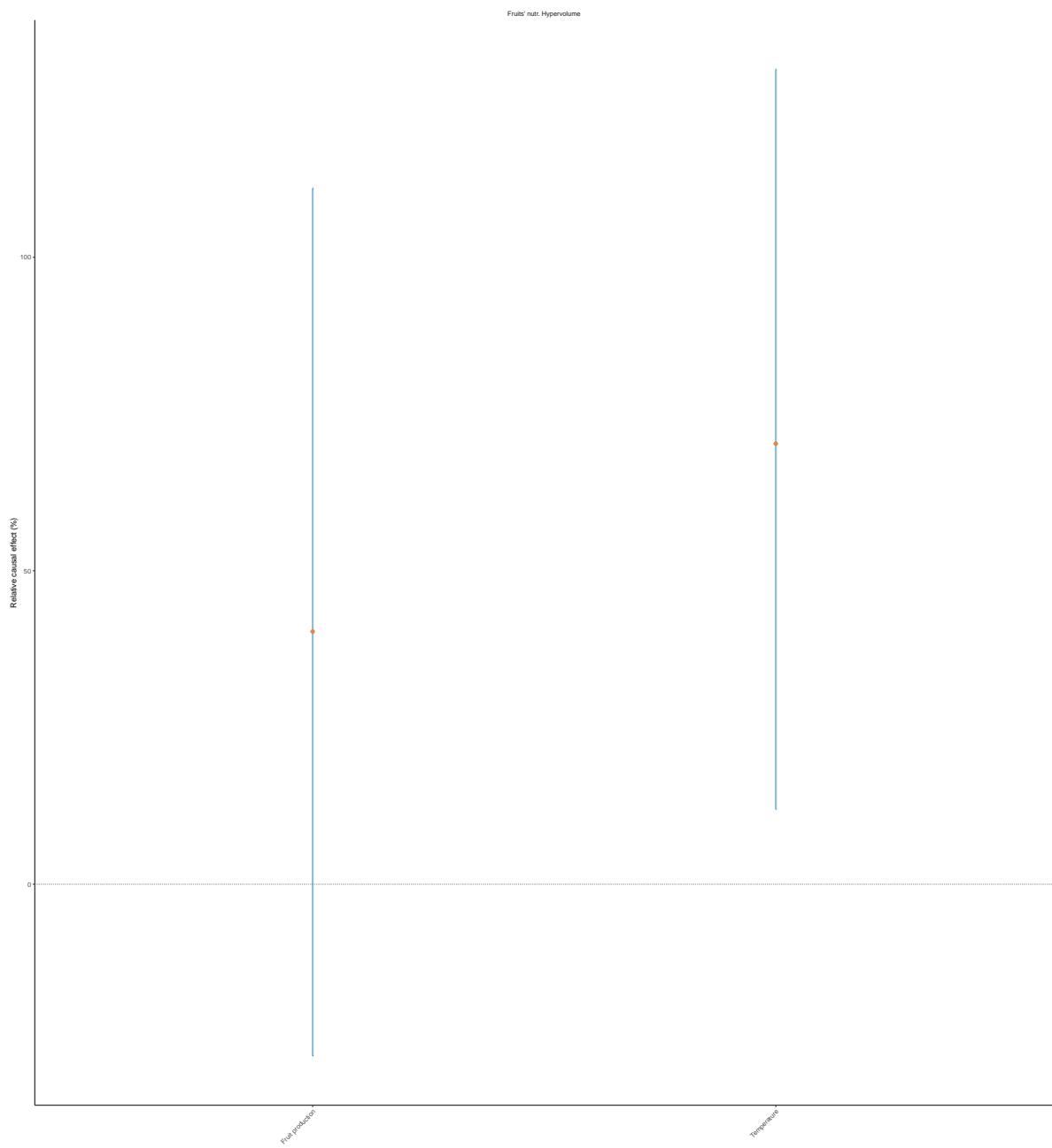


Figure S20: Causal effects of exposure variables on fruits' nutritional functional space. An asterisk indicates effects mediated through the variable in parentheses

#### S4.9.4 $X \rightarrow$ fruits' morphological functional space

```
knitr::kable(betas_H2_G_z[grep('mor_HV$', betas_H2_G_z[[1]]), ],
             digits = 2,
             caption = "Factors affecting fruits' nutritional functional space. Average values, 95% CIs, and prob")
```

Table S6: Factors affecting fruits' nutritional functional space. Average values, 95% CIs, and probabilities of directional effects for the relationships assessed in the generative model. An effect is marked as TRUE if  $P(\text{effect} > 0) > 0.7$  or  $P(\text{effect} < 0) > 0.7$  or.

Directional effect	Mean	0.025%	0.975%	$P(\text{effect} > 0)$	$P(\text{effect} < 0)$	effect
temp -> mor_HV	0.03	-0.25	0.32	0.58	0.42	FALSE
rain -> mor_HV	0.06	-0.26	0.38	0.65	0.35	FALSE
fruitAB -> mor_HV	0.21	-0.15	0.60	0.87	0.13	TRUE
fruitS -> mor_HV	-0.03	-0.33	0.29	0.45	0.55	FALSE
BIRD -> mor_HV	0.04	-0.21	0.29	0.63	0.37	FALSE

##### S4.9.4.1 Fruit production $\rightarrow$ fruits' morphological functional space

```
effect_fruitAB_mor <-  
  causal_effects(  
    pars = post_H2_G,  
    alpha_1 = 'alpha_mor_HV',  
    f_1 = 'f_mor_HV',  
    beta_1 = 'beta_fruitAB_mor_HV',  
    var = dat$fruit_abun,  
    y_obs = d$HV_plants_morfo,  
    level_1 = T  
)
```

Percentage increase of intervening *fruit production*: average value and 95% CI.

```
quantile(effect_fruitAB_mor$relative_change,  
        c(0.025, 0.5, 0.975))
```

```
2.5%      50%      97.5%  
-26.34763 36.18145 102.29290
```

Probability of the causal effect being positive:

```
unique(effect_fruitAB_mor$p_effect)
```

```
[1] 0.8650675
```

#### S4.9.4.2 Rainfall → Fruit production → fruits' morphological functional space

```
effect_rain_fruitAB_mor <-
  causal_effects(
    pars = post_H2_G,
    alpha_1 = 'alpha_fruitAB',
    f_1 = 'f_fruitAB',
    beta_1 = 'beta_rain_fruitAB',
    alpha_2 = 'alpha_mor_HV',
    f_2 = 'f_mor_HV',
    beta_2 = 'beta_fruitAB_mor_HV',
    var = dat$z_rainfall,
    y_obs = d$HV_plants_morfo,
    level_1 = F,
    level_2 = T
  )
```

Percentage increase of intervening *rainfall* (*mediation path*): average value and 95% CI.

```
quantile(effect_rain_fruitAB_mor$relative_change,
         c(0.025, 0.5, 0.975))
```

2.5%	50%	97.5%
-20.136423	-2.357868	6.901229

Probability of the causal effect being positive:

```
unique(effect_rain_fruitAB_mor$p_effect)
```

```
[1] 0.2628686
```

#### S4.9.4.3 Plot

```
df_causal_mor <-
  rbind(effect_fruitAB_mor,
        effect_rain_fruitAB_mor) |>
  group_by(effect) |>
  transmute(mu = median(relative_change),
            li = quantile(relative_change, 0.025),
            ls = quantile(relative_change, 0.975)) |>
  unique() |> ungroup() |>
  mutate(x =
    c('Fruit production',
      '*Rainfall          \n (fruit production)'),
```

```

y = "Fruits' morph. hypervolume")

causal_plot_mor_HV <-
  rbind(effect_fruitAB_mor,
        effect_rain_fruitAB_mor) |>
  group_by(effect) |>
  transmute(mu = median(relative_change),
            li = quantile(relative_change, 0.025),
            ls = quantile(relative_change, 0.975)) |>
  unique() |> ungroup() |>
  mutate(x =
    c('Fruit production',
      '*Rainfall \n (fruit production)'),
    y = "Fruits' morph. Hypervolume") |>
  ggplot() +
  geom_hline(yintercept = 0, linetype = 3, linewidth = 0.25) +
  geom_errorbar(aes(x, ymin = li, ymax = ls),
                width = 0, linewidth = 1.25, alpha = 0.5,
                color = '#1e81b0') +
  geom_point(aes(x, mu), color = '#e28743', size = 2) +
  labs(y = 'Relative causal effect (%)', x = '') +
  facet_wrap(~y) +
  theme_classic() +
  theme(panel.grid = element_blank(),
        strip.background = element_blank(),
        axis.line = element_line(linewidth = 0.25),
        axis.ticks = element_line(linewidth = 0.25),
        #text = element_text(family = 'Times New Roman'),
        axis.text.x = element_text(angle = 45, hjust = 1, vjust = 1),
        axis.text = element_text(size = 8.5))

```

```
causal_plot_mor_HV
```

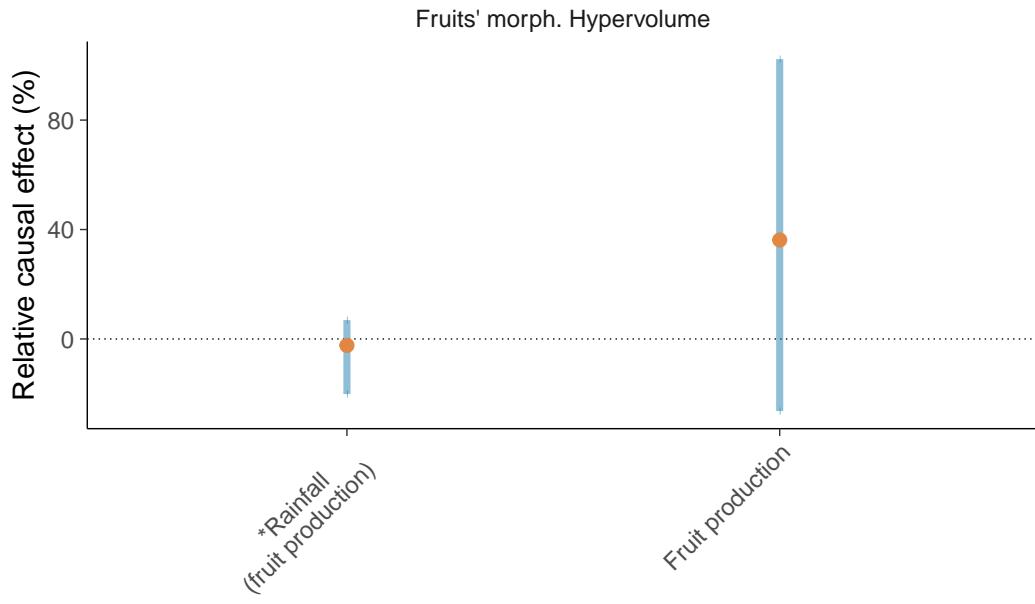


Figure S21: Causal effects of exposure variables on fruits' morphological functional space. An asterisk indicates effects mediated through the variable in parentheses

#### S4.9.5 $X \rightarrow$ fruits' functional space

```
knitr::kable(betas_H2_G_z[grep('PLAN_HV$', betas_H2_G_z[[1]]), ],
             digits = 2,
             caption = "Factors affecting fruits' nutritional functional space. Average values, 95% CIs, and probabilities of directional effects for the relationships assessed in the generative model. An effect is marked as TRUE if P(effect > 0) > 0.7 or P(effect < 0) > 0.7 or.")
```

Table S7: Factors affecting fruits' nutritional functional space. Average values, 95% CIs, and probabilities of directional effects for the relationships assessed in the generative model. An effect is marked as TRUE if  $P(\text{effect} > 0) > 0.7$  or  $P(\text{effect} < 0) > 0.7$  or.

