PNAS
www.pnas.org

Supplementary Information for

Network motifs involving both competition and facilitation predict biodiversity in alpine plant communities

Gianalberto Losapio, Christian Schöb, Phillip P.A. Staniczenko, Francesco Carrara, Gian Marco Palamara, Consuelo M. De Moraes, Mark C. Mescher, Rob W. Brooker, Bradley J. Butterfield, Ragan M. Callaway, Lohengrin A. Cavieres, Zaal Kikvidze, Christopher J. Lortie, Richard Michalet, Francisco I. Pugnaire, Jordi Bascompte

Correspondence to: Gianalberto Losapio losapiog@stanford.edu and Christian Schöb christian.schoeb@usys.ethz.ch

This file includes:

Figures S1 to S6
Tables S1 to S5

Other Supplementary materials for this manuscript include the following:

Data and Code

Figures

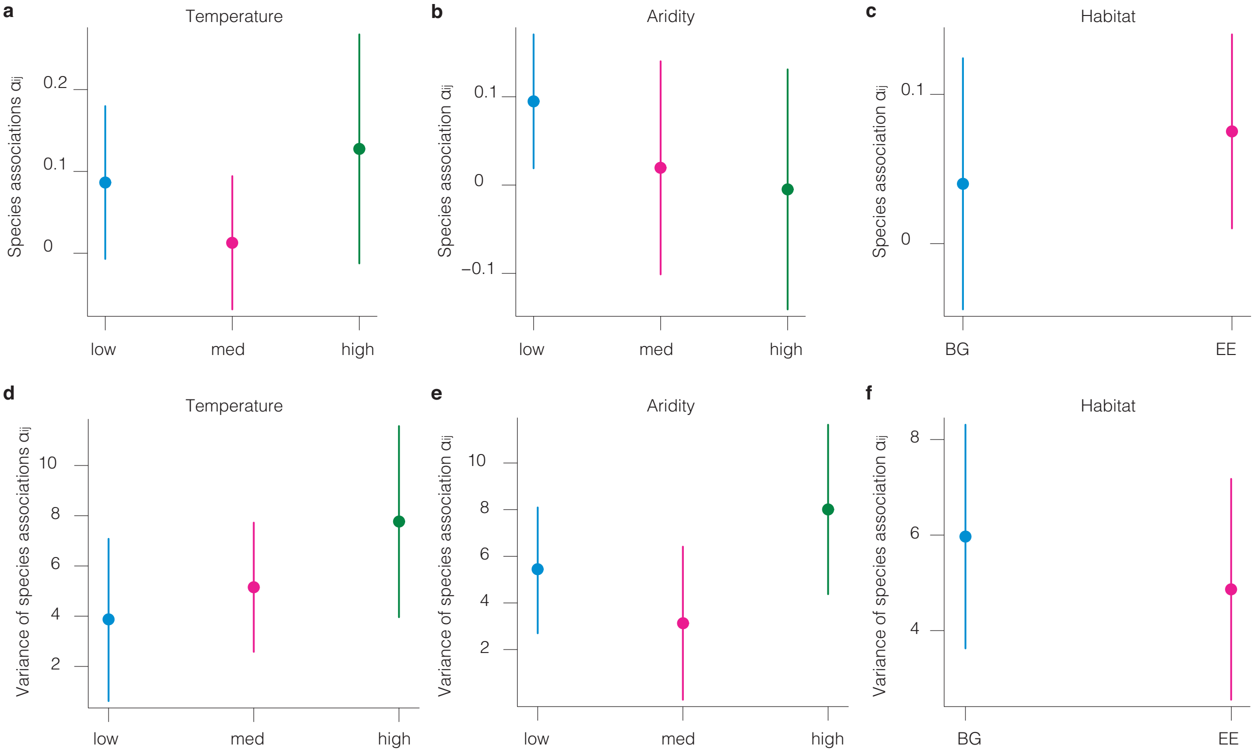


Fig. S1 Distribution of species associations α_{ij} across environmental conditions. **a-c**, Mean of α_{ij}. **d-f**, Variance of α_{ij}. We show fitted mean and 95% confidence interval conditional to each predictor (temperature, aridity, and habitat) and marginalized over the others. Results (Extended Data Table 1) reveal that species associations are consistent over environmental conditions across alpine communities.

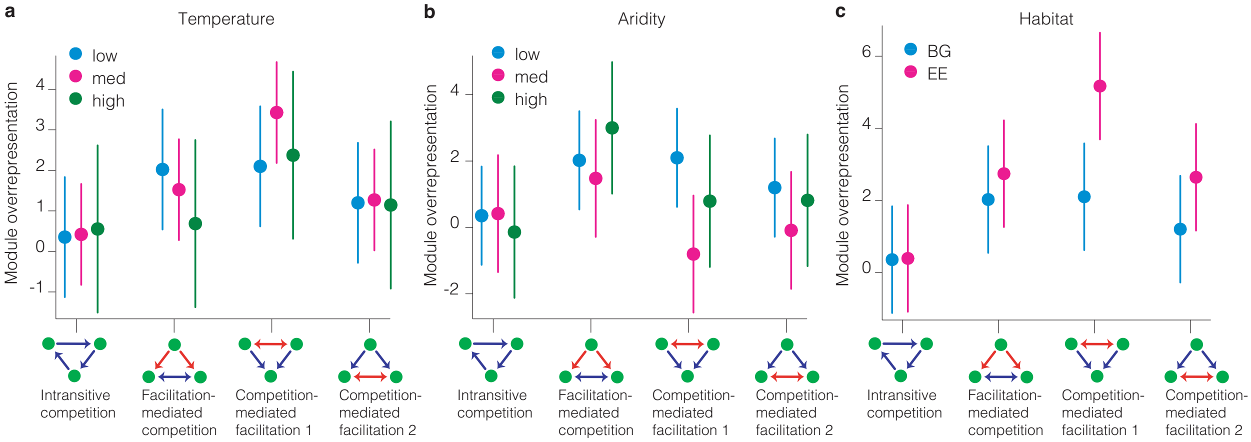


Fig. S2 Distribution of module overrepresentation across environmental conditions. Shown are mean and 95% confidence interval of fitted response conditional to temperature (**a**), aridity (**b**) and habitat ('BG' = bare ground; 'EE' = ecosystem engineer) (**c**) and marginalized over all other predictors. Results (Tab. S2) indicate that the overrepresentation of network modules can change with environmental conditions. In particular, competition-mediated facilitation type-2 decreases with aridity and increases in the presence of ecosystem engineers.

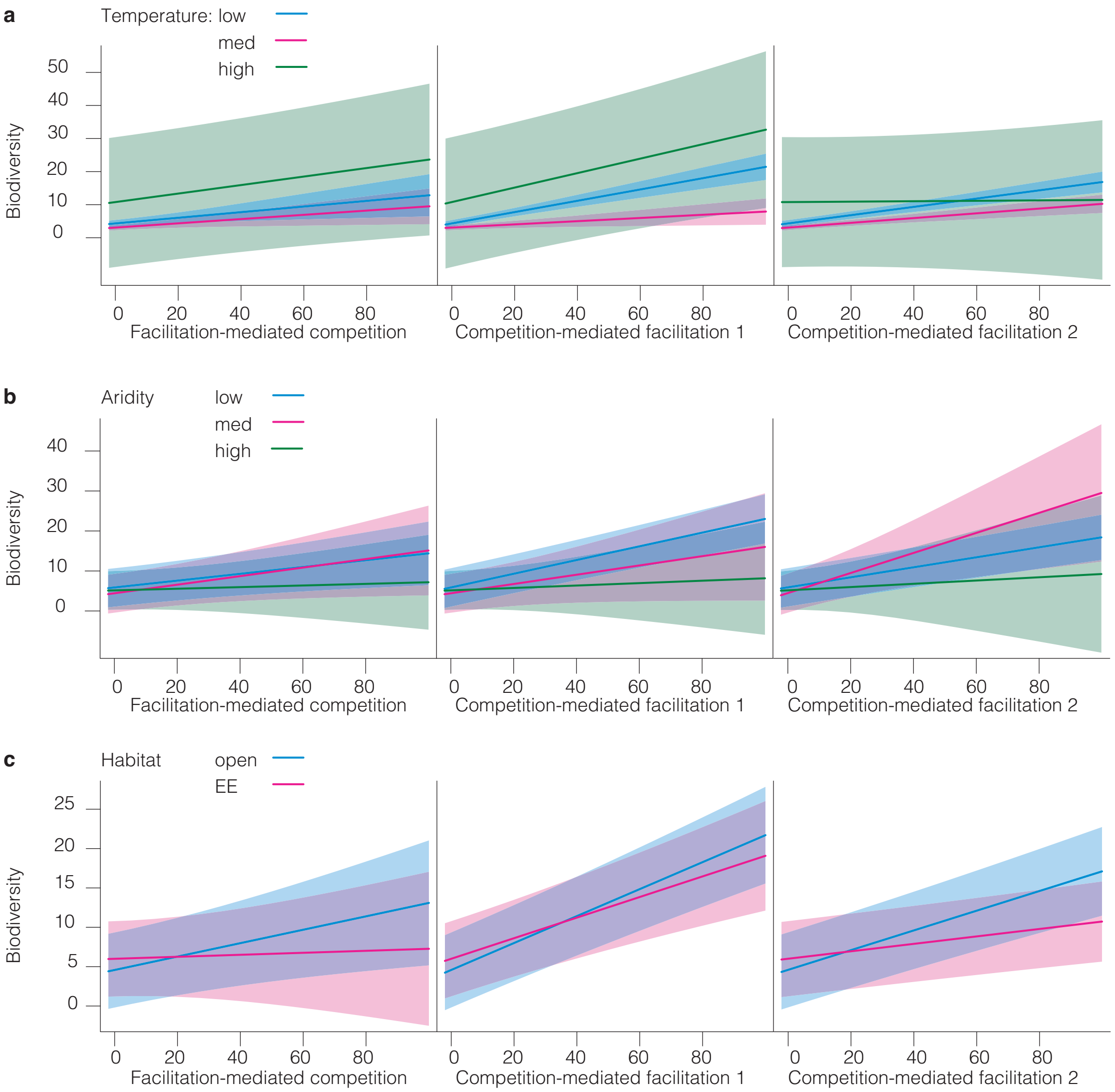


Fig. S3 Network motifs and biodiversity across environmental conditions in alpine plant communities worldwide. Relationship between biodiversity (*y*-axis) and overrepresentation of network motifs (*z*-score, *x*-axis) across different conditions of temperature (**a**), aridity (**b**), and habitat (**c**) ('BG' = bare ground; 'EE' = ecosystem engineers). Figures show fitted trends and 95% CI. For clarity, we plot only motifs of facilitation-mediated competition, competition-mediated facilitation type-1 and type-2, while the module of intransitive competition was dropped from the plotted model (Tab. S3).