Directional effect	Mean	0.025%	0.975%	$P(\text{effect} > 0)$	$P(\text{effect} < 0)$	effect
temp -> PLAN_HV	-0.01	-0.14	0.11	0.42	0.58	FALSE
rain -> PLAN_HV	0.07	-0.06	0.20	0.85	0.15	TRUE
fruitAB -> PLAN_HV	-0.14	-0.29	0.00	0.03	0.97	TRUE

Directional effect	Mean	0.025%	0.975%	P(effect > 0)	P(effect < 0)	effect
fruitS -> PLAN_HV	0.07	-0.05	0.19	0.88	0.12	TRUE
nut -> PLAN_HV	0.59	0.48	0.70	1.00	0.00	TRUE
mor -> PLAN_HV	0.47	0.38	0.56	1.00	0.00	TRUE
BIRD -> PLAN_HV	0.03	-0.06	0.12	0.74	0.26	TRUE

Relative importance of the *rainfall* explaining fruits' functional space, compared to fruit production, fruit richness and birds functional space.

```
zzz <-
  sapply(c(3, 4, 7), FUN =
    function(x) {
      ((abs(betas_H2_G_z[grep('PLAN_HV$', betas_H2_G_z[[1]]), ]$Mean[2]) -
        abs(betas_H2_G_z[grep('PLAN_HV$', betas_H2_G_z[[1]]), ]$Mean[x])) /
       abs(betas_H2_G_z[grep('PLAN_HV$', betas_H2_G_z[[1]]), ]$Mean[x])) * 100
    })
mean(zzz); sqrt(sd(zzz)/length(zzz))
```

[1] 26.23529

[1] 5.639259

```
tibble(var = betas_H2_G_z[grep('PLAN_HV$', betas_H2_G_z[[1]]), ][[1]][c(3, 4, 7)],
`Importance of rainfall` = zzz)
```

```
# A tibble: 3 x 2
  var           `Importance of rainfall`
  <chr>          <dbl>
1 fruitAB -> PLAN_HV      -52.1
2 fruitS -> PLAN_HV      -1.71
3 BIRD -> PLAN_HV       132.
```

Relative importance of the *fruit production* affecting fruits' functional space, compared to rainfall, fruit richness and birds functional space.

```
zzz <-
  sapply(c(2, 4, 7), FUN =
    function(x) {
      ((abs(betas_H2_G_z[grep('PLAN_HV$', betas_H2_G_z[[1]]), ]$Mean[3]) -
        abs(betas_H2_G_z[grep('PLAN_HV$', betas_H2_G_z[[1]]), ]$Mean[x])) /
       abs(betas_H2_G_z[grep('PLAN_HV$', betas_H2_G_z[[1]]), ]$Mean[x])) * 100
    })
mean(zzz); sqrt(sd(zzz)/length(zzz))
```

```
[1] 199.6364
```

```
[1] 7.317919
```

```
tibble(var = betas_H2_G_z[grep('PLAN_HV$', betas_H2_G_z[[1]]), ][[1]][c(2, 4, 7)],  
`Importance of fruit production` = zzz)
```

```
# A tibble: 3 x 2  
  var           `Importance of fruit production`  
  <chr>          <dbl>  
1 rain -> PLAN_HV      109.  
2 fruitS -> PLAN_HV     105.  
3 BIRD -> PLAN_HV      385.
```

#### S4.9.5.1 Rainfall → fruits' functional space

```
effect_rain_PLAN <-  
causal_effects(  
  pars = post_H2_G,  
  alpha_1 = 'alpha_PLAN_HV',  
  f_1 = 'f_PLAN_HV',  
  beta_1 = 'beta_rain_PLAN_HV',  
  var = dat$z_rainfall,  
  y_obs = d$HV_plant,  
  level_1 = T  
)
```

Percentage increase of intervening *Rainfall*: average value and 95% CI.

```
quantile(effect_rain_PLAN$relative_change,  
        c(0.025, 0.5, 0.975))
```

```
 2.5%      50%    97.5%  
-22.07176  23.81410  68.73723
```

Probability of the causal effect being positive:

```
unique(effect_rain_PLAN$p_effect)
```

```
[1] 0.8545727
```

#### S4.9.5.2 Fruit production → fruits' functional space

```

effect_fruitAB_PLAN <-
causal_effects(
  pars = post_H2_G,
  alpha_1 = 'alpha_PLAN_HV',
  f_1 = 'f_PLAN_HV',
  beta_1 = 'beta_fruitAB_PLAN_HV',
  var = dat$fruit_abun,
  y_obs = d$HV_plant,
  level_1 = T
)

```

Percentage increase of intervening *fruit production*: average value and 95% CI.

```

quantile(effect_fruitAB_PLAN$relative_change,
c(0.025, 0.5, 0.975))

```

2.5%	50%	97.5%
-89.9122395	-43.8393942	0.3336013

Probability of the causal effect being positive:

```

unique(effect_fruitAB_PLAN$p_effect)

```

[1] 0.02598701

#### S4.9.5.3 Rainfall → Fruit production → fruits' functional space

```

effect_rain_fruitAB_PLAN <-
causal_effects(
  pars = post_H2_G,
  alpha_1 = 'alpha_fruitAB',
  f_1 = 'f_fruitAB',
  beta_1 = 'beta_rain_fruitAB',
  alpha_2 = 'alpha_PLAN_HV',
  f_2 = 'f_PLAN_HV',
  beta_2 = 'beta_fruitAB_PLAN_HV',
  var = dat$z_rainfall,
  y_obs = d$HV_plants_morfo,
  level_1 = F,
  level_2 = T
)

```

Percentage increase of intervening *rainfall (mediation path)*: average value and 95% CI.

```

quantile(effect_rain_fruitAB_PLAN$relative_change,
c(0.025, 0.5, 0.975))

```

2.5%	50%	97.5%
-3.101406	2.140541	10.830891

Probability of the causal effect being positive:

```
unique(effect_rain_fruitAB_PLAN$p_effect)
```

```
[1] 0.808096
```

#### S4.9.5.4 Plot

```
df_causal_PLAN <-  
  rbind(effect_rain_PLAN,  
        effect_fruitAB_PLAN,  
        effect_rain_fruitAB_PLAN) |>  
  group_by(effect) |>  
  transmute(mu = median(relative_change),  
            li = quantile(relative_change, 0.025),  
            ls = quantile(relative_change, 0.975)) |>  
  unique() |> ungroup() |>  
  mutate(x =  
    c('Rainfall',  
      'Fruit production',  
      '*Rainfall      \n (fruit production)'),  
    y = "Fruits' total hypervolume")  
  
causal_plot_PLAN_HV <-  
  rbind(effect_rain_PLAN,  
        effect_fruitAB_PLAN,  
        effect_rain_fruitAB_PLAN) |>  
  group_by(effect) |>  
  transmute(mu = median(relative_change),  
            li = quantile(relative_change, 0.025),  
            ls = quantile(relative_change, 0.975)) |>  
  unique() |> ungroup() |>  
  mutate(x =  
    c('Rainfall',  
      'Fruit production',  
      '*Rainfall      \n (fruit production)'),  
    y = "Fruits' total Hypervolume") |>  
  ggplot() +  
  geom_hline(yintercept = 0, linetype = 3, linewidth = 0.25) +  
  geom_errorbar(aes(x, ymin = li, ymax = ls),  
                width = 0, linewidth = 1.25, alpha = 0.5,  
                color = '#1e81b0') +  
  geom_point(aes(x, mu), color = '#e28743', size = 2) +  
  labs(y = 'Relative causal effect (%)', x = '') +  
  facet_wrap(~y) +  
  theme_classic() +  
  theme(panel.grid = element_blank(),  
        strip.background = element_blank(),  
        axis.line = element_line(linewidth = 0.25),  
        axis.ticks = element_line(linewidth = 0.25),  
        # text = element_text(family = 'Times New Roman'),  
        axis.text.x = element_text(angle = 45, hjust = 1, vjust = 1),  
        axis.text = element_text(size = 8.5))
```

```
causal_plot_PLAN_HV
```

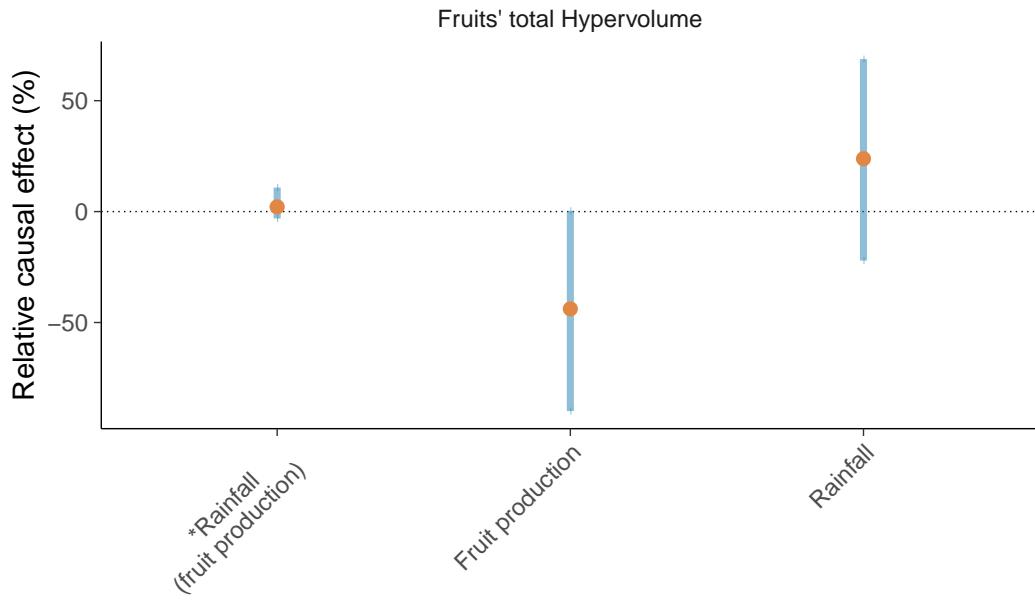


Figure S22: Causal effects of exposure variables on fruits' morphological functional space. An asterisk indicates effects mediated through the variable in parentheses

#### S4.9.6 Network metrics

#### S4.9.7 $X \rightarrow \text{Nestedness}$

```
knitr::kable(betas_nestednessG_z[grep('NSS$', betas_nestednessG_z[[1]]), ],
  digits = 2,
  caption = "Factors affecting fruits' nutritional functional space. Average values, 95% CIs, and prob")
```

Table S8: Factors affecting fruits' nutritional functional space. Average values, 95% CIs, and probabilities of directional effects for the relationships assessed in the generative model. An effect is marked as TRUE if  $P(\text{effect} > 0) > 0.7$  or  $P(\text{effect} < 0) > 0.7$  or.

Directional effect	Mean	0.025%	0.975%	$P(\text{effect} > 0)$	$P(\text{effect} < 0)$	effect
temp -> NSS	0.18	-0.16	0.54	0.84	0.16	TRUE
rain -> NSS	-0.17	-0.50	0.16	0.16	0.84	TRUE
fruitAB -> NSS	-0.22	-0.64	0.20	0.15	0.85	TRUE
fruitS -> NSS	0.02	-0.33	0.35	0.56	0.44	FALSE

Directional effect	Mean	0.025%	0.975%	P(effect > 0)	P(effect < 0)	effect
nut -> NSS	0.02	-0.39	0.45	0.52	0.48	FALSE
mor -> NSS	0.22	-0.13	0.58	0.89	0.11	TRUE
PLAN -> NSS	-0.29	-0.78	0.20	0.12	0.88	TRUE
BIRD -> NSS	-0.08	-0.33	0.17	0.25	0.75	TRUE

Relative importance of the fruits' functional space explaining network nestedness, compared to temperature, fruit production, fruit richness and birds functional space.

```
zzz <-  
  sapply(c(1:3, 6,8), FUN =  
    function(x) {  
      ((abs(betas_nestednessG_z[grep('NSS$', betas_nestednessG_z[[1]]), ]$Mean[7]) -  
       abs(betas_nestednessG_z[grep('NSS$', betas_nestednessG_z[[1]]), ]$Mean[x])) /  
       abs(betas_nestednessG_z[grep('NSS$', betas_nestednessG_z[[1]]), ]$Mean[x])) * 100  
    })  
  
mean(zzz); sqrt(sd(zzz)/length(zzz))
```

[1] 91.95376

[1] 4.417185

```
tibble(var = betas_nestednessG_z[grep('NSS$', betas_nestednessG_z[[1]]), ][[1]][c(1:3, 6,8)],  
`Importance of fruit's functional space` = zzz)
```

```
# A tibble: 5 x 2  
  var           `Importance of fruit's functional space`  
  <chr>          <dbl>  
1 temp -> NSS            62.2  
2 rain -> NSS            72.8  
3 fruitAB -> NSS          30.4  
4 mor -> NSS             31.2  
5 BIRD -> NSS            263.
```

#### S4.9.7.1 Temperature → Nestedness

```
effect_temp_NSS <-  
causal_effects(  
  pars = post_nestednessG,  
  alpha_1 = 'alpha_NSS',  
  f_1 = 'f_NSS',  
  beta_1 = 'beta_temp_NSS',  
  var = dat$z_temperature,  
  y_obs = d$nestedness,  
  level_1 = T  
)
```

Percentage increase of intervening *temperature*: average value and 95% CI.

```
quantile(effect_temp_NSS$relative_change,  
        c(0.025, 0.5, 0.975))
```

```
2.5%      50%      97.5%  
-37.50837 43.00947 129.07128
```

Probability of the causal effect being positive:

```
unique(effect_temp_NSS$p_effect)
```

```
[1] 0.8383333
```

#### S4.9.7.2 Rainfall → Nestedness

```
effect_rain_NSS <-  
  causal_effects(  
    pars = post_nestednessG,  
    alpha_1 = 'alpha_NSS',  
    f_1 = 'f_NSS',  
    beta_1 = 'beta_rain_NSS',  
    var = dat$z_rainfall,  
    y_obs = d$nestedness,  
    level_1 = T  
)
```

Percentage increase of intervening *Rainfall*: average value and 95% CI.

```
quantile(effect_rain_NSS$relative_change,  
        c(0.025, 0.5, 0.975))
```

```
2.5%      50%      97.5%  
-125.23755 -41.94568  40.70405
```

Probability of the causal effect being positive:

```
unique(effect_rain_NSS$p_effect)
```

```
[1] 0.155
```

#### S4.9.7.3 Fruit production → Nestedness

```

effect_fruitAB_NSS <-
causal_effects(
  pars = post_nestednessG,
  alpha_1 = 'alpha_NSS',
  f_1 = 'f_NSS',
  beta_1 = 'beta_fruitAB_NSS',
  var = dat$fruit_abun,
  y_obs = d$nestedness,
  level_1 = T
)

```

Percentage increase of intervening *fruit production*: average value and 95% CI.

```

quantile(effect_fruitAB_NSS$relative_change,
  c(0.025, 0.5, 0.975))

```

2.5%	50%	97.5%
-141.77314	-50.74249	44.73585

Probability of the causal effect being positive:

```

unique(effect_fruitAB_NSS$p_effect)

```

[1] 0.1458333

#### S4.9.7.4 *Fruit's morphological functional space* → Nestedness

```

effect_mor_NSS <-
causal_effects(
  pars = post_nestednessG,
  alpha_1 = 'alpha_NSS',
  f_1 = 'f_NSS',
  beta_1 = 'beta_mor_NSS',
  var = dat$HV_plants_morfo,
  y_obs = d$nestedness,
  level_1 = T
)

```

Percentage increase of intervening *fruit's morphological functional space*: average value and 95% CI.

```

quantile(effect_mor_NSS$relative_change,
  c(0.025, 0.5, 0.975))

```

2.5%	50%	97.5%
-35.44072	59.57016	154.32734

Probability of the causal effect being positive:

```
unique(effect_mor_NSS$p_effect)
```

```
[1] 0.8925
```

#### S4.9.7.5 *Fruit's functional space → Nestedness*

```
effect_PLAN_NSS <-
  causal_effects(
    pars = post_nestednessG,
    alpha_1 = 'alpha_NSS',
    f_1 = 'f_NSS',
    beta_1 = 'beta_PLAN_NSS',
    var = dat$HV_plant,
    y_obs = d$nestedness,
    level_1 = T
  )
```

Percentage increase of intervening *fruit's functional space*: average value and 95% CI.

```
quantile(effect_PLAN_NSS$relative_change,
         c(0.025, 0.5, 0.975))
```

2.5%	50%	97.5%
-185.15490	-67.35002	46.39846

Probability of the causal effect being positive:

```
unique(effect_PLAN_NSS$p_effect)
```

```
[1] 0.1241667
```

#### S4.9.7.6 *Bird's functional space → Nestedness*

```
effect_BIRD_NSS <-
  causal_effects(
    pars = post_nestednessG,
    alpha_1 = 'alpha_NSS',
    f_1 = 'f_NSS',
    beta_1 = 'beta_BIRD_NSS',
    var = dat$HV_bird,
    y_obs = d$nestedness,
    level_1 = T
  )
```

Percentage increase of intervening *bird's functional space*: average value and 95% CI.

```
quantile(effect_BIRD_NSS$relative_change,
         c(0.025, 0.5, 0.975))
```

```
2.5%      50%     97.5%
-93.39382 -23.23935 48.58048
```

Probability of the causal effect being positive:

```
unique(effect_BIRD_NSS$p_effect)
```

```
[1] 0.2516667
```

#### S4.9.7.7 Plot direct effects

```
direct_nss <- rbind(effect_temp_NSS,
                      effect_rain_NSS,
                      effect_fruitAB_NSS,
                      effect_mor_NSS,
                      effect_PLAN_NSS,
                      effect_BIRD_NSS)

direct_nss$effect <- as.factor(direct_nss$effect)

causal_plot_NSS_DIRECT <-
  direct_nss |>
    ggplot(aes(x = relative_change, y = effect, fill = after_stat(x))) +
    geom_density_ridges(scale = 1, rel_min_height = 0.005,
                        alpha = 0.5,
                        color = 'lightblue3',
                        fill = 'lightblue',
                        linewidth = 0.25) +
    # geom_density_ridges_gradient(scale = 1.15, rel_min_height = 0.01,
    #                               linewidth = 0.1) +
    # scale_fill_viridis_c(name = "Direction", option = "C") +
    # scale_fill_viridis_c(name = "", option = "C") +
    geom_vline(xintercept = 0, linetype = 3, linewidth = 0.35, color = 'tan1') +
    scale_y_discrete(labels = c(expression('Bifs' %->% "N"),
                                 expression('FP' %->% "N"),
                                 expression('Mifs' %->% "N"),
                                 expression('FTifs' %->% "N"),
                                 expression('R' %->% "N"),
                                 expression('T' %->% "N"))) +
    labs(x = 'Relative causal effect (%)', x = '') +
    theme_classic() +
    theme(panel.grid = element_blank(),
          strip.background = element_blank(),
          axis.line = element_line(linewidth = 0.25),
          axis.ticks = element_line(linewidth = 0.25),
          legend.position = 'none',
          #legend.position = c(0.925, 0.8),
          legend.background = element_blank(),
          legend.key.size = unit(0.25, 'cm'),
          legend.text = element_text(size = 5),
          legend.title = element_text(size = 6),
          axis.title.y = element_blank(),
          axis.text = element_text(size = 8.5))
```

```
causal_plot_NSS_DIRECT
```

Picking joint bandwidth of 9.82

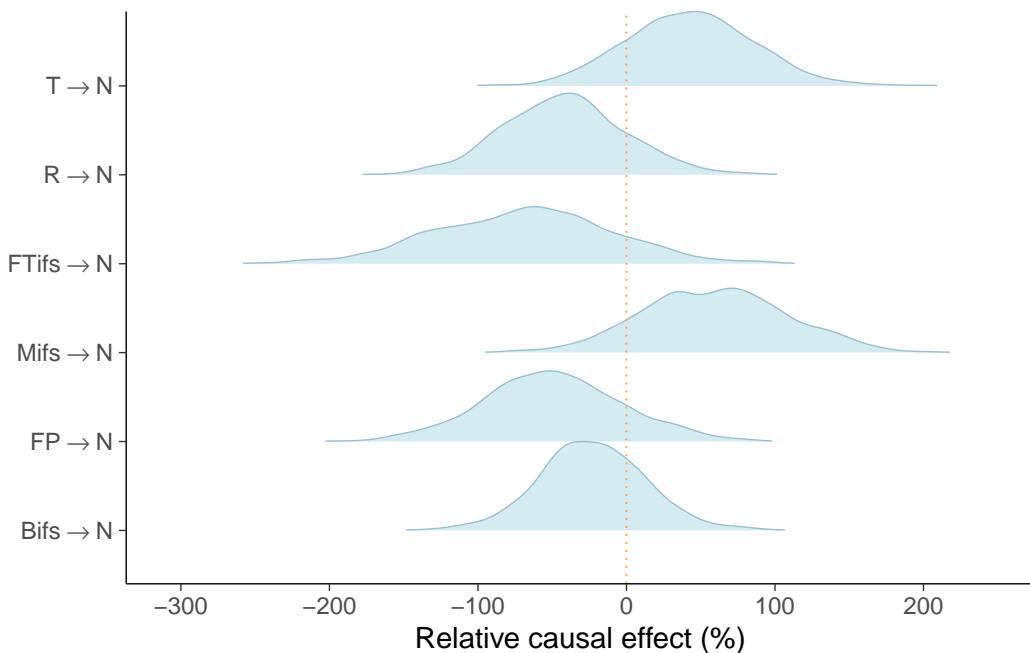


Figure S23: Causal direct effects of exposure variables on network nestedness

```
p_direct_nss <-
  sapply(levels(direct_nss$effect), FUN =
    function(i) {
      sapply(levels(direct_nss$effect), FUN =
        function(j) {
          i <- direct_nss[direct_nss$effect == i, ]$relative_change
          j <- direct_nss[direct_nss$effect == j, ]$relative_change
          mean(i < j)
        })
    })
colnames(p_direct_nss) <- c('Bifs', 'Fp', 'Mifs', 'FTifs', 'R', 'T')
rownames(p_direct_nss) <- c('Bifs', 'Fp', 'Mifs', 'FTifs', 'R', 'T')
```

```
knitr::kable(lower.tri.remove(p_direct_nss),
  digits = 3,
  caption = 'Probability that a causal direct effect on the row is lower than the effect on the column')
```

Table S9: Probability that a causal direct effect on the row is lower than the effect on the columns to explaining network nestedness

	Bifs	Fp	Mifs	FTifs	R	T
Bifs	0	0.667	0.085	0.731	0.642	0.132
Fp	NA	0.000	0.062	0.601	0.442	0.094
Mifs	NA	NA	0.000	0.903	0.948	0.603
FTifs	NA	NA	NA	0.000	0.369	0.061
R	NA	NA	NA	NA	0.000	0.117
T	NA	NA	NA	NA	NA	0.000

#### S4.9.7.8 Bottom-up effects (temperature and rainfall)

The following code uses the structure of the model to estimate the bottom-up effect of climatic predictors on network nestedness.