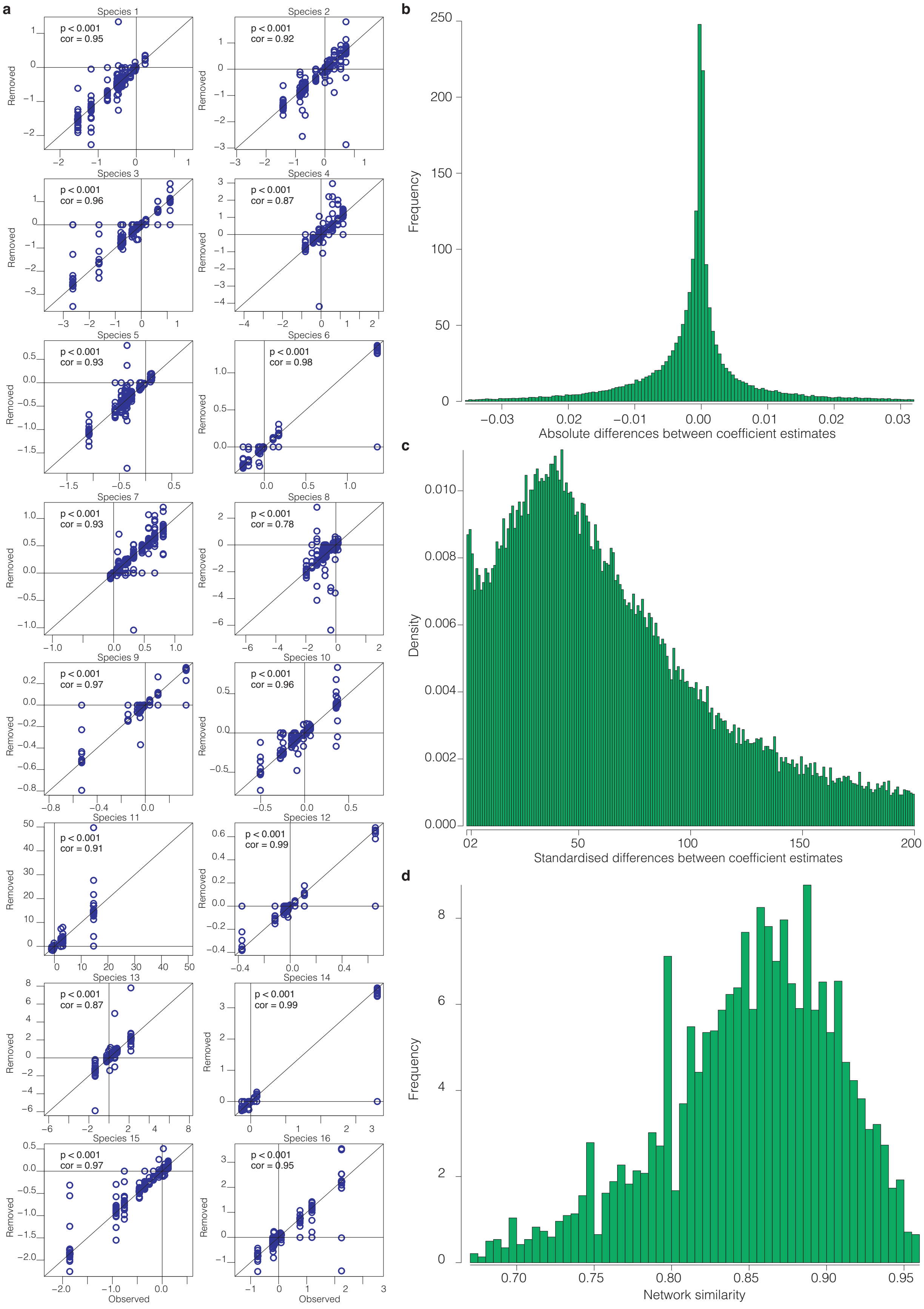

a
Species 1
p < 0.001
cor = 0.95
Species 2
p < 0.001
cor = 0.92
Species 3
p < 0.001
cor = 0.96
Species 4
p < 0.001
cor = 0.87
Species 5
p < 0.001
cor = 0.93
Species 6
p < 0.001
cor = 0.98
Species 7
p < 0.001
cor = 0.93
Species 8
p < 0.001
cor = 0.78
Species 9
p < 0.001
cor = 0.97
Species 10
p < 0.001
cor = 0.96
Species 11
p < 0.001
cor = 0.91
Species 12
p < 0.001
cor = 0.99
Species 13
p < 0.001
cor = 0.87
Species 14
p < 0.001
cor = 0.99
Species 15
p < 0.001
cor = 0.97
Species 16
p < 0.001
cor = 0.95
Removed
Observed
b
Frequency
0
50
100
150
200
250
−0.03
0.02
−0.01
0.00
0.01
0.02
0.03
Absolute differences between coefficient estimates
c
Density
0.000
0.002
0.004
0.006
0.008
0.010
02
50
100
150
200
Standardised differences between coefficient estimates
d
Frequency
0
2
4
6
8
0.70
0.75
0.80
0.85
0.90
0.95
Network similarity

Fig. S4 Robustness of Bayesian Poisson model to sample removal. a, Comparison of coefficients between observed data and data with a sample left out, for all samples for each focal species in a randomly-chosen community (ecosystem engineer habitat at Cathedral Lake 2, Canada). Within each species plot, each dot is a coefficient α_{ij} in the complete, observed dataset (*x*-axis) and in the dataset with individual samples removed (*y*-axis); results of Pearson correlation tests are reported at top-left. **b**, Distribution (*y*-axis) of absolute differences between coefficients in full and partial datasets (*x*-axis) across all communities and sites. For visualization
purposes we show the values within the 95% CI. **c**, Distribution (y-axis) of standardised coefficient differences between full and partial datasets (*x*-axis) across all communities and sites. The higher the value, the smaller the difference. The distribution is significantly higher than the critical value of two standard deviations. The plotted distribution is truncated to 200 for clarity. **d**, Distribution (*y*-axis) of similarity values between observed and partial networks (*x*-axis)
across all communities and sites. For visualization purposes, we show the values within the 95% CI.

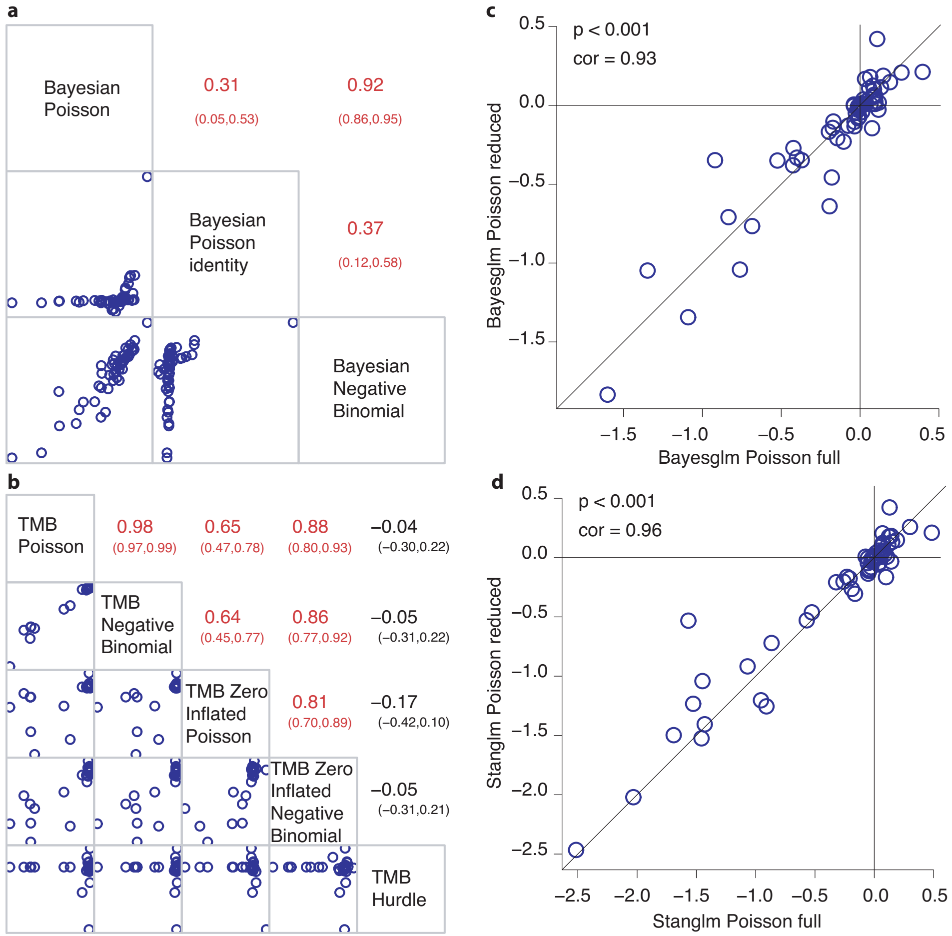


Fig. S5 Robustness of Bayesian Poisson model to changing distribution, model structure, and removing rare species. a-b, Correlation matrix of coefficient estimates (panels in the lower triangle) across different regression models (main diagonal) in a randomly-chosen community (ecosystem engineer habitat at Cathedral Lake 2, Canada). In the panels in the upper triangle, we report the Pearson correlation between estimates with 95% CI in parenthesis of the different models; in red the significant correlations. **c-d**, Correlation of species coefficients estimated with the full data (x-axis) and those estimated with partial data without rare species (y-axis) for two Bayesian Poisson models (*bayesglm* of *arm* and *stan_glm* of *rstanarm* in panels c and d, respectively) in a randomly-chosen community (ecosystem engineer habitat at Cathedral Lake 2, Canada). Results of Pearson correlation tests are reported at the top-left of each plot.

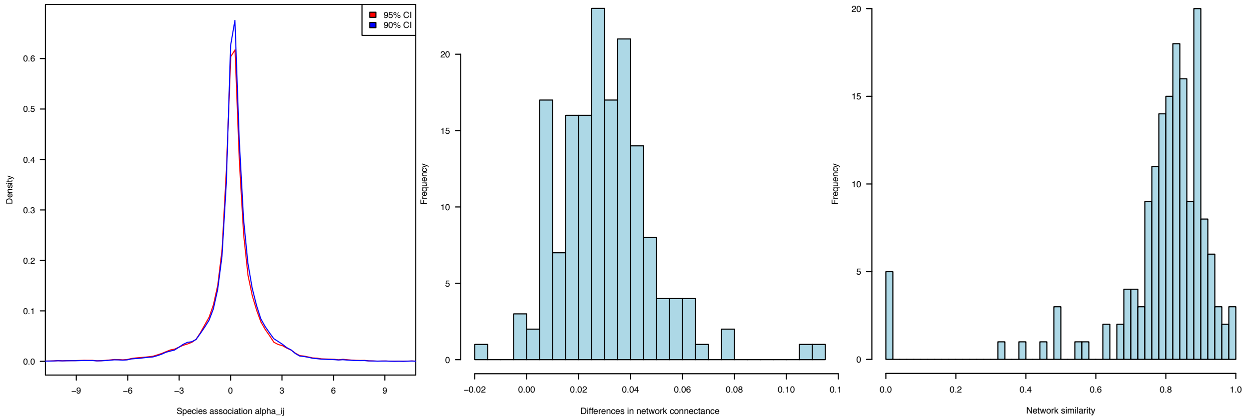
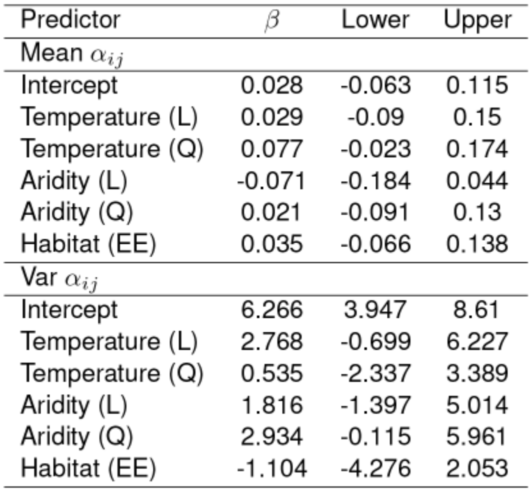


Fig. S6 Robustness of associations networks to changing the CI cutoff from 95% to 90%. Distribution of species associations α_{ij} (left), differences in network connectance (middle), and network similarity in terms of species-specific associations (right). The distribution of inferred species interactions α_{ij} does not change with changing the CI cutoff. Similarly, the differences in network connectance are negligible, indicating that the overall network does not change with changing the CI cutoff. Finally, the 95% CI and 90% CI networks are also similar in terms of specific association patterns, indicating that associations between species pairs do not change either.

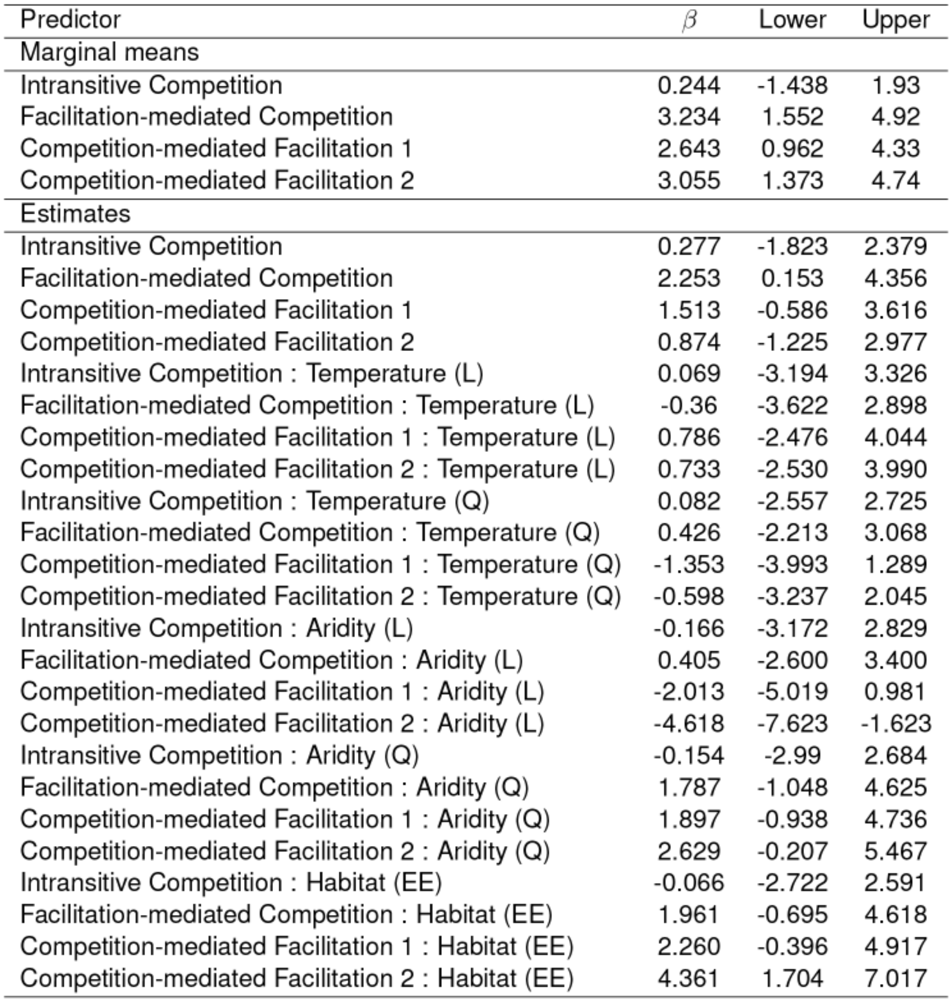
Tables

Tab. S1 Summary of mixed-effects models of species associations α_{ij}. We modelled the response of mean and variance of species associations to environmental conditions, including temperature (linear and quadratic), aridity (linear and quadratic), and habitat (ecosystem engineering and bare ground). Estimates (β), lower and upper 95% confidence intervals are reported. For graphical results, see Fig. S1.



Predictor	β	Lower	Upper
Mean α_{ij}			
Intercept	0.028	-0.063	0.115
Temperature (L)	0.029	-0.09	0.15
Temperature (Q)	0.077	-0.023	0.174
Aridity (L)	-0.071	-0.184	0.044
Aridity (Q)	0.021	-0.091	0.13
Habitat (EE)	0.035	-0.066	0.138
Var α_{ij}			
Intercept	6.266	3.947	8.61
Temperature (L)	2.768	-0.699	6.227
Temperature (Q)	0.535	-2.337	3.389
Aridity (L)	1.816	-1.397	5.014
Aridity (Q)	2.934	-0.115	5.961
Habitat (EE)	-1.104	-4.276	2.053

Tab. S2 Summary of marginal mean contrasts and mixed-effects models of network modules. First, we compared the overrepresentation of network modules (intransitive competition, facilitation-mediated competition, competition-mediated facilitation type-1 and type-2) among each other. Second, we modelled the overrepresentation of network modules across a range of environmental conditions, including temperature (linear and quadratic), aridity (linear and quadratic), and habitat (ecosystem engineering and bare ground). Estimates (β), lower and upper 95% confidence intervals are reported. For graphical results, see Fig. S2.

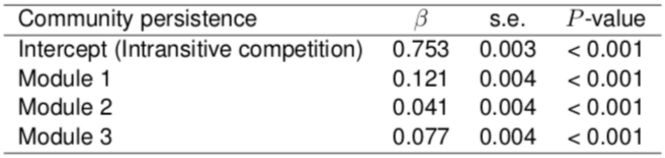


Predictor	β	Lower	Upper
Marginal means			
Intransitive Competition	0.244	-1.438	1.93
Facilitation-mediated Competition	3.234	1.552	4.92
Competition-mediated Facilitation 1	2.643	0.962	4.33
Competition-mediated Facilitation 2	3.055	1.373	4.74
Estimates			
Intransitive Competition	0.277	-1.823	2.379
Facilitation-mediated Competition	2.253	0.153	4.356
Competition-mediated Facilitation 1	1.513	-0.586	3.616
Competition-mediated Facilitation 2	0.874	-1.225	2.977
Intransitive Competition : Temperature (L)	0.069	-3.194	3.326
Facilitation-mediated Competition : Temperature (L)	-0.36	-3.622	2.898
Competition-mediated Facilitation 1 : Temperature (L)	0.786	-2.476	4.044
Competition-mediated Facilitation 2 : Temperature (L)	0.733	-2.530	3.990
Intransitive Competition : Temperature (Q)	0.082	-2.557	2.725
Facilitation-mediated Competition : Temperature (Q)	0.426	-2.213	3.068
Competition-mediated Facilitation 1 : Temperature (Q)	-1.353	-3.993	1.289
Competition-mediated Facilitation 2 : Temperature (Q)	-0.598	-3.237	2.045
Intransitive Competition : Aridity (L)	-0.166	-3.172	2.829
Facilitation-mediated Competition : Aridity (L)	0.405	-2.600	3.400
Competition-mediated Facilitation 1 : Aridity (L)	-2.013	-5.019	0.981
Competition-mediated Facilitation 2 : Aridity (L)	-4.618	-7.623	-1.623
Intransitive Competition : Aridity (Q)	-0.154	-2.99	2.684
Facilitation-mediated Competition : Aridity (Q)	1.787	-1.048	4.625
Competition-mediated Facilitation 1 : Aridity (Q)	1.897	-0.938	4.736
Competition-mediated Facilitation 2 : Aridity (Q)	2.629	-0.207	5.467
Intransitive Competition : Habitat (EE)	-0.066	-2.722	2.591
Facilitation-mediated Competition : Habitat (EE)	1.961	-0.695	4.618
Competition-mediated Facilitation 1 : Habitat (EE)	2.260	-0.396	4.917
Competition-mediated Facilitation 2 : Habitat (EE)	4.361	1.704	7.017