Temperature:

```
# Bottom-up effects (temperature)

# temp      fruitS  PLAN_HV NSS
# temp      nut_HV  PLAN_HV NSS
# temp      BIRD_HV PLAN_HV NSS
# temp      BIRD_HV NSS
# temp NSS

bottom_up_tem_NSS <-
  lapply(1, FUN =
    function(x) {

      f_nut <- grep('f_nut_HV', colnames(post_nestednessG$f))
      alpha_nut <- grep('alpha_nut_HV', colnames(post_nestednessG$alpha))

      f_Frs <- grep('f_fruitS', colnames(post_nestednessG$f))
      alpha_Frs <- grep('alpha_fruitS', colnames(post_nestednessG$alpha))

      f_FT <- grep('f_PLAN_HV', colnames(post_nestednessG$f))
      alpha_FT <- grep('alpha_PLAN_HV', colnames(post_nestednessG$alpha))

      f_BIRD <- grep('f_BIRD_HV', colnames(post_nestednessG$f))
      alpha_BIRD <- grep('alpha_BIRD_HV', colnames(post_nestednessG$alpha))

      f_NSS <- grep('f_NSS', colnames(post_nestednessG$f))
      alpha_NSS <- grep('alpha_NSS', colnames(post_nestednessG$alpha))

      contrast <-
        lapply(c(min, max), FUN =
          function(the_fun) {
            est <-
              with(post_nestednessG,
                {
                  # temp -> nut
                  mu_nut <-
                    apply(alpha[, alpha_nut], 1, mean) +
                    apply(f[, f_nut], 1, mean) +
                    beta$temp_nut_HV * the_fun(dat$z_temperature)
                })
            })
        })})
```

```

# temp -> Fr. S
mu_FrS <-
  apply(alpha[, alpha_Frs], 1, mean) +
  apply(f[, f_Frs], 1, mean) +
  beta$beta_temp_fruitS * the_fun(dat$z_temperature)

# temp -> Birds ifs
mu_BIRD <-
  apply(alpha[, alpha_BIRD], 1, mean) +
  apply(f[, f_BIRD], 1, mean) +
  beta$beta_temp_fruitS * the_fun(dat$z_temperature)

# FT ifs

mu_FT <-
  apply(alpha[, alpha_FT], 1, mean) +
  apply(f[, f_FT], 1, mean) +
  beta$beta_temp_PLAN_HV * the_fun(dat$z_temperature) +
  beta$beta_BIRD_PLAN_HV * mu_BIRD +
  beta$beta_nut_PLAN_HV * mu_nut +
  beta$beta_fruitS_PLAN_HV * mu_FrS

# temp -> nut -> modu
# temp -> Fr. S -> FT ifs -> modu
# temp -> Birds ifs -> FT ifs -> modu

mu_NSS <-
  apply(alpha[, alpha_NSS], 1, mean) +
  apply(f[, f_NSS], 1, mean) +
  beta$beta_temp_NSS * the_fun(dat$z_temperature) +
  beta$beta_BIRD_NSS * mu_BIRD +
  beta$beta_nut_NSS * mu_nut +
  beta$beta_fruitS_NSS * mu_FrS +
  beta$beta_PLAN_NSS * mu_FT

mean(d$nestedness) + mu_NSS * sd(d$nestedness)

}

)
est
})

contrast <- as.data.frame(do.call('cbind', contrast))
colnames(contrast) <- c('min', 'max')
contrast$contrast <- contrast$max - contrast$min
contrast$p_effect <- mean(contrast$contrast > 0)
contrast$relative_change <- ((contrast$contrast * 100) / mean(d$nestedness))
contrast$effect <- 'total_temp'
as_tibble(contrast)

})[[1]]

```

95% credibility interval of the relative increment from intervention

```
quantile(bottom_up_tem_NSS$relative_change, c(0.025, 0.5, 0.975))
```

	2.5%	50%	97.5%
-48.46252	34.91733	114.99509	

Probability that the increment is above zero

```
mean(bottom_up_tem_NSS$relative_change < 0)
```

```
[1] 0.1991667
```

Rainfall:

```
# Bottom-up effects (rainfall)

# rain      fruitAB nut_HV  PLAN_HV NSS
# rain      fruitAB mor_HV  PLAN_HV NSS
# rain      fruitAB mor_HV NSS
# rain      fruitAB BIRD_HV PLAN_HV NSS
# rain      fruitAB BIRD_HV NSS
# rain      fruitAB PLAN_HV NSS
# rain      fruitAB NSS
# rain      fruitS  PLAN_HV NSS
# rain      BIRD_HV PLAN_HV NSS
# rain      BIRD_HV NSS
# rain      PLAN_HV NSS
# rain      NSS

bottom_up_rain_NSS <-
  lapply(1, FUN =
    function(x) {

      f_mor <- grep('f_mor_HV', colnames(post_nestednessG$f))
      alpha_mor <- grep('alpha_mor_HV', colnames(post_nestednessG$alpha))

      f_nut <- grep('f_nut_HV', colnames(post_nestednessG$f))
      alpha_nut <- grep('alpha_nut_HV', colnames(post_nestednessG$alpha))

      f_Frs <- grep('f_fruitS', colnames(post_nestednessG$f))
      alpha_Frs <- grep('alpha_fruitS', colnames(post_nestednessG$alpha))

      f_Fp <- grep('f_fruitAB', colnames(post_nestednessG$f))
      alpha_Fp <- grep('alpha_fruitAB', colnames(post_nestednessG$alpha))

      f_FT <- grep('f_PLAN_HV', colnames(post_nestednessG$f))
      alpha_FT <- grep('alpha_PLAN_HV', colnames(post_nestednessG$alpha))

      f_BIRD <- grep('f_BIRD_HV', colnames(post_nestednessG$f))
      alpha_BIRD <- grep('alpha_BIRD_HV', colnames(post_nestednessG$alpha))

      f_NSS <- grep('f_NSS', colnames(post_nestednessG$f))
      alpha_NSS <- grep('alpha_NSS', colnames(post_nestednessG$alpha))

      contrast <-
        lapply(c(min, max), FUN =
          function(the_fun) {
            est <-
              with(post_nestednessG,
                {
                  # rain -> Fp
                  mu_Fp <-
                    apply(alpha[, alpha_Fp], 1, mean) +
                    apply(f[, f_Fp], 1, mean) +
                    beta$beta_rain_fruitAB * the_fun(dat$z_rainfall)

                  # rain -> nut
                  # Fp -> nut
                  mu_nut <-
                    apply(alpha[, alpha_nut], 1, mean) +

```

```

apply(f[, f_nut], 1, mean) +
beta$beta_rain_nut_HV * the_fun(dat$z_rainfall) +
beta$beta_fruitAB_nut_HV * mu_Fp

# rain -> morph
# Fp -> morph
mu_morp <-
  apply(alpha[, alpha_mor], 1, mean) +
  apply(f[, f_mor], 1, mean) +
  beta$beta_rain_mor_HV * the_fun(dat$z_rainfall) +
  beta$beta_fruitAB_mor_HV * mu_Fp

# rain -> Birds ifs
mu_BIRD <-
  apply(alpha[, alpha_BIRD], 1, mean) +
  apply(f[, f_BIRD], 1, mean) +
  beta$beta_rain_fruitS * the_fun(dat$z_rainfall)

# rain -> Fr. S
mu_FrS <-
  apply(alpha[, alpha_Frs], 1, mean) +
  apply(f[, f_Frs], 1, mean) +
  beta$beta_rain_fruitS * the_fun(dat$z_rainfall)

# FT ifs
mu_FT <-
  apply(alpha[, alpha_FT], 1, mean) +
  apply(f[, f_FT], 1, mean) +
  beta$beta_rain_PLAN_HV * the_fun(dat$z_rainfall) +
  beta$beta_BIRD_PLAN_HV * mu_BIRD +
  beta$beta_nut_PLAN_HV * mu_nut +
  beta$beta_fruitS_PLAN_HV * mu_FrS +
  beta$beta_fruitAB_PLAN_HV * mu_Fp +
  beta$beta_mor_PLAN_HV * mu_morp

# temp -> nut -> modu
# temp -> Fr. S -> FT ifs -> modu
# temp -> Birds ifs -> FT ifs -> modu

mu_NSS <-
  apply(alpha[, alpha_NSS], 1, mean) +
  apply(f[, f_NSS], 1, mean) +
  beta$beta_rain_NSS * the_fun(dat$z_temperature) +
  beta$beta_BIRD_NSS * mu_BIRD +
  beta$beta_mor_NSS * mu_morp +
  beta$beta_nut_NSS * mu_nut +
  beta$beta_fruitS_NSS * mu_FrS +
  beta$beta_fruitAB_NSS * mu_Fp +
  beta$beta_PLAN_NSS * mu_FT

mean(d$nestedness) + mu_NSS * sd(d$nestedness)
}

)
est
})

contrast <- as.data.frame(do.call('cbind', contrast))
colnames(contrast) <- c('min', 'max')
contrast$contrast <- contrast$max - contrast$min
contrast$p_effect <- mean(contrast$contrast < 0)
contrast$relative_change <- ((contrast$contrast * 100) / mean(d$nestedness))
contrast$effect <- 'total_Rain'
as_tibble(contrast)

})[[1]]

```

95% credibility interval of the relative increment from intervention

```
quantile(bottom_up_rain_NSS$relative_change, c(0.025, 0.5, 0.975))
```

2.5%	50%	97.5%
-126.50217	-48.31301	34.09529

Probability that the increment is above zero

```
mean(bottom_up_rain_NSS$relative_change < 0)
```

```
[1] 0.8858333
```

#### S4.9.7.9 Plot bottom-up effects

```
causal_plot_NSS_total <-  
  rbind(bottom_up_rain_NSS,  
        bottom_up_tem_NSS) |>  
  ggplot(aes(x = relative_change, y = effect, fill = after_stat(x))) +  
  geom_density_ridges(scale = 0.9, rel_min_height = 0.005,  
                      alpha = 0.5,  
                      color = 'lightblue3',  
                      fill = 'lightblue',  
                      linewidth = 0.25) +  
  # geom_density_ridges_gradient(scale = 1.15, rel_min_height = 0.01,  
  #                               linewidth = 0.1) +  
  # scale_fill_viridis_c(name = "Direction", option = "C") +  
  # scale_fill_viridis_c(name = "", option = "C") +  
  geom_vline(xintercept = 0, linetype = 3, linewidth = 0.35, color = 'tan1') +  
  scale_y_discrete(labels = c(expression('R (bottom-up)' %->% "N"),  
                             expression('T (bottom-up)' %->% "N")))+  
  labs(x = 'Relative causal effect (%)', x = '') +  
  theme_classic() +  
  theme(panel.grid = element_blank(),  
        strip.background = element_blank(),  
        axis.line = element_line(linewidth = 0.25),  
        axis.ticks = element_line(linewidth = 0.25),  
        legend.position = 'none',  
        #legend.position = c(0.925, 0.8),  
        #legend.background = element_blank(),  
        legend.key.size = unit(0.25, 'cm'),  
        legend.text = element_text(size = 5),  
        legend.title = element_text(size = 6),  
        axis.title.y = element_blank(),  
        axis.text = element_text(size = 8.5))  
  
causal_plot_NSS_total
```

Picking joint bandwidth of 8.75

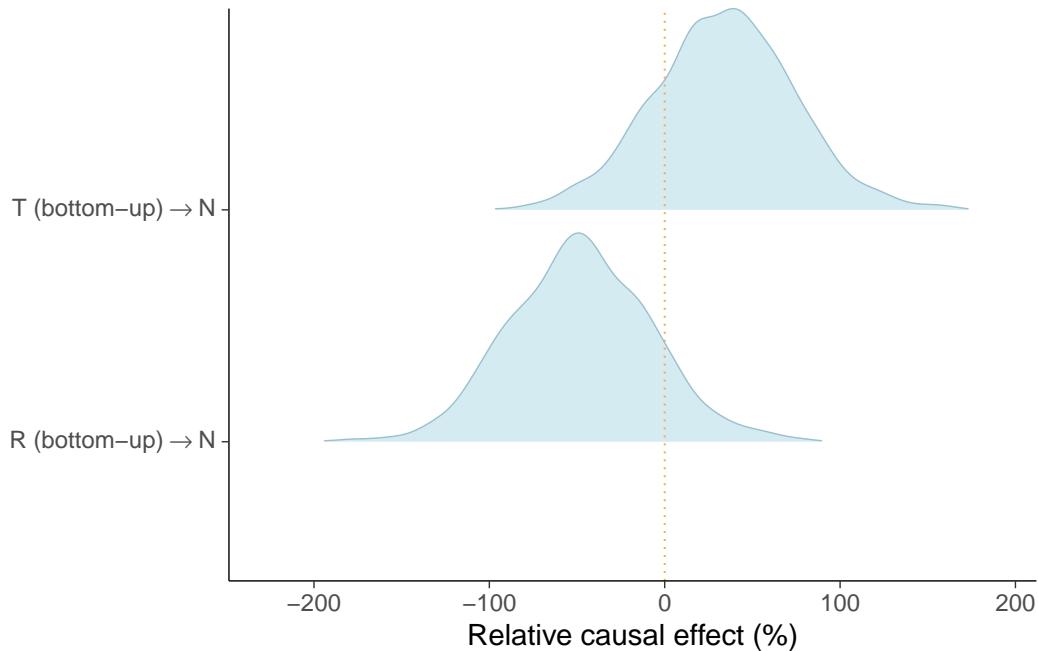


Figure S24: Causal bottom-up effect of intervening rainfall or temperature on the nestedness of seed dispersal networks

#### S4.9.7.10 Contrast between direct and bottom-up effect

Probability that direct causal effect of rainfall is higher than its bottom-up effect.

```
mean(effect_rain_NSS$relative_change > bottom_up_rain_NSS$relative_change)
```

```
[1] 0.5691667
```

Median and 95% credibility interval of the contrast between direct and bottom-up effects of rainfall

```
quantile(effect_rain_NSS$relative_change - bottom_up_rain_NSS$relative_change, c(0.025, 0.5, 0.975))
```