Tab. S3 Summary of mixed-effects model of biodiversity. We modelled the response of biodiversity (average number of plant species per plot across each community) to overrepresentation of network modules (intransitive competition, facilitation-mediated competition, competition-mediated facilitation type-1 and type-2) after accounting for environmental conditions, including temperature (linear and quadratic), aridity (linear and quadratic), and habitat (ecosystem engineering and bare ground), and other general network descriptors (median of species associations α_{ij}, connectance, and ratio positive/negative associations). Estimates (β), lower and upper 95% confidence intervals are reported. For graphical results, see Fig. S3.

Predictor	β	Lower	Upper
Intercept	3.047	2.455	3.642
Temperature (L)	0.182	-0.967	1.352
Temperature (Q)	1.224	0.328	2.100
Aridity (L)	-0.920	-1.821	0.001
Aridity (Q)	0.749	0.075	1.442
Habitat (EE)	1.391	1.165	1.619
Median α_{ij}	0.138	-0.153	0.432
Connectance	0.440	0.163	0.715
Ratio positive/negative	0.034	-0.050	0.117
Intransitive Competition	-1.020	-2.556	0.514
Facilitation-mediated Competition	0.075	0.016	0.134
Competition-mediated Facilitation 1	0.079	0.027	0.130
Competition-mediated Facilitation 2	0.083	0.016	0.150
Temperature (L) : Median α_{ij}	-1.491	-2.202	-0.782
Temperature (Q) : Median α_{ij}	-1.135	-1.624	-0.645
Temperature (L) : Connectance	-0.339	-0.905	0.224
Temperature (Q) : Connectance	-0.217	-0.643	0.209
Temperature (L) : Ratio positive/negative	0.073	-0.104	0.250
Temperature (Q) : Ratio positive/negative	0.041	-0.061	0.143
Aridity (L) : Median α_{ij}	1.650	1.046	2.256
Aridity (Q) : Median α_{ij}	0.975	0.515	1.436
Aridity (L) : Connectance	-0.444	-1.000	0.114
Aridity (Q) : Connectance	0.557	0.155	0.960
Aridity (L) : Ratio positive/negative	-0.001	-0.002	0.000
Aridity (Q) : Ratio positive/negative	0.001	0.000	0.002
Habitat (EE) : Median α_{ij}	-0.428	-0.942	0.085
Habitat (EE) : Connectance	-0.117	-0.387	0.152
Habitat (EE) : Ratio positive/negative	0.001	0.000	0.002
Temperature (L) : Intransitive Competition	-0.054	-0.316	0.208
Temperature (Q) : Intransitive Competition	-0.222	-1.123	0.682
Temperature (L) : Facilitation-mediated Competition	0.027	-0.046	0.101
Temperature (Q) : Facilitation-mediated Competition	0.018	-0.040	0.076
Temperature (L) : Competition-mediated Facilitation 1	0.023	-0.070	0.117
Temperature (Q) : Competition-mediated Facilitation 1	0.104	0.046	0.162
Temperature (L) : Competition-mediated Facilitation 2	-0.090	-0.195	0.015
Temperature (Q) : Competition-mediated Facilitation 2	-0.013	-0.075	0.049
Aridity (L) : Intransitive Competition	-2.199	-5.368	0.966
Aridity (Q) : Intransitive Competition	-1.348	-3.183	0.484
Aridity (L) : Facilitation-mediated Competition	-0.021	-0.090	0.048
Aridity (Q) : Facilitation-mediated Competition	-0.048	-0.124	0.029
Aridity (L) : Competition-mediated Facilitation 1	-0.075	-0.161	0.011
Aridity (Q) : Competition-mediated Facilitation 1	-0.011	-0.104	0.083
Aridity (L) : Competition-mediated Facilitation 2	-0.030	-0.154	0.094
Aridity (Q) : Competition-mediated Facilitation 2	-0.124	-0.250	0.001
Habitat (EE) : Intransitive Competition	-0.191	-1.285	0.904
Habitat (EE) : Facilitation-mediated Competition	-0.072	-0.139	-0.005
Habitat (EE) : Competition-mediated Facilitation 1	-0.037	-0.076	0.002
Habitat (EE) : Competition-mediated Facilitation 2	-0.071	-0.098	-0.044

Tab. S4 Effects of network modules on community persistence. We modelled the effects of network modules on species persistence in isolated modules using generalised additive models. Estimates (β), standard errors and *P*-values are reported.



Community persistence	β	s.e.	P-value
Intercept (Intransitive competition)	0.753	0.003	< 0.001
Module 1	0.121	0.004	< 0.001
Module 2	0.041	0.004	< 0.001
Module 3	0.077	0.004	< 0.001

Tab. S5 Estimate for biodiversity responses in two other null models. We randomized the community data using two additional null models (*c0_ind* and *vaznull* in ref *26*) and building resulting random networks. Then, we modelled the response of biodiversity (average number of plant species per plot across each community) to overrepresentation of network modules (intransitive competition, facilitation-mediated competition, competition-mediated facilitation type-1 and type-2) after accounting for environmental conditions, including temperature (linear and quadratic), aridity (linear and quadratic), and habitat (ecosystem engineering and bare ground), and other general network descriptors (median of species associations α_{ij}). Estimates (β), lower (Low) and upper (Up) 95% confidence intervals are reported for each null model.

Predictor	c_00 β	c_00 Low	c_00 Up	vzq $\hat{\beta}$	vzq Low	vzq Up
Intercept	2.967	2.407	3.530	2.868	2.315	3.419
Median α_{ij}	-0.193	-0.431	0.050	-0.226	-0.371	-0.079
Temperature (L)	-0.337	-1.443	0.766	0.207	-0.806	1.234
Temperature (Q)	1.150	0.273	2.030	0.538	-0.170	1.233
Aridity (L)	-0.320	-1.183	0.556	-0.218	-1.085	0.666
Aridity (Q)	1.053	0.405	1.708	1.027	0.391	1.678
Habitat (EE)	1.588	1.383	1.792	1.582	1.381	1.784
Intransitive Competition	-3.157	-8.893	2.581	-0.146	-3.252	2.959
Facilitation-mediated Competition	0.055	0.018	0.093	0.147	0.050	0.244
Competition-mediated Facilitation 1	-0.667	-0.913	-0.423	0.088	0.013	0.162
Competition-mediated Facilitation 2	0.103	-0.096	0.300	0.210	0.082	0.339
Temperature (L) : Intransitive Competition	-2.086	-6.015	1.839	-0.029	-0.324	0.264
Temperature (Q) : Intransitive Competition	1.627	-0.905	4.168	0.982	-0.436	2.411
Temperature (L) : Facilitation-mediated Competition	0.010	-0.032	0.051	-0.017	-0.148	0.113
Temperature (Q) : Facilitation-mediated Competition	0.018	-0.012	0.048	0.145	0.043	0.247
Temperature (L) : Competition-mediated Facilitation 1	0.080	-0.120	0.280	0.041	-0.063	0.145
Temperature (Q) : Competition-mediated Facilitation 1	0.051	-0.068	0.171	0.077	0.022	0.132
Temperature (L) : Competition-mediated Facilitation 2	0.033	-0.301	0.368	-0.157	-0.318	0.003
Temperature (Q) : Competition-mediated Facilitation 2	0.001	-0.193	0.195	0.102	0.002	0.203
Aridity (L) : Intransitive Competition	-4.274	-16.586	8.049	0.464	-5.782	6.715
Aridity (Q) : Intransitive Competition	-4.835	-11.937	2.269	0.285	-3.322	3.894
Aridity (L) : Facilitation-mediated Competition	-0.040	-0.073	-0.007	-0.212	-0.301	-0.123
Aridity (Q) : Facilitation-mediated Competition	-0.005	-0.047	0.037	0.077	-0.066	0.221
Aridity (L) : Competition-mediated Facilitation 1	-0.035	-0.154	0.084	-0.101	-0.168	-0.035
Aridity (Q) : Competition-mediated Facilitation 1	-0.027	-0.245	0.193	0.000	-0.110	0.111
Aridity (L) : Competition-mediated Facilitation 2	0.064	-0.385	0.511	-0.085	-0.196	0.026
Aridity (Q) : Competition-mediated Facilitation 2	-0.081	-0.358	0.196	-0.138	-0.310	0.033
Habitat (EE) : Intransitive Competition	0.798	-0.293	1.908	1.252	-0.470	2.987
Habitat (EE) : Facilitation-mediated Competition	-0.041	-0.082	0.001	-0.140	-0.261	-0.018
Habitat (EE) : Competition-mediated Facilitation 1	0.703	0.467	0.941	-0.079	-0.153	-0.004
Habitat (EE) : Competition-mediated Facilitation 2	0.012	-0.008	0.032	-0.148	-0.276	-0.019

Introduction

This Supplemetary Material reports the code to reproduce the analyses and figures carried out in Losapio *et al.* (2020). The scope of this tutorial is to increase its reproducibility, clarity, transparency and dissemination. In order to use this tutorial effectively an HPC cluster is reccommended. Data analysis was done using R version 3.6.0. This document was compiled with the 'rmarkdown package, version 1.13. This tutorial is licensed under CC BY-NC-ND 4.0, which means you are free to share, copy and redistribute this tutorial in any medium or format under the terms of attribution of appropriate credit, non-commercial purposes and no derivatives. The citation is:

> Losapio, G., Schöb, C., Staniczenko, P.P.A., Carrara, F., Palamara, G.M., De Moraes, C.M., Mescher, M.C., Brooker, R.W., Butterfield, B.J., Callaway, R.M., Cavieres, L.A., Kikvidze, Z., Lortie, C.J., Michalet, R., Pugnaire, F.I. & Bascompte, J. 2020. Network motifs involving both competition and facilitation predict biodiversity in alpine plant communities.

Software preparation

Install R (https://www.r-project.org) (R Core Team 2020) if you do not have it already. Then, install and load the following packages.

```r
library(reshape)
library(vegan)
library(igraph)
library(bipartite)
library(parallel)
library(foreach)
library(doParallel)
library(arm)
library(rstanarm)
library(car)
library(lme4)
library(lmerTest)
library(effects)
library(emmeans)
library(truncnorm)
library(lattice)
library(gplots)
library(ggplot2)
library(deSolve)
library(mgcv)
library(itsadug)
library(tidyr)
library(ggridges)
library(tidyverse)
```

```
library(pscl)
library(betalink)
library(glmmTMB)
library(corrgram)
library(bayesplot)
library(countreg)
```

Set your working directory and prepare for parallelisation.

```
setwd("/mydir")
numCores <- detectCores()
registerDoParallel(numCores)
```

Data import

Download data at the following link:

https://polybox.ethz.ch/index.php/s/ngHzOumWVsk5omz/download

Un-zip the folder in your home directory (`mydir`) and import those files.

```
datafiles             <-list.files(path="/datafiles", full.names=TRUE)
datafiles.list        <-lapply(datafiles,read.table)
names(datafiles.list) <-list.files(path="/datafiles", full.names=FALSE)
biodiv                <-read.table("biodiv.csv", sep=",", head=T)
envdata               <-read.table("envdata.csv", sep=",", head=T)

# create a dataframe where summary data will be kept
nsites   <- length(datafiles)
sites.df <- data.frame(label=rep(names(datafiles.list),2),
                    site=rep(c(100,101,11:19,1,23:26,2,31:34,36:39,3,
                               41:47,49,4, 50,52:54,56:59,5,60:65,67:81,
                               83:89,8,90:95,97:99,9),2),
                    # two habitats: cushion and open
                    habitat=rep(gl(2, nsites, nsites*2,
                                   label=c("cush","open"))))
sites.df$habitat <- relevel(sites.df$habitat, ref="open")
```

Data preparation

We will create a community matrix M per each plant community per habitat per site k in which plant species S are in columns, plots p are in rows, and species abundance N (i.e., number of individuals per species) are the entries, such as:

$$\mathbf{M_k} : \begin{bmatrix} N_{1,1} & \cdots & N_{1,s} \\ \vdots & \ddots & \vdots \\ N_{p,1} & \cdots & N_{p,s} \end{bmatrix} \tag{1}$$

```r
data.cush.list <- rep(list(NA),nsites) # communities in EcosystEng habitat
data.open.list <- rep(list(NA),nsites) # communities in Bareground habitat
for(zz in 1:nsites){
  mat <- datafiles.list[[zz]]
  mat1 <- mat[-1,-c(1,2)]
  for(i in 1:ncol(mat1)){mat1[,i] <- as.numeric(as.character(mat1[,i]))}
  # first two columns are useless
  mat2 = data.matrix(mat1)
  colnames(mat2) = as.character(unlist(mat[1,3:ncol(mat)]))
  rownames(mat2)<-NULL
  # drop absent species
  data.cush.list[[zz]] <- empty(subset(mat2, mat$V2[-1]=="cushion"))
  data.open.list[[zz]] <- empty(subset(mat2, mat$V2[-1]=="open"))
}
```

Data exploration

We check the number of species and their mean, variance and dispersion in each community.

```r
# number of species
sites.df$nsp=0
for(zz in 1:nsites){
  sites.df$nsp[zz]        <- ncol(data.cush.list[[zz]])
  sites.df$nsp[zz+nsites] <- ncol(data.open.list[[zz]])
}
# mean, variance and dispersion of each species
dispsp <- rep(rep(list(NA),nsites),2)
for(i in 1:nsites){
  # cushion
  dispsp[[i]]            <- matrix(NA,nrow=3,ncol=sites.df$nsp[i])
  dispsp[[i]][1,]        <- colSums(data.cush.list[[i]])/
    nrow(data.cush.list[[i]])
  dispsp[[i]][2,]        <- apply(data.cush.list[[i]],2,var)
  dispsp[[i]][3,]        <- dispsp[[i]][2,]/dispsp[[i]][1,]
  # open
  dispsp[[i+nsites]]     <- matrix(NA,nrow=3,ncol=sites.df$nsp[i+nsites])
  dispsp[[i+nsites]][1,] <- colSums(data.open.list[[i]])/
    nrow(data.open.list[[i]])
  dispsp[[i+nsites]][2,] <- apply(data.open.list[[i]],2,var)
```

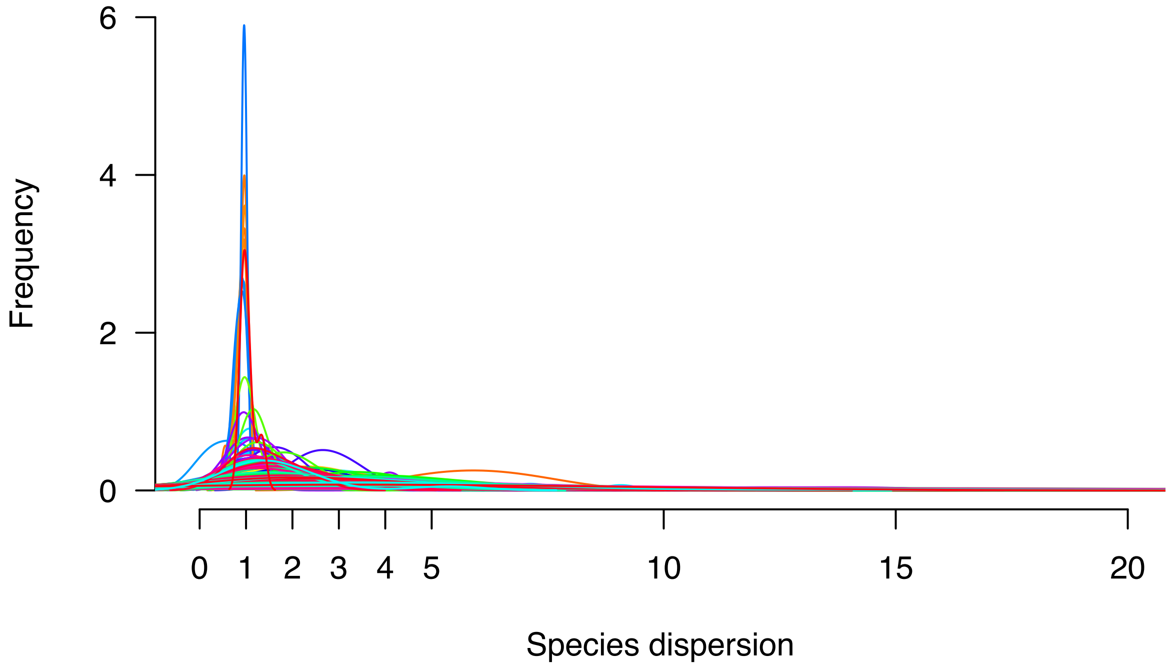
```
  dispsp[[i+nsites]][3,] <- dispsp[[i+nsites]][2,]/dispsp[[i+nsites]][1,]
}
alldisp <- NA
for(i in 1:nsites){
  alldisp <- c(alldisp,dispsp[[i]][3,])
  alldisp <- c(alldisp,dispsp[[i+nsites]][3,])
}
alldisp<-alldisp[-1]
round(quantile(alldisp),2)
```

```
##     0%    25%    50%    75%   100%
##   0.25   1.00   2.05   4.14 150.33
```