2.5%	50%	97.5%
-61.119868	4.920169	72.721530

Probability that direct causal effect of temperature is higher than its bottom-up effect.

```
mean(effect_temp_NSS$relative_change > bottom_up_tem_NSS$relative_change)
```

```
[1] 0.6958333
```

Median and 95% credibility interval of the contrast between direct and bottom-up effects of temperature

```
quantile(effect_temp_NSS$relative_change - bottom_up_tem_NSS$relative_change,  
        c(0.025, 0.5, 0.975))
```

	2.5%	50%	97.5%
-	-27.724551	8.171583	50.949322

```
causal_plot_NSS_total_CONT <-  
  rbind(tibble(val = effect_rain_NSS$relative_change -  
               bottom_up_rain_NSS$relative_change,  
               contrast = 'Direct vs bottom-up effect (R)'),  
        tibble(val = effect_temp_NSS$relative_change -  
               bottom_up_tem_NSS$relative_change,  
               contrast = 'Direct vs bottom-up effect (T)')) |>  
  ggplot(aes(x = val),  
         color = 'lighth') +  
  geom_density(alpha = 0.5,  
              color = 'lightblue3',  
              fill = 'lightblue',  
              linewidth = 0.25) +  
  facet_wrap(~ contrast, ncol = 2, scales = 'free') +  
  geom_vline(xintercept = 0, linetype = 3, linewidth = 0.35, color = 'tan1') +  
  labs(x = 'Relative causal effect (%)', y = 'Density') +  
  theme_classic() +  
  theme(panel.grid = element_blank(),  
        strip.background = element_blank(),  
        axis.line = element_line(linewidth = 0.25),  
        axis.ticks = element_line(linewidth = 0.25),  
        legend.position = 'none',  
        #legend.position = c(0.925, 0.8),  
        legend.background = element_blank(),  
        legend.key.size = unit(0.25, 'cm'),  
        legend.text = element_text(size = 5),  
        legend.title = element_text(size = 6),  
        axis.text = element_text(size = 8.5))  
  
causal_plot_NSS_total_CONT
```

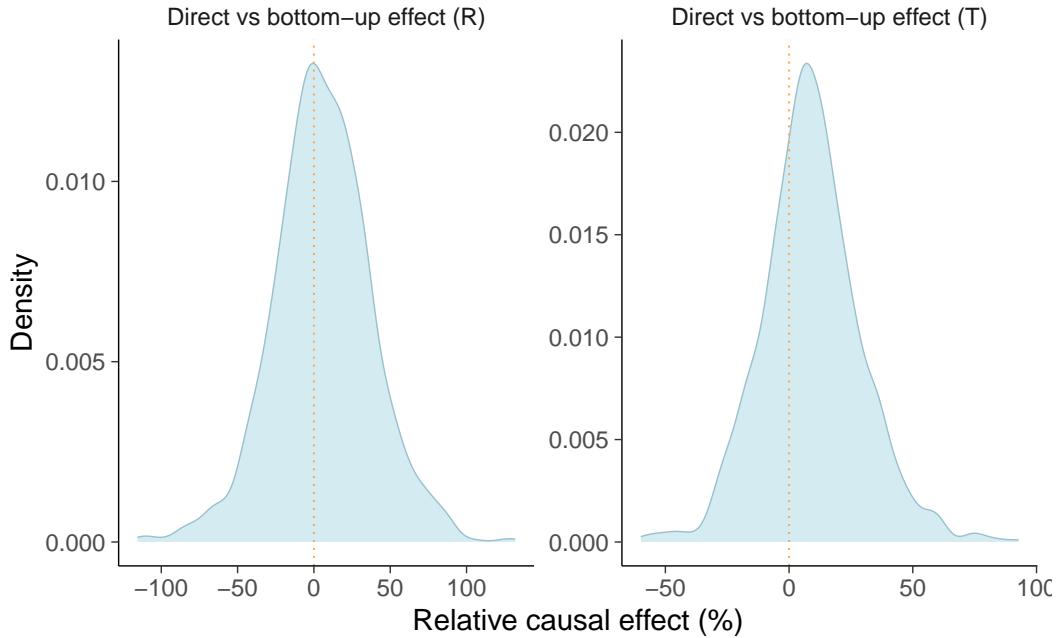


Figure S25: Contrast between causal direct and bottom-up effects from climatic predictors on nestedness of seed dispersal networks

#### S4.9.7.11 Variance decomposition

We extract the posterior distribution of the dispersion parameters of periodic (i.e., temporal correlation) and quadratic (i.e., spatial autocorrelation) kernels of Gaussian processes, and the dispersion parameter of the likelihood function. Then we used to decompose the variance explained by the different components of the model. The code below does that:

```

GP_nestedness <- mod_nestedness$draws(c('sigma_NSS',
                                         'sigma_f_NSS',
                                         'gamma_NSS',
                                         'eta_site_NSS'),
                                         format = 'df')

total_sigma_NSS <-
  GP_nestedness$sigma_NSS^2 +
  GP_nestedness$sigma_f_NSS^2 +
  GP_nestedness$eta_site_NSS^2
prop_GP_temp_NSS <- GP_nestedness$sigma_f_NSS^2 / total_sigma_NSS
prop_GP_site_NSS <- GP_nestedness$eta_site_NSS^2 / total_sigma_NSS
prop_PRED_NSS <- GP_nestedness$sigma_NSS^2 / total_sigma_NSS

GP_NSS <-
  tibble(component = c('Tiemporal processes',
                      'Spatial processes',
                      'Fized effects'),
        mu = c(mean(prop_GP_temp_NSS),
               mean(prop_GP_site_NSS),
               mean(prop_PRED_NSS)))
  
```

```

    mean(prop_PRED_NSS)),
sd = c(sd(prop_GP_temp_NSS),
       sd(prop_GP_site_NSS),
       sd(prop_PRED_NSS))

knitr::kable(GP_NSS,
            digits = 3,
            caption = 'Variance decomposition of the components explaining intra-annual pattern of nestedness in'

```

Table S10: Variance decomposition of the components explaining intra-annual pattern of nestedness in seed dispersal networks. Average and SD values of proportions are provided

component	mu	sd
Tiemporal processes	0.348	0.282
Spatial processes	0.067	0.110
Fized effects	0.585	0.274

Length-scale parameter of the periodic Gaussian processes indicating how quickly the correlation between two points decay as the distance (i.e., time) between them increases.

```
mean(GP_nestedness$gamma_NSS); sd(GP_nestedness$gamma_NSS)
```

```
[1] 1.051952
```

```
[1] 0.5479633
```

#### S4.9.8 $X \rightarrow Modularity$

```

knitr::kable(betas_MODU_G_z[grep('MODU$', betas_MODU_G_z[[1]]), ],
             digits = 2,
             caption = "Factors affecting fruits' nutritional functional space. Average values, 95% CIs, and probabilities of directional effects for the relationships assessed in the generative model. An effect is marked as TRUE if P(effect > 0) > 0.7 or P(effect < 0) > 0.7 or."

```

Table S11: Factors affecting fruits' nutritional functional space. Average values, 95% CIs, and probabilities of directional effects for the relationships assessed in the generative model. An effect is marked as TRUE if  $P(\text{effect} > 0) > 0.7$  or  $P(\text{effect} < 0) > 0.7$  or.

Directional effect	Mean	0.025%	0.975%	$P(\text{effect} > 0)$	$P(\text{effect} < 0)$	effect
temp -> MODU	-0.05	-0.34	0.27		0.38	0.62 FALSE

Directional effect	Mean	0.025%	0.975%	P(effect > 0)	P(effect < 0)	effect
rain -> MODU	0.16	-0.10	0.41	0.89	0.11	TRUE
fruitAB -> MODU	0.02	-0.08	0.12	0.68	0.32	FALSE
fruitS -> MODU	0.01	-0.10	0.10	0.55	0.45	FALSE
nut -> MODU	0.04	-0.05	0.14	0.79	0.21	TRUE
mor -> MODU	0.01	-0.08	0.11	0.60	0.40	FALSE
PLAN -> MODU	0.04	-0.06	0.14	0.77	0.23	TRUE
BIRD -> MODU	0.00	-0.09	0.10	0.50	0.50	FALSE

Relative importance of rainfall explaining network modularity, compared to temperature, fruit production, fruit richness and birds functional space.

```
zzz <-
  sapply(c(5, 7), FUN =
    function(x) {
      ((abs(betas_MODU_G_z[grep('MODU$', betas_MODU_G_z[[1]]), ]$Mean[2]) -
        abs(betas_MODU_G_z[grep('MODU$', betas_MODU_G_z[[1]]), ]$Mean[x])) /
       abs(betas_MODU_G_z[grep('MODU$', betas_MODU_G_z[[1]]), ]$Mean[x])) * 100
    })
mean(zzz); sqrt(sd(zzz)/length(zzz))
```

[1] 300.5711

[1] 4.574306

```
tibble(var = betas_MODU_G_z[grep('MODU$', betas_MODU_G_z[[1]]), ][[1]][c(5, 7)],
`Importance of rainfall` = zzz)
```

```
# A tibble: 2 x 2
  var           `Importance of rainfall`
  <chr>          <dbl>
1 nut -> MODU     271.
2 PLAN -> MODU    330.
```

#### S4.9.8.1 Rainfall → modularity

```
effect_rain_MODU <-
  causal_effects(
    pars = post_MODU_G,
    alpha_1 = 'alpha_MODU',
    f_1 = 'f_MODU',
    beta_1 = 'beta_rain_MODU',
    var = dat$z_rainfall,
    y_obs = d$modularity,
    level_1 = T
  )
```

Percentage increase of intervening *Rainfall*: average value and 95% CI.

```
quantile(effect_rain_MODU$relative_change,  
        c(0.025, 0.5, 0.975))
```

```
2.5%      50%      97.5%  
-34.46644 76.77813 231.04260
```

Probability of the causal effect being positive:

```
unique(effect_rain_MODU$p_effect)
```

```
[1] 0.8915542
```

#### S4.9.8.2 Fruits' nutritional functional space → modularity

```
effect_nut_MODU <-  
  causal_effects(  
    pars = post_MODU_G,  
    alpha_1 = 'alpha_MODU',  
    f_1 = 'f_MODU',  
    beta_1 = 'beta_nut_MODU',  
    var = dat$HV_plants_nut,  
    y_obs = d$modularity,  
    level_1 = T  
)
```

Percentage increase of intervening *fruits' nutritional functional space*: average value and 95% CI.

```
quantile(effect_nut_MODU$relative_change,  
        c(0.025, 0.5, 0.975))
```

```
2.5%      50%      97.5%  
-18.88033 16.26551 55.45604
```

Probability of the causal effect being positive:

```
unique(effect_nut_MODU$p_effect)
```

```
[1] 0.7926037
```

#### S4.9.8.3 Fruits' functional space → modularity

```

effect_PLAN_MODU <-
causal_effects(
  pars = post_MODU_G,
  alpha_1 = 'alpha_MODU',
  f_1 = 'f_MODU',
  beta_1 = 'beta_PLAN_MODU',
  var = dat$HV_plant,
  y_obs = d$modularity,
  level_1 = T
)

```

Percentage increase of intervening *fruits' functional space*: average value and 95% CI.

```

quantile(effect_PLAN_MODU$relative_change,
  c(0.025, 0.5, 0.975))

```

	2.5%	50%	97.5%
	-23.43328	14.74298	57.56109

Probability of the causal effect being positive:

```

unique(effect_PLAN_MODU$p_effect)

```

[1] 0.7666167

#### S4.9.8.4 Plot direct effects

```

causal_plot_MODU_DIRECT <-
rbind(effect_rain_MODU,
      effect_nut_MODU,
      effect_PLAN_MODU) |>
ggplot(aes(x = relative_change, y = effect, fill = after_stat(x))) +
geom_density_ridges(scale = 1, rel_min_height = 0.005,
  alpha = 0.5,
  color = 'lightblue3',
  fill = 'lightblue',
  linewidth = 0.25) +
# geom_density_ridges_gradient(scale = 1.15, rel_min_height = 0.01,
#   linewidth = 0.1) +
# scale_fill_viridis_c(name = "Direction", option = "C") +
#scale_y_discrete(labels = c(expression('Nifs' %->% "M"),
#  expression('FTifs' %->% "M"),
#  expression('R' %->% "M"))) +
lims(x = c(-100, 300)) +
labs(x = 'Relative causal effect (%)', x = '') +
theme_classic() +
theme(panel.grid = element_blank(),
  strip.background = element_blank(),
  axis.line = element_line(linewidth = 0.25),
  axis.ticks = element_line(linewidth = 0.25),
  legend.position = 'none',

```

```

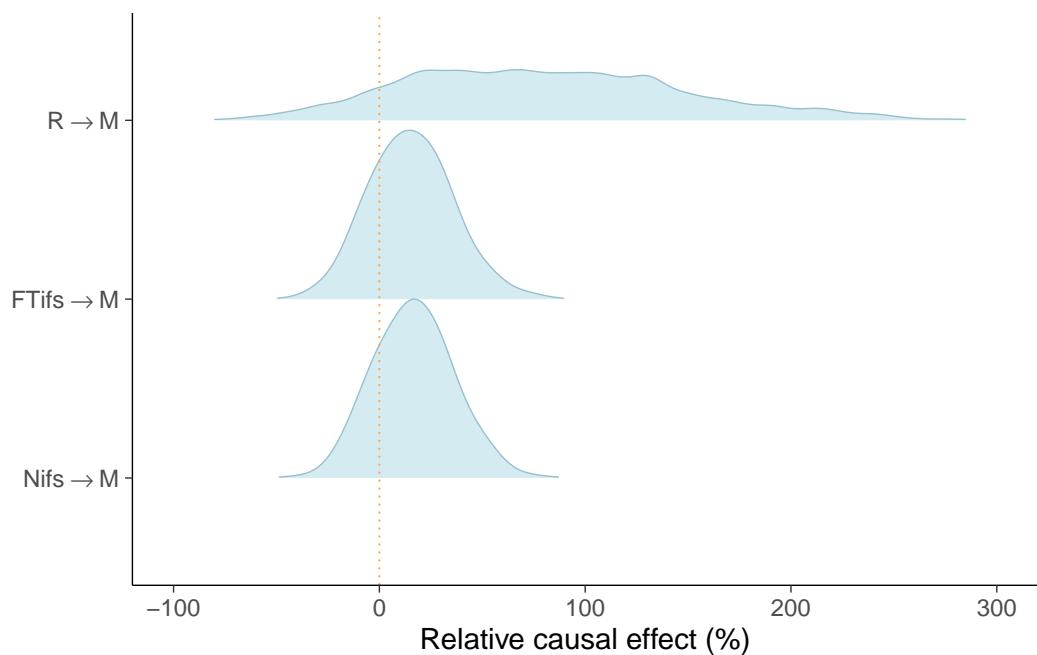
#legend.position = c(0.925, 0.8),
legend.background = element_blank(),
legend.key.size = unit(0.25, 'cm'),
legend.text = element_text(size = 5),
legend.title = element_text(size = 6),
axis.title.y = element_blank(),
axis.text = element_text(size = 8.5))

causal_plot_MODU_DIRECT

```

Picking joint bandwidth of 7.04

Warning: Removed 10 rows containing non-finite outside the scale range  
(`stat\_density\_ridges()`).



```

causal_plot_MODU_DIRECT

```

Picking joint bandwidth of 7.04

Warning: Removed 10 rows containing non-finite outside the scale range  
(`stat\_density\_ridges()`).

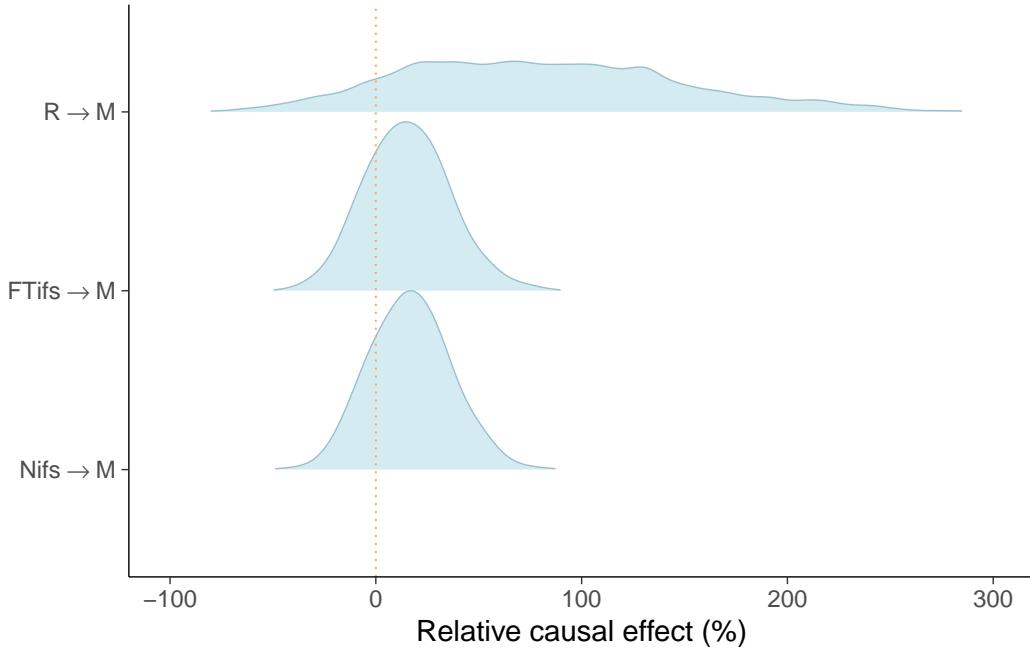


Figure S26: Causal direct effects of exposure variables on network modularity

#### S4.9.8.5 Bottom-up effects (temperature and rainfall)

The following code uses the structure of the model to estimate the bottom-up effect of climatic predictors on network modularity.

Temperature:

```
# Bottom-up effect (temperature)

# temp -> nut -> modu
# temp -> Fr. S -> FT ifs -> modu
# temp -> Birds ifs -> FT ifs -> modu

bottom_up_tem_MODU <-
  lapply(1, FUN =
    function(x) {

      f_nut <- grep('f_nut_HV', colnames(post_MODU_G$f))
      alpha_nut <- grep('alpha_nut_HV', colnames(post_MODU_G$alpha))

      f_Frs <- grep('f_fruitS', colnames(post_MODU_G$f))
      alpha_Frs <- grep('alpha_fruitS', colnames(post_MODU_G$alpha))

      f_FT <- grep('f_PLAN_HV', colnames(post_MODU_G$f))
      alpha_FT <- grep('alpha_PLAN_HV', colnames(post_MODU_G$alpha))

      f_BIRD <- grep('f_BIRD_HV', colnames(post_MODU_G$f))
      alpha_BIRD <- grep('alpha_BIRD_HV', colnames(post_MODU_G$alpha))

      f_MODU <- grep('f_MODU', colnames(post_MODU_G$f))
```

```

alpha_MODU <- grep('alpha_MODU', colnames(post_MODU_G$alpha))

contrast <-
  lapply(c(min, max), FUN =
    function(the_fun) {
      est <-
        with(post_MODU_G,
        {
          # temp -> nut
          mu_nut <-
            apply(alpha[, alpha_nut], 1, mean) +
            apply(f[, f_nut], 1, mean) +
            beta$beta_temp_nut_HV * the_fun(dat$z_temperature)

          # temp -> Fr. S
          mu_FrS <-
            apply(alpha[, alpha_Frs], 1, mean) +
            apply(f[, f_Frs], 1, mean) +
            beta$beta_temp_fruitS * the_fun(dat$z_temperature)

          # temp -> Birds ifs
          mu_BIRD <-
            apply(alpha[, alpha_BIRD], 1, mean) +
            apply(f[, f_BIRD], 1, mean) +
            beta$beta_temp_fruitS * the_fun(dat$z_temperature)

          # FT ifs
          mu_FT <-
            apply(alpha[, alpha_FT], 1, mean) +
            apply(f[, f_FT], 1, mean) +
            beta$beta_temp_PLAN_HV * the_fun(dat$z_temperature) +
            beta$beta_BIRD_PLAN_HV * mu_BIRD +
            beta$beta_nut_PLAN_HV * mu_nut +
            beta$beta_fruitS_PLAN_HV * mu_FrS

          # temp -> nut -> modu
          # temp -> Fr. S -> FT ifs -> modu
          # temp -> Birds ifs -> FT ifs -> modu

          mu_MODU <-
            apply(alpha[, alpha_MODU], 1, mean) +
            apply(f[, f_MODU], 1, mean) +
            beta$beta_temp_MODU * the_fun(dat$z_temperature) +
            beta$beta_BIRD_MODU * mu_BIRD +
            beta$beta_nut_MODU * mu_nut +
            beta$beta_fruitS_MODU * mu_FrS +
            beta$beta_PLAN_MODU * mu_FT

            inv_logit(mu_MODU)
        })
    }
  )
})[[1]]
})[[1]]
```

95% credibility interval of the relative increment from intervention

```
quantile(bottom_up_tem_MODU$relative_change, c(0.025, 0.5, 0.975))
```

	2.5%	50%	97.5%
	-131.744557	-9.807862	102.910099

Probability that the increment is above zero

```
mean(bottom_up_tem_MODU$contrast < 0)
```

```
[1] 0.5707146
```

Rainfall:

```
# Bottom-up effect (rainfall)

# rain -> fruitAB -> nut_HV -> PLAN_HV MODU
# rain -> fruitAB -> nut_HV -> MODU
# rain -> fruitAB -> mor_HV -> PLAN_HV -> MODU
# rain -> fruitAB -> BIRD_HV -> PLAN_HV -> MODU
# rain -> fruitAB -> PLAN_HV -> MODU
# rain -> fruitS -> PLAN_HV MODU
# rain -> BIRD_HV -> PLAN_HV -> MODU
# rain -> PLAN_HV -> MODU
# rain -> MODU

bottom_up_rain_MODU <-
  lapply(1, FUN =
    function(x) {

      f_mor <- grep('f_mor_HV', colnames(post_MODU_G$f))
      alpha_mor <- grep('alpha_mor_HV', colnames(post_MODU_G$alpha))

      f_nut <- grep('f_nut_HV', colnames(post_MODU_G$f))
      alpha_nut <- grep('alpha_nut_HV', colnames(post_MODU_G$alpha))

      f_Frs <- grep('f_fruitS', colnames(post_MODU_G$f))
      alpha_Frs <- grep('alpha_fruitS', colnames(post_MODU_G$alpha))

      f_Fp <- grep('f_fruitAB', colnames(post_MODU_G$f))
      alpha_Fp <- grep('alpha_fruitAB', colnames(post_MODU_G$alpha))

      f_FT <- grep('f_PLAN_HV', colnames(post_MODU_G$f))
      alpha_FT <- grep('alpha_PLAN_HV', colnames(post_MODU_G$alpha))

      f_BIRD <- grep('f_BIRD_HV', colnames(post_MODU_G$f))
      alpha_BIRD <- grep('alpha_BIRD_HV', colnames(post_MODU_G$alpha))

      f_MODU <- grep('f_MODU', colnames(post_MODU_G$f))
      alpha_MODU <- grep('alpha_MODU', colnames(post_MODU_G$alpha))

      contrast <-
        lapply(c(min, max), FUN =
          function(the_fun) {
            est <-
```

```

with(post_MODU_G,
{
  # rain -> Fp
  mu_Fp <-
    apply(alpha[, alpha_Fp], 1, mean) +
    apply(f[, f_Fp], 1, mean) +
    beta$beta_rain_fruitAB * the_fun(dat$z_rainfall)

  # rain -> nut
  # Fp -> nut
  mu_nut <-
    apply(alpha[, alpha_nut], 1, mean) +
    apply(f[, f_nut], 1, mean) +
    beta$beta_rain_nut_HV * the_fun(dat$z_rainfall) +
    beta$beta_fruitAB_nut_HV * mu_Fp

  # rain -> morph
  # Fp -> morph
  mu_morp <-
    apply(alpha[, alpha_mor], 1, mean) +
    apply(f[, f_mor], 1, mean) +
    beta$beta_rain_mor_HV * the_fun(dat$z_rainfall) +
    beta$beta_fruitAB_mor_HV * mu_Fp

  # rain -> Birds ifs
  mu_BIRD <-
    apply(alpha[, alpha_BIRD], 1, mean) +
    apply(f[, f_BIRD], 1, mean) +
    beta$beta_rain_fruits * the_fun(dat$z_rainfall)

  # rain -> Fr. S
  mu_FrS <-
    apply(alpha[, alpha_Frs], 1, mean) +
    apply(f[, f_Frs], 1, mean) +
    beta$beta_rain_fruitS * the_fun(dat$z_rainfall)

  # FT ifs
  mu_FT <-
    apply(alpha[, alpha_FT], 1, mean) +
    apply(f[, f_FT], 1, mean) +
    beta$beta_rain_PLAN_HV * the_fun(dat$z_rainfall) +
    beta$beta_BIRD_PLAN_HV * mu_BIRD +
    beta$beta_nut_PLAN_HV * mu_nut +
    beta$beta_fruitS_PLAN_HV * mu_FrS +
    beta$beta_fruitAB_PLAN_HV * mu_Fp +
    beta$beta_mor_PLAN_HV * mu_morp

  # temp -> nut -> modu
  # temp -> Fr. S -> FT ifs -> modu
  # temp -> Birds ifs -> FT ifs -> modu

  mu_MODU <-
    apply(alpha[, alpha_MODU], 1, mean) +
    apply(f[, f_MODU], 1, mean) +
    beta$beta_rain_MODU * the_fun(dat$z_temperature) +
    beta$beta_BIRD_MODU * mu_BIRD +
    beta$beta_mor_MODU * mu_morp +
    beta$beta_nut_MODU * mu_nut +
    beta$beta_fruitS_MODU * mu_FrS +
    beta$beta_fruitAB_MODU * mu_Fp +
    beta$beta_PLAN_MODU * mu_FT

  inv_logit(mu_MODU)
}

)
est