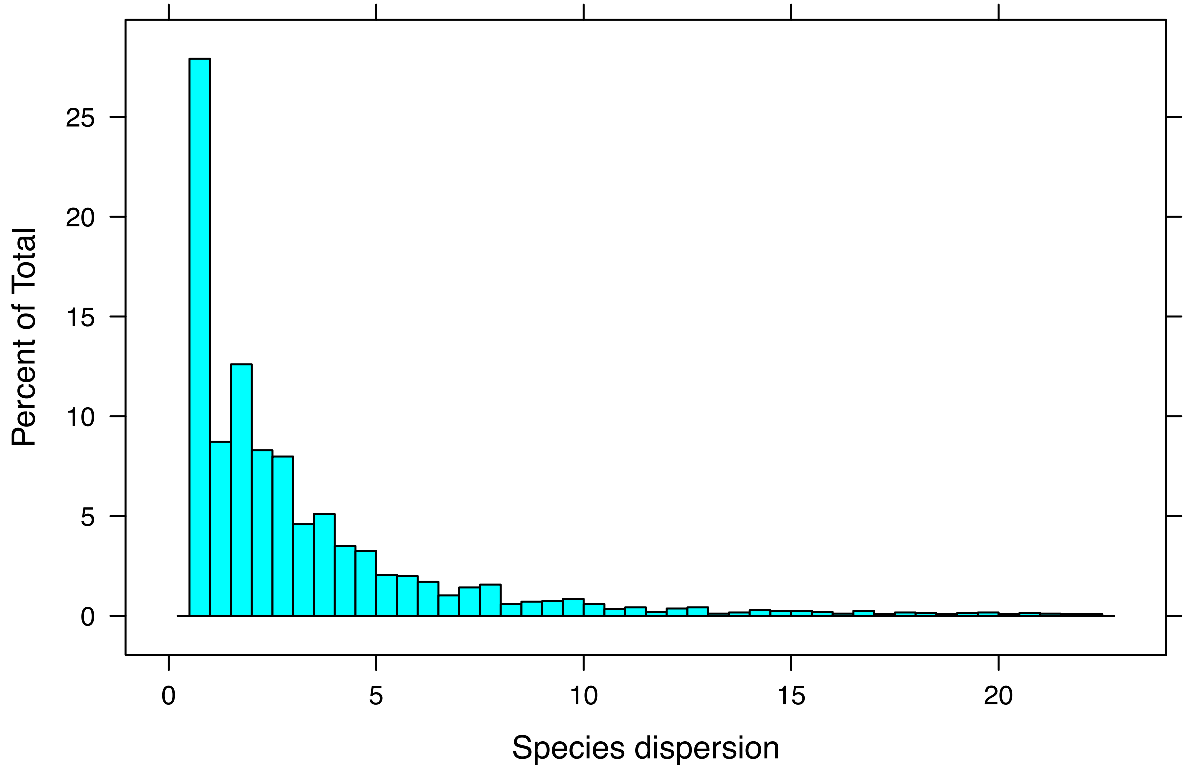
```
round(quantile(alldisp, c(0.025, 0.975)),2)
```

```
##  2.5% 97.5%
##  0.88 22.18
```

```
alldisp95<-alldisp[which(alldisp>0.88&alldisp<22.18)]

# we plot these results
plot(0,0,type="n",xlim=c(-0.15,20), ylim=c(0,6), xaxt="n",
     xlab="Species dispersion",ylab="Frequency",yaxt="n", axes=F)
  axis(1,at=c(0,1,2,3,4,5,10,15,20), las=1)
  axis(2,at=c(0,2,4,6), las=1)
for(i in 1:nsites){
  lines(density(dispsp[[i]][3,], adjust=2), col=colcurv[i])
  lines(density(dispsp[[i+nsites]][3,], adjust=2), col=colcurv[i+nsites])
}
```



```
histogram(alldisp95, breaks=50, xlab="Species dispersion")
```



These results indicate that there is a certain degree of variability in the dispersion of species abundances, ranging from underdispersed to overdispersed.

Ecological Networks

Empirical communities

Species associations

We estimate species associations among all plant species S occurring within the same community k. This is done following the model of Harris (Harris 2016), precisely the formula reported at lines 58-64 at the following link:

https://github.com/davharris/rosalia/blob/93b11593df9fc79752a516482ec3490d611375c0/Harris_2015/recovery.R

Differently to that model, we used a Poisson distribution given that our community matrices are based on species abundances. The Bayesian linear regression model looks like:

$$E(y_i) \propto \exp(b_i + \sum_{k=1}^{M} \sum_{j=1}^{S} \alpha_{ij} x_{jk}), \tag{2}$$

where b_i is the intercept term, x_{jk} is the abundance of each neighbouring species j in a sample k, and α_{ij} is the association between a target species i and a neighboring species j, summed over all species (S) and over all samples k in a community (M). Species associations α_{ij} are estimated using the `bayesglm` function of `arm` package. Only α_{ij} coefficients whose credible intervals do not include 0 were retained.

```r
statmod <- function(datai, dataj, i, nsp){
  mod <- bayesglm(datai~dataj, family="poisson")
  estimcoef <- coef(mod)[-1]
  posterior <- coef(sim(mod))[,-1]
  for(k in 1:(nsp-1)){
    if(sign(quantile(posterior[,k],0.025))!=
       sign(quantile(posterior[,k],0.975)))
      estimcoef[k] <- 0
  }
  cicoef     <- rep(0,nsp)
  cicoef[-i] <- estimcoef
  return(cicoef)
}

# This model formulation is equivalent to specyfing the formula as:
fmlamod <- as.formula(paste(paste(colnames(comdata)[j], "~"),
                            paste(spi, collapse="+")))
```

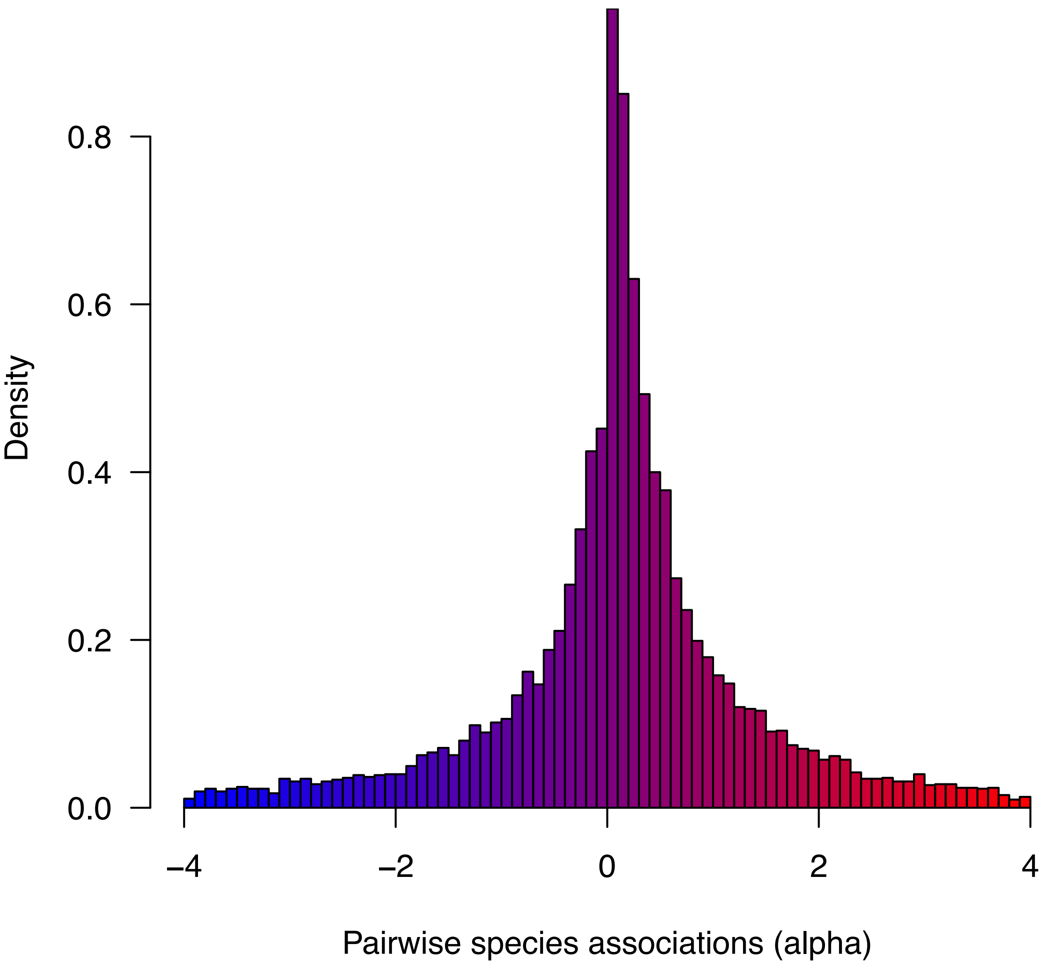
Those retained α_{ij} are then organised into a square, adjacency matrix $A_k = (S, \alpha_{ij})$ with size equals the number of species S per community k as

$$\mathbf{A_k} : \begin{bmatrix} \alpha_{1,1} & \cdots & \alpha_{1,j} \\ \vdots & \ddots & \vdots \\ \alpha_{i,1} & \cdots & \alpha_{i,j} \end{bmatrix} \tag{3}$$

```r
# adjacency matrices for cushion habitats
coef.cush_o <- foreach(zz=1:nsites, .inorder = TRUE) %dopar% {
  nsp <- ncol(data.cush.list[[zz]])
  if(nsp >=5){
  coef.site <- foreach(i=1:nsp, .combine=rbind, .inorder = TRUE) %dopar%{
    statmod(data.cush.list[[zz]][,i], data.cush.list[[zz]][,-i], i, nsp)
}}
  else NA
}
# adjacency matrices for open habitats
coef.open_o <- foreach(zz=1:nsites, .inorder = TRUE) %dopar% {
  nsp <- ncol(data.open.list[[zz]])
  if(nsp >=5){
  coef.site <- foreach(i=1:nsp, .combine=rbind, .inorder = TRUE) %dopar%{
    statmod(data.open.list[[zz]][,i], data.open.list[[zz]][,-i], i, nsp)
  }}
  else NA
}
```

We plot all the retained α_{ij} (within an interval of -4–4 for improving visualisation).

```r
aij_o_cut <- unlist(aij_o)
aij_o_cut <- aij_o_cut[which(aij_o_cut<4&aij_o_cut>-4)]
colfunc<-colorRampPalette(c("blue","red"))

par(yaxs="i",las=1)
hist(aij_o_cut, breaks = seq(from=-4,to=4,length.out=81),
     freq=FALSE, col=colfunc(81),
     main="", xlab= "Pairwise species associations (alpha)")
```



Network modules

Each adjacency matrix A_k represents a plant–plant network of a plant community per habitat per site. We will proceed with analysing the frequency of three-species associations in each observed plant–plant networks.