```

```

    })

contrast <- as.data.frame(do.call('cbind', contrast))
colnames(contrast) <- c('min', 'max')
contrast$contrast <- contrast$max - contrast$min
contrast$p_effect <- mean(contrast$contrast > 0)
contrast$relative_change <- ((contrast$contrast * 100) / mean(dat$modularity))
contrast$effect <- 'total_Rain'
as_tibble(contrast)

})[[1]]

```

95% credibility interval of the relative increment from intervention

```
quantile(bottom_up_rain_MODU$relative_change, c(0.025, 0.5, 0.975))
```

```

2.5%      50%      97.5%
-41.15901  61.75784 162.00855

```

Probability that the increment is above zero

```
mean(bottom_up_rain_MODU$contrast > 0)
```

```
[1] 0.8905547
```

#### S4.9.8.6 Plot bottom-up effects

```

causal_plot_MODU_total <-
  rbind(bottom_up_rain_MODU,
        bottom_up_tem_MODU) |>
  ggplot(aes(x = relative_change, y = effect, fill = after_stat(x))) +
  geom_density_ridges(scale = 0.9, rel_min_height = 0.005,
                      alpha = 0.5,
                      color = 'lightblue3',
                      fill = 'lightblue',
                      linewidth = 0.25) +
  # geom_density_ridges_gradient(scale = 1.15, rel_min_height = 0.01,
  #                               linewidth = 0.1) +
  # scale_fill_viridis_c(name = "Direction", option = "C") +
  # scale_y_discrete(labels = c(expression('R (bottom-up)' %-%>% "M"),
  #                             expression('T (bottom-up)' %-%>% "M")))) +
  labs(x = 'Relative causal effect (%)', x = '') +
  theme_classic() +
  theme(panel.grid = element_blank(),
        strip.background = element_blank(),
        axis.line = element_line(linewidth = 0.25),
        axis.ticks = element_line(linewidth = 0.25),
        legend.position = 'none',
        #legend.position = c(0.925, 0.8),
        legend.background = element_blank(),

```

```

legend.key.size = unit(0.25, 'cm'),
legend.text = element_text(size = 5),
legend.title = element_text(size = 6),
axis.title.y = element_blank(),
axis.text = element_text(size = 8.5))

causal_plot_MODU_total

```

Picking joint bandwidth of 10.9

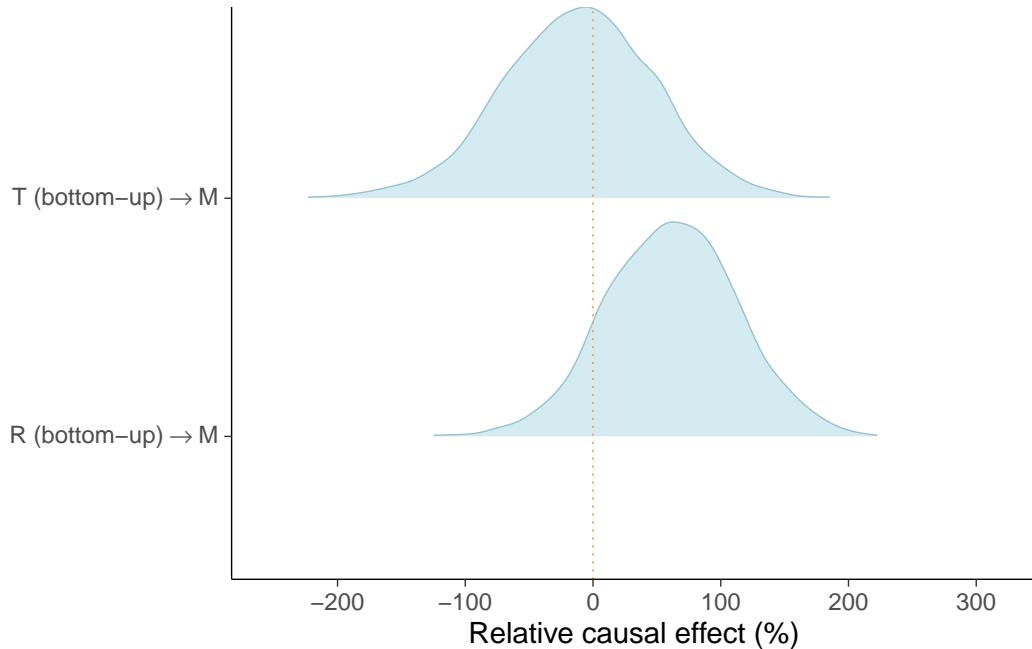


Figure S27: Causal bottom-up effect of intervening rainfall or temperature on the modularity of seed dispersal networks

#### S4.9.8.7 Contrast between direct and bottom-up effect

Probability that direct causal effect of rainfall is higher than its bottom-up effect.

```
mean(effect_rain_MODU$relative_change > bottom_up_rain_MODU$relative_change)
```

```
[1] 0.8235882
```

Median and 95% credibility interval of the contrast between direct and bottom-up effects of rainfall

```
quantile(effect_rain_MODU$relative_change - bottom_up_rain_MODU$relative_change,  
        c(0.025, 0.5, 0.975))
```

```
2.5%      50%     97.5%  
-14.07746 15.46408 82.27843
```

```
causal_plot_MODU_total_CONT <-  
  tibble(val = effect_rain_MODU$relative_change -  
         bottom_up_rain_MODU$relative_change,  
         contrast = 'Direct vs bottom-up effect (R)') |>  
  ggplot(aes(x = val),  
         color = 'lightgreen') +  
  geom_density(alpha = 0.5,  
               color = 'lightblue3',  
               fill = 'lightblue',  
               linewidth = 0.25) +  
  facet_wrap(~ contrast) +  
  geom_vline(xintercept = 0, linetype = 3, linewidth = 0.35, color = 'tan1') +  
  labs(x = 'Relative causal effect (%)', y = 'Density') +  
  theme_classic() +  
  theme(panel.grid = element_blank(),  
        strip.background = element_blank(),  
        axis.line = element_line(linewidth = 0.25),  
        axis.ticks = element_line(linewidth = 0.25),  
        legend.position = 'none',  
        #legend.position = c(0.925, 0.8),  
        legend.background = element_blank(),  
        legend.key.size = unit(0.25, 'cm'),  
        legend.text = element_text(size = 5),  
        legend.title = element_text(size = 6),  
        axis.text = element_text(size = 8.5))  
  
causal_plot_MODU_total_CONT
```

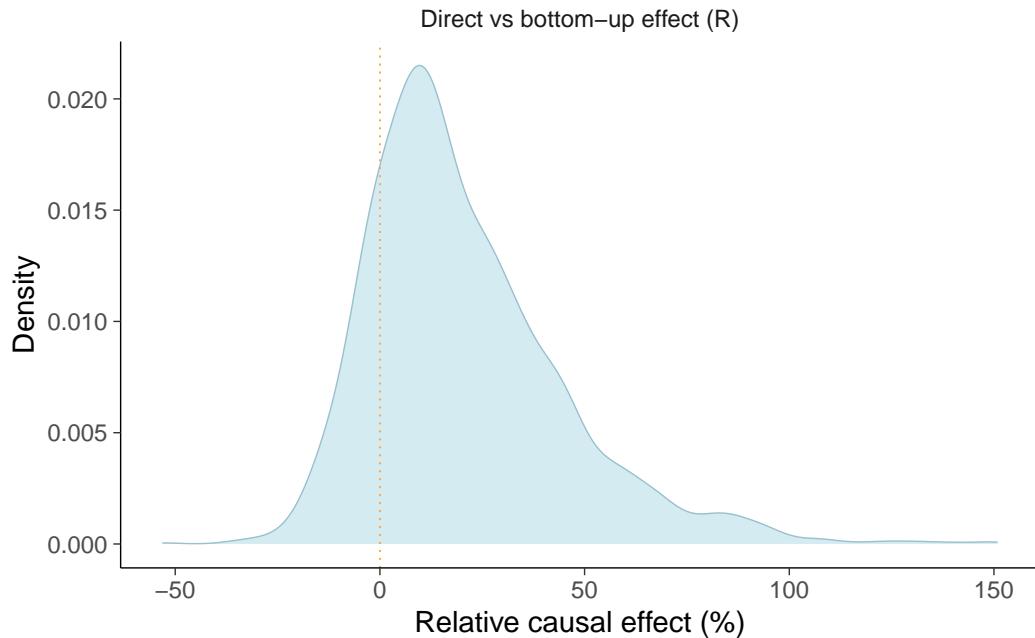


Figure S28: Contrast between causal direct and bottom-up effects from climatic predictors on modularity of seed dispersal networks

#### S4.9.8.8 Variance decomposition

Procedures for conducting variance decomposition analysis can be consulted at the `generative_quantity` box in the Stan model. Here we extract the posterior distribution of the explained variance:

```
GP_modu <- mod_MODU_G$draws(c('gamma_MODU', 'prop_GP_temp',
                                'prop_fixed', 'prop_GP_space'), format = 'df')

GP_MODU <-
  do.call('rbind',
  lapply(2:4, FUN =
    function(x) {
      tibble(component = c('Temporal processes',
                          'Fixed effects',
                          'Spatial processes')[x-1],
            mu = median(GP_modu[[x]]),
            sd = sd(GP_modu[[x]]))
    }))

knitr::kable(GP_MODU,
             digits = 3,
             caption = 'Variance decomposition of the components explaining intra-annual pattern of modularity in
```

Table S12: Variance decomposition of the components explaining intra-annual pattern of modularity in seed dispersal networks. Average and SD of proportions are provided

component	mu	sd
Temporal processes	0.302	0.154
Fixed effects	0.028	0.015
Spatial processes	0.004	0.033

Length-scale parameter of the periodic Gaussian processes indicating how quickly the correlation between two points decay as the distance (i.e., time) between them increases.

```
mean(GP_modu$gamma_MODU); sd(GP_modu$gamma_MODU)
```

```
[1] 2.816809
```

```
[1] 1.664316
```

#### S4.9.9 $X \rightarrow H'_2$

```
knitr::kable(betas_H2_G_z[grep('H2$', betas_H2_G_z[[1]]), ],
              digits = 2,
              caption = "Factors affecting fruits' nutritional functional space. Average values, 95% CIs, and prob
```

Table S13: Factors affecting fruits' nutritional functional space. Average values, 95% CIs, and probabilities of directional effects for the relationships assessed in the generative model. An effect is marked as TRUE if  $P(\text{effect} > 0) > 0.7$  or  $P(\text{effect} < 0) > 0.7$  or.

Directional effect	Mean	0.025%	0.975%	$P(\text{effect} > 0)$	$P(\text{effect} < 0)$	effect
temp -> H2	-0.09	-0.39	0.20	0.27	0.73	TRUE
rain -> H2	0.21	-0.07	0.52	0.93	0.07	TRUE
fruitAB -> H2	0.01	-0.10	0.11	0.56	0.44	FALSE
fruitS -> H2	0.00	-0.10	0.10	0.49	0.51	FALSE
nut -> H2	0.01	-0.10	0.10	0.55	0.45	FALSE
mor -> H2	0.00	-0.10	0.10	0.46	0.54	FALSE
PLAN -> H2	0.02	-0.09	0.12	0.62	0.38	FALSE
BIRD -> H2	0.01	-0.10	0.10	0.56	0.44	FALSE

Relative importance of rainfall explaining network  $H'_2$ .

```
((abs(betas_H2_G[grep('H2$', betas_H2_G$var), ]$mu[2]) -
  abs(betas_H2_G[grep('H2$', betas_H2_G$var), ]$mu[1])) /
  abs(betas_H2_G[grep('H2$', betas_H2_G$var), ]$mu[1])) * 100
```

[1] 129.1684

#### S4.9.9.1 Rainfall $\rightarrow H'_2$

```
effect_rain_H2 <-
  causal_effects(
    pars = post_H2_G,
    alpha_1 = 'alpha_H2',
    f_1 = 'f_H2',
    beta_1 = 'beta_rain_H2',
    var = dat$z_rainfall,
    y_obs = d$H2,
    level_1 = T
  )
```

Percentage increase of intervening *Rainfall*: average value and 95% CI.

```
quantile(effect_rain_H2$relative_change,
         c(0.025, 0.5, 0.975))
```

2.5%	50%	97.5%
-14.26338	42.67772	90.14868

Probability of the causal effect being positive:

```
unique(effect_rain_H2$p_effect)
```

[1] 0.9315342

#### S4.9.9.2 Temperature $\rightarrow H'_2$

```
effect_temp_H2 <-
  causal_effects(
    pars = post_H2_G,
    alpha_1 = 'alpha_H2',
    f_1 = 'f_H2',
    beta_1 = 'beta_temp_H2',
    var = dat$z_temperature,
    y_obs = d$H2,
    level_1 = T
  )
```

Percentage increase of intervening *temperature*: average value and 95% CI.

```
quantile(effect_temp_H2$relative_change,
         c(0.025, 0.5, 0.975))
```

```
2.5%      50%     97.5%
-73.55650 -18.02079 39.73685
```

Probability of the causal effect being positive:

```
unique(effect_temp_H2$p_effect)
```

```
[1] 0.2663668
```

#### S4.9.9.3 Plot direct effect

```
causal_plot_H2 <-
  rbind(effect_rain_H2,
        effect_temp_H2) |>
  ggplot(aes(x = relative_change, y = effect, fill = after_stat(x))) +
  geom_density_ridges(scale = 1, rel_min_height = 0.001,
                      alpha = 0.5,
                      color = 'lightblue3',
                      fill = 'lightblue',
                      linewidth = 0.25) +
  # geom_density_ridges_gradient(scale = 1.15, rel_min_height = 0.01,
  #                             linewidth = 0.1) +
  # scale_fill_viridis_c(name = "Direction", option = "C") +
  # scale_y_discrete(labels = c(expression('R' %-%>% "H2'"),
  #                            expression('T' %-%>% "H2')))) +
  #lims(y = c(-100, 224)) +
  labs(x = 'Relative causal effect (%)', x = '') +
  theme_classic() +
  theme(panel.grid = element_blank(),
        strip.background = element_blank(),
        axis.line = element_line(linewidth = 0.25),
        axis.ticks = element_line(linewidth = 0.25),
        legend.position = 'none',
        legend.position = c(0.925, 0.8),
        legend.background = element_blank(),
        legend.key.size = unit(0.25, 'cm'),
        legend.text = element_text(size = 5),
        legend.title = element_text(size = 6),
        axis.title.y = element_blank(),
        axis.text = element_text(size = 8.5))
```

```
causal_plot_H2
```

Picking joint bandwidth of 5.52

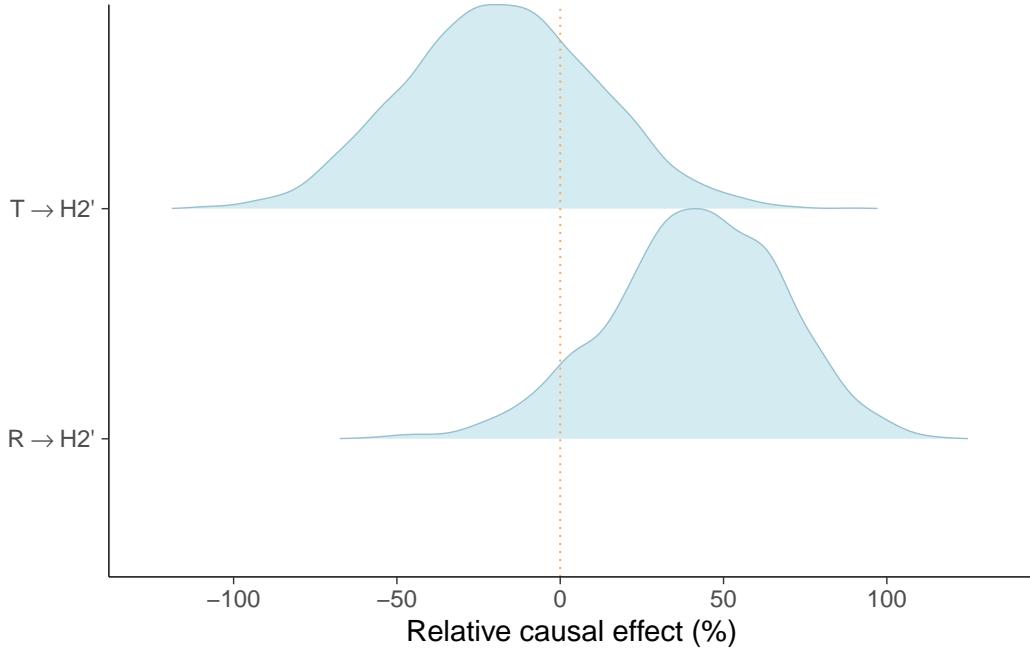


Figure S29: ausal direct effects of exposure variables on network specialization

#### S4.9.9.4 Variance decomposition

Procedures for conducting variance decomposition analysis can be consulted at the `generative_quantity` box in the `Stan` model. Here we extract the posterior distribution of the explained variance:

```
GP_modu_H2 <- mod_H2_G$draws(c('gamma_H2', 'prop_GP_temp',
                                'prop_fixed', 'prop_GP_space'), format = 'df')

GP_H2 <- do.call('rbind',
                  lapply(2:4, FUN =
                        function(x) {
                          tibble(component = c('Temporal processes',
                                              'Fixed effects',
                                              'Spatial processes')[x-1],
                          mu = median(GP_modu_H2[[x]]),
                          sd = sd(GP_modu_H2[[x]]))
}))
```

```
knitr::kable(GP_H2,
             digits = 3,
             caption = 'Variance decomposition of the components explaining intra-annual pattern of specialization')
```

Table S14: Variance decomposition of the components explaining intra-annual pattern of specialization in seed dispersal networks. Average and SD of proportions are provided

component	mu	sd
Temporal processes	0.008	0.045
Fixed effects	0.026	0.019
Spatial processes	0.006	0.044

Length-scale parameter of the periodic Gaussian processes indicating how quickly the correlation between two points decay as the distance (i.e., time) between them increases.

```
mean(GP_modu_H2$gamma_H2); sd(GP_modu_H2$gamma_H2)
```

```
[1] 1.22192
```

```
[1] 0.7134558
```

## S5 Computational environment

```
sessionInfo()
```

```
R version 4.5.1 (2025-06-13)
Platform: aarch64-apple-darwin20
Running under: macOS Sequoia 15.5
```

```
Matrix products: default
BLAS: /Library/Frameworks/R.framework/Versions/4.5-arm64/Resources/lib/libRblas.0.dylib
LAPACK: /Library/Frameworks/R.framework/Versions/4.5-arm64/Resources/lib/libRlapack.dylib

locale:
[1] en_US.UTF-8/en_US.UTF-8/en_US.UTF-8/C/en_US.UTF-8/en_US.UTF-8

time zone: America/Sao_Paulo
tzcode source: internal

attached base packages:
[1] parallel stats      graphics grDevices utils      datasets methods
[8] base
```

other attached packages:

```
[1] ggridges_0.5.6      ggsankey_0.0.99999  gtools_3.9.5
[4] tidyverse_1.3.1     ggfortify_0.4.18   patchwork_1.3.1
[7] readxl_1.4.5       hypervolume_3.1.6   Rcpp_1.1.0
[10] iNEXT_3.0.1       igraph_2.1.4        bipartite_2.21
[13] vegan_2.7-1       permute_0.9-8      sna_2.8
[16] network_1.19.0    statnet.common_4.12.0 cowplot_1.2.0
[19] rethinking_2.42    posterior_1.6.1    cmdstanr_0.9.0.9000
[22] magrittr_2.0.3   forcats_1.0.0      lubridate_1.9.4
[25] ggplot2_3.5.2     dplyr_1.1.4
```

loaded via a namespace (and not attached):

```
[1] RColorBrewer_1.1-3  tensorA_0.36.2.1   rstudioapi_0.17.1
[4] jsonlite_2.0.0      shape_1.4.6.1     farver_2.1.2
[7] rmarkdown_2.29       fields_16.3.1    vctrs_0.6.5
[10] terra_1.8-54       htmlltools_0.5.8.1 progress_1.2.3
[13] distributional_0.5.0 cellranger_1.1.0 pROC_1.18.5
[16] caret_7.0-1        parallelly_1.45.0 pracma_2.4.4
[19] KernSmooth_2.23-26 plyr_1.8.9      palmerpenguins_0.1.1
[22] lifecycle_1.0.4    iterators_1.0.14 pkgconfig_2.0.3
[25] Matrix_1.7-3      R6_2.6.1       fastmap_1.2.0
[28] future_1.58.0     magic_1.6-1     digest_0.6.37
[31] ps_1.9.1          pdist_1.2.1    labeling_0.4.3
[34] timechange_0.3.0   abind_1.4-8     mgcv_1.9-3
[37] compiler_4.5.1    proxy_0.4-27   withr_3.0.2
[40] doParallel_1.0.17  backports_1.5.0 Rttf2pt1_1.3.12
[43] maps_3.4.3        MASS_7.3-65    lava_1.8.1
[46] loo_2.8.0          ModelMetrics_1.2.2.2 tools_4.5.1
[49] extrafontdb_1.0    future.apply_1.20.0 nnet_7.3-20
[52] glue_1.8.0         nlme_3.1-168   grid_4.5.1
[55] checkmate_2.3.2   cluster_2.1.8.1 reshape2_1.4.4
[58] generics_0.1.4    recipes_1.3.1   gtable_0.3.6
[61] class_7.3-23      data.table_1.17.8 hms_1.1.3
[64] utf8_1.2.6         foreach_1.5.2   pillar_1.11.0
[67] stringr_1.5.1     spam_2.11-1    splines_4.5.1
[70] lattice_0.22-7    survival_3.8-3 ks_1.15.1
[73] tidyselect_1.2.1   knitr_1.50    gridExtra_2.3
[76] stats4_4.5.1      xfun_0.52    hardhat_1.4.1
[79] timeDate_4041.110 matrixStats_1.5.0 stringi_1.8.7
[82] yaml_2.3.10        evaluate_1.0.4 codetools_0.2-20
[85] extrafont_0.19     tibble_3.3.0   cli_3.6.5
[88] rpart_4.1.24       geometry_0.5.2 processx_3.8.6
```

```
[91] globals_0.18.0      coda_0.19-4.1      fastcluster_1.3.0
[94] gower_1.0.2        prettyunits_1.2.0    dotCall64_1.2
[97] mclust_6.1.1       listenv_0.9.1      viridisLite_0.4.2
[100] mvtnorm_1.3-3     ipred_0.9-15      scales_1.4.0
[103] prodlim_2025.04.28 e1071_1.7-16    purrr_1.0.4
[106] crayon_1.5.3      rlang_1.1.6
```