The first three-species associations is intransitive competition, which correspond to the following module:

$$\begin{bmatrix} 0 & -1 & 0 \\ 0 & 0 & -1 \\ -1 & 0 & 0 \end{bmatrix} \tag{4}$$

To measure the frequency of intransitive competition modules we use the function `triad_census` of the `igraph` package and consider its 10th element (a –> -b, b –> -c, c –> -a). We search through a network of only negative links.

```
# intransitive competition
intcomp<-function(mat){
  mat[which(mat>0)]<-0
  gnet<-graph.adjacency(t(mat), weighted=TRUE)
  triads<-triad_census(gnet)[10]
```

```
  return(triads)
}

# networks with only negative links
gnet.neg_o <- rep(rep(list(NA),nsites),2)
names(gnet.neg_o) <- c(names(coef.cush_o),names(coef.open_o))

for(i in 1:nsites){
  if(sites.df$nsp[i]>=5){
    # cushion
    mat <- coef.cush_o[[i]]
    mat[which(mat>0)] <- 0
    gnet.neg_o[[i]] <- graph.adjacency(t(mat), weighted=TRUE)
    # open
    mat <- coef.open_o[[i]]
    mat[which(mat>0)] <- 0
    gnet.neg_o[[i+nsites]] <- graph.adjacency(t(mat), weighted=TRUE)
}}
```

To measure the frequency of facilitation-mediated competition (a –> b, a –> c, -b <–> -c) we write an algorightm that search through the adjacency matrix the frequency of the module

$$\begin{bmatrix} 0 & 0 & 0 \\ 1 & 0 & -1 \\ 1 & -1 & 0 \end{bmatrix} \tag{5}$$

```
# facilitation-mediated competition
facmcom<-function(mat){
  nsp<-nrow(mat)
  mat[which(is.na(mat))]<-0
  matp<- mat; matp[which(matp<0)]<-0; matp[which(matp>0)]<-1
  matn<- mat; matn[which(matn>0)]<-0; matn[which(matn<0)]<-1
  #
  ntrip<-0
  for(i in 1:nsp){
    # first, subset two species facilitated by a third
    nei<-sum(matp[,i])
    idnei<-as.numeric(which(matp[,i]==1))
      if(nei>=2){
      # number of search = number of possible pairs = n!/2!(n-2)!
      nos<- factorial(nei)/(2*factorial(nei-2))
      counter<-0
      # then, for each facilitated pair,
      # look if they are both negatively associated
      for(k in 1:(nei-1)){for(z in (k+1):nei){
```

```
        counter<-counter+1
        if(matn[idnei[k],idnei[z]]==1&matn[idnei[z],idnei[k]]==1)
          ntrip <- ntrip+1
      }
        if(counter==nos) break}
    }
  }
  return(ntrip)
}
```

To measure the frequency of competition-mediated facilitation 1 (b –> -a, c –> -a, b <–> c) we write an algorightm that search through the adjacency matrix the frequency of the module

$$\begin{bmatrix} 0 & -1 & -1 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix} \tag{6}$$

```
# competition-mediated facilitation type 1
commfac1<-function(mat){
  nsp<-nrow(mat)
  mat[which(is.na(mat))]<-0
  matp<- mat; matp[which(matp<0)]<-0; matp[which(matp>0)]<-1
  matn<- mat; matn[which(matn>0)]<-0; matn[which(matn<0)]<-1
  #
  ntrip<-0
  for(i in 1:nsp){
    # first, subset two species competing with a third
    nei<-sum(matn[i,])
    idnei<-as.numeric(which(matn[i,]==1))
      if(nei>=2){
      # number of search = number of possible pairs = n!/2!(n-2)!
      nos<- factorial(nei)/(2*factorial(nei-2))
      counter<-0
      # then, for each competing pair,
      # look if they are both positively associated
      for(k in 1:(nei-1)){for(z in (k+1):nei){
        counter<-counter+1
        if(matp[idnei[k],idnei[z]]==1&matp[idnei[z],idnei[k]]==1)
          ntrip <- ntrip+1
      }
        if(counter==nos) break}
    }
  }
  return(ntrip)
}
```

To measure the frequency of competition-mediated facilitation 2 (a –> -b, a –> -c, b<–>c) we write an algorightm that search through the adjacency matrix the frequency of the module

$$\begin{bmatrix} 0 & 0 & 0 \\ -1 & 0 & 1 \\ -1 & 1 & 0 \end{bmatrix} \tag{7}$$

```r
# competition-mediated facilitation type 2
commfac2<-function(mat){
  nsp<-nrow(mat)
  colnames(mat)<-rownames(mat)<-paste("sp",1:nsp,sep="")
  mat[which(is.na(mat))]<-0
  matp<- mat; matp[which(matp<0)]<-0; matp[which(matp>0)]<-1
  matn<- mat; matn[which(matn>0)]<-0; matn[which(matn<0)]<-1
  #
  ntrip<-0
  for(i in 1:nsp){
    # first, subset two species outcompeted by a third
    nei<-sum(matn[,i])
    idnei<-as.numeric(which(matn[,i]==1))
      if(nei>=2){
      # number of search = number of possible pairs = n!/2!(n-2)!
      nos<- factorial(nei)/(2*factorial(nei-2))
      counter<-0
      # then, for each outcompeted pair,
      # look if they are both positively associated
      for(k in 1:(nei-1)){for(z in (k+1):nei){
        counter<-counter+1
        if(matp[idnei[k],idnei[z]]==1&matp[idnei[z],idnei[k]]==1)
          ntrip <- ntrip+1
      }
        if(counter==nos) break}
    }
  }
  return(ntrip)
}
```

Finally, we measure the frequency of each three-species association module in each observed plant–plant network.

```r
# observed modules
sites.df$intcomp_o <- NA

for(i in 1:nrow(sites.df)){
  if(sites.df$nsp[i]>=5){
    sites.df$intcomp_o[i] <- triad_census(gnet.neg_o[[i]])[10]
```

```
}}

sites.df$facmc_o    <-NA
sites.df$commfac1_o <-NA
sites.df$commfac2_o <-NA

for(zz in 1:nsites){
  if(sites.df$nsp[zz]>=5){
    sites.df$facmc_o[zz]    <-facmcom(coef.cush_o[[zz]])
    sites.df$commfac1_o[zz] <-commfac1(coef.cush_o[[zz]])
    sites.df$commfac2_o[zz] <-commfac2(coef.cush_o[[zz]])
    }
  if(sites.df$nsp[zz+nsites]>=5){
    sites.df$facmc_o[zz+nsites]    <-facmcom(coef.open_o[[zz]])
    sites.df$commfac1_o[zz+nsites] <-commfac1(coef.open_o[[zz]])
    sites.df$commfac2_o[zz+nsites] <-commfac2(coef.open_o[[zz]])
    }
}
```

Randomisation

Instead of randomising the observed network, as usually done in network ecology, we randomise the raw data – the community matrix M_k (equation 1) with species abundance per plot. We use the `r00_ind` null model of `vegan` which is a quantitative shuffle method that preserves both species occurrence (i.e., matrix filling) and abundance while randomising individuals among plots across species (Oksanen *et al.* 2019). This null model first transforms the quantitative matrix into a binary matrix, then randomises species distribution across the matrix while preserving the number of plots occupied, finally randomly assigned species density per plot while preserving the overall abundance (Oksanen *et al.* 2019).

```
nsimu <- 99
nullmods.cush <- rep(list(rep(list(NA),nsimu)),nsites)
nullmods.open <- rep(list(rep(list(NA),nsimu)),nsites)

for(zz in 1:nsites){
  print(zz)
  nm<-vegan::nullmodel(data.cush.list[[zz]], "r00_ind")
  nulmat<-simulate(nm,nsimu)
  for(kk in 1:nsimu){
    nullmods.cush[[zz]][[kk]]<-nulmat[,,kk]
  }
  nm<-vegan::nullmodel(data.open.list[[zz]], "c0_ind")
  nulmat<-simulate(nm,nsimu)
  for(kk in 1:nsimu){
    nullmods.open[[zz]][[kk]]<-nulmat[,,kk]
```

```
  }
}
```

Then, we estimate species associations α_{ij} for each randomised community matrix using equation 2 and organise those coefficients into adjacency matrices for random networks (equation 3).

```
# cushion
nullcoef.cush_o <- rep(list(rep(list(NA),nsimu)),nsites)

for(zz in 1:nsites){
  print(zz)
  nsp <- ncol(nullmods.cush[[zz]][[1]])
  if(nsp >=5){
    for(kk in 1:nsimu){
      print(kk)
      nullcoef.cush_o[[zz]][[kk]]<-matrix(0,nsp,nsp)
      for(i in 1:nsp){
        mod <- bayesglm(nullmods.cush[[zz]][[kk]][,i]~
                          nullmods.cush[[zz]][[kk]][,-i],
                          family="poisson")
        estimcoef <- coef(mod)[-1]
        posterior <- coef(sim(mod))[,-1]
        for(k in 1:(nsp-1)){
          if(sign(quantile(posterior[,k],0.025))!=
             sign(quantile(posterior[,k],0.975)))
            estimcoef[k] <- 0
        }
        nullcoef.cush_o[[zz]][[kk]][i,-i] <- estimcoef
}}}}

# open
nullcoef.open_o <- rep(list(rep(list(NA),nsimu)),nsites)

for(zz in 1:nsites){
  print(zz)
  nsp <- ncol(nullmods.open[[zz]][[1]])
  if(nsp >=5){
    for(kk in 1:nsimu){
      print(kk)
      nullcoef.open_o[[zz]][[kk]]<-matrix(0,nsp,nsp)
      for(j in 1:nsp){
        mod <- bayesglm(nullmods.open[[zz]][[kk]][,i]~
                          nullmods.open[[zz]][[kk]][,-i],
                          family="poisson")
```

```
        estimcoef <- coef(mod)[-1]
        posterior <- coef(sim(mod))[,-1]
        for(k in 1:(nsp-1)){
          if(sign(quantile(posterior[,k],0.025))!=
             sign(quantile(posterior[,k],0.975)))
             estimcoef[k] <- 0
        }
        nullcoef.open_o[[zz]][[kk]][i,-i] <- estimcoef
}}}}
```

In addition, we randomise using the `c0_ind` null model of `vegan` (Oksanen *et al.* 2019) as well the `vaznull` null model of `bipartite` (Dormann *et al.* 2008) (results shown in Extended Data) and repeat the same procedure.

Network motif analysis

We proceed with estimating the overrepresentation of the observed three-species associations across plant communities. In other words, we are interested in estimating the degree of deviance of observed patterns against those expected by chances (i.e., random networks). This overrepresentation is measured by means of *z-score* for each module m as

$$z_m = \frac{(obs) - e\bar{x}p}{sd(exp)} \tag{8}$$

where the number of observed network modules obs is compared with the mean and standard deviation of the same type of network module m across random networks. To do so, we first measure the frequency of the four network modules (i.e., intransitive competition, facilitation-mediated competition, competition-mediated facilitation 1, competition-mediated facilitation 2) in each random network and calculate their mean and sd. Then, we calcuate the *z-score* for each network module m across observed networks in each plant community globally.

Furthermore, we estimate the significance of the observed pattern by means of *P-value* as

$$\hat{P} = 1 - \frac{\sum_i^{sim} I[obs > exp]}{sim + 1} \tag{9}$$

where $I[obs > exp]$ is an indicator function that equals 1 if the observed module frequency was greater than the random value and 0 otherwise across 99 simulations + 1 empirical value.

```
sites.df$intcomp_o_nullmean  <-NA
sites.df$intcomp_o_nullsd    <-NA
sites.df$facmc_o_nullmean    <-NA
sites.df$facmc_o_nullsd      <-NA
sites.df$commfac1_o_nullmean <-NA
sites.df$commfac1_o_nullsd   <-NA
sites.df$commfac2_o_nullmean <-NA
sites.df$commfac2_o_nullsd   <-NA
```

```
sites.df$intcomp_o_pvalue  <-NA
sites.df$facmc_o_pvalue    <-NA
sites.df$commfac1_o_pvalue <-NA
sites.df$commfac2_o_pvalue <-NA

# p-value network modules
pval<-function(obs, rndm){
  nm <- length(rndm)
  ip <- rep(0, nm)
  for(i in 1:nm){
    if(obs > rndm[i]) ip[i] <- 1
  }
  pvalue <- 1-(sum(ip)/(nm+1))
  return(pvalue)
}

# intransitive competition
for(zz in 1:nsites){
  print(zz)
  # cushion
  if(sites.df$nsp[zz]>=5){
  vec<-(rep(0,nsimu))
  for(kk in 1:nsimu){
    mat<- nullcoef.cush_o[[zz]][[kk]]
    mat[which(mat>0)]<-0
    snet<-graph.adjacency(t(mat), weighted=TRUE)
    vec[kk]<-triad_census(snet)[10]
  }
  sites.df$intcomp_o_nullmean[zz] <-mean(vec)
  sites.df$intcomp_o_nullsd[zz]   <-sd(vec)
  sites.df$intcomp_o_pvalue[zz]   <-
    pval(sites.df$intcomp_o[zz],vec)
  }
  # open
  if(sites.df$nsp[zz+nsites]>=5){
    vec<-(rep(0,nsimu))
    for(kk in 1:nsimu){
      mat<- nullcoef.open_o[[zz]][[kk]]
      mat[which(mat>0)]<-0
      snet<-graph.adjacency(t(mat), weighted=TRUE)
      vec[kk]<-triad_census(snet)[10]
    }
  sites.df$intcomp_o_nullmean[zz+nsites] <-mean(vec)
  sites.df$intcomp_o_nullsd[zz+nsites]   <-sd(vec)
```

```r
  sites.df$intcomp_o_pvalue[zz+nsites]   <-
    pval(sites.df$intcomp_o[zz+nsites],vec)
  }
}

# facilitation-mediated competition
for(zz in 1:nsites){
  print(zz)
  # cushion
  if(sites.df$nsp[zz]>=5){
    vec<-(rep(0,nsimu))
    for(kk in 1:nsimu){
      vec[kk]<-facmcom(nullcoef.cush_o[[zz]][[kk]])
    }
    sites.df$facmc_o_nullmean[zz]<-mean(vec)
    sites.df$facmc_o_nullsd[zz]<-sd(vec)
    sites.df$facmc_o_pvalue[zz]<-pval(sites.df$facmc_o[zz],vec)
  }
  # open
  if(sites.df$nsp[zz+nsites]>=5){
    vec<-(rep(0,nsimu))
    for(kk in 1:nsimu){
      vec[kk]<-facmcom(nullcoef.open_o[[zz]][[kk]])
    }
    sites.df$facmc_o_nullmean[zz+nsites] <-mean(vec)
    sites.df$facmc_o_nullsd[zz+nsites]   <-sd(vec)
    sites.df$facmc_o_pvalue[zz+nsites]   <-
      pval(sites.df$facmc_o[zz+nsites],vec)
    }
}

# competition-mediated facilitation type 1
for(zz in 1:nsites){
  print(zz)
  # cushion
  if(sites.df$nsp[zz]>=5){
    vec<-(rep(0,nsimu))
    for(kk in 1:nsimu){
      vec[kk]<-commfac1(nullcoef.cush_o[[zz]][[kk]])
    }
    sites.df$commfac1_o_nullmean[zz] <-mean(vec)
    sites.df$commfac1_o_nullsd[zz]   <-sd(vec)
    sites.df$commfac1_o_pvalue[zz]   <-
      pval(sites.df$commfac1_o[zz],vec)
```

```r
  }
  # open
  if(sites.df$nsp[zz+nsites]>=5){
    vec<-(rep(0,nsimu))
    for(kk in 1:nsimu){
      vec[kk]<-commfac1(nullcoef.open_o[[zz]][[kk]])
    }
    sites.df$commfac1_o_nullmean[zz+nsites] <-mean(vec)
    sites.df$commfac1_o_nullsd[zz+nsites]   <-sd(vec)
    sites.df$commfac1_o_pvalue[zz+nsites]   <-
      pval(sites.df$commfac1_o[zz+nsites],vec)
  }
}

# competition-mediated facilitation type 2
for(zz in 1:nsites){
  print(zz)
  # cushion
  if(sites.df$nsp[zz]>=5){
    vec<-(rep(0,nsimu))
    for(kk in 1:nsimu){
      vec[kk]<-commfac2(nullcoef.cush_o[[zz]][[kk]])
    }
    sites.df$commfac2_o_nullmean[zz] <-mean(vec)
    sites.df$commfac2_o_nullsd[zz]   <-sd(vec)
    sites.df$commfac2_o_pvalue[zz]   <-
      pval(sites.df$commfac2_o[zz],vec)
  }
  # open
  if(sites.df$nsp[zz+nsites]>=5){
    vec<-(rep(0,nsimu))
    for(kk in 1:nsimu){
      vec[kk]<-commfac2(nullcoef.open_o[[zz]][[kk]])
    }
    sites.df$commfac2_o_nullmean[zz+nsites] <-mean(vec)
    sites.df$commfac2_o_nullsd[zz+nsites]   <-sd(vec)
    sites.df$commfac2_o_pvalue[zz+nsites]   <-
      pval(sites.df$commfac2_o[zz+nsites],vec)
  }
}

# z-scores network modules
for(i in 1:nrow(sites.df)){
  if(sites.df$nsp[i]>=5){
```

```r
   # necessery to correct for sd = 0 to avoid Inf values
    if(sites.df$intcomp_o_nullsd[i]!=0)
      sites.df$intcomp_o_zval[i]<-
        (sites.df$intcomp_o[i]-sites.df$intcomp_o_nullmean[i])/
        sites.df$intcomp_o_nullsd[i]
    if(sites.df$intcomp_o_nullsd[i]==0&sites.df$intcomp_o[i]==0)
      sites.df$intcomp_o_zval[i] <- 0
#
    if(sites.df$intfac_o_nullsd[i]!=0)
      sites.df$intfac_o_zval[i]<-
        (sites.df$intfac_o[i]-sites.df$intfac_o_nullmean[i])/
        sites.df$intfac_o_nullsd[i]
    if(sites.df$intfac_o_nullsd[i]==0&sites.df$intfac_o[i]==0)
     sites.df$intfac_o_zval[i] <- 0
#
    if(sites.df$facmc_o_nullsd[i]!=0)
     sites.df$facmc_o_zval[i]<-
      (sites.df$facmc_o[i]-sites.df$facmc_o_nullmean[i])/
      sites.df$facmc_o_nullsd[i]
    if(sites.df$facmc_o_nullsd[i]==0&sites.df$facmc_o[i]==0)
     sites.df$facmc_o_zval[i] <- 0
#
    if(sites.df$commfac1_o_nullsd[i]!=0)
     sites.df$commfac1_o_zval[i]<-
      (sites.df$commfac1_o[i]-sites.df$commfac1_o_nullmean[i])/
      sites.df$commfac1_o_nullsd[i]
    if(sites.df$commfac1_o_nullsd[i]==0&sites.df$commfac1_o[i]==0)
     sites.df$commfac1_o_zval[i] <- 0
#
    if(sites.df$commfac2_o_nullsd[i]!=0)
     sites.df$commfac2_o_zval[i]<-
      (sites.df$commfac2_o[i]-sites.df$commfac2_o_nullmean[i])/
      sites.df$commfac2_o_nullsd[i]
    if(sites.df$commfac2_o_nullsd[i]==0&sites.df$commfac2_o[i]==0)
      sites.df$commfac2_o_zval[i] <- 0
  }
  else {
    sites.df$intcomp_o_zval[i]  <-NA
    sites.df$intfac_o_zval[i]   <-NA
    sites.df$facmc_o_zval[i]    <-NA
    sites.df$commfac1_o_zval[i] <-NA
    sites.df$commfac1_o_zval[i] <-NA
  }
}
```

Other network modules

We also look at the prevalence of nine other network modules.

```r
### intransitive facilitation
# a -> b ; b -> c ; c -> a
intfac<-function(mat){
    mat[which(mat<0)]<-0
    gnet<-graph.adjacency(t(mat), weighted=TRUE)
    triads<-triad_census(gnet)[10]
    return(triads)
}=
### apparent competition
# a -> +b ; b -> -c ; a -> -c
appcom<-function(mat){
    nsp<-nrow(mat)
    mat[which(is.na(mat))]<-0
    matp<- mat; matp[which(matp<0)]<-0; matp[which(matp>0)]<-1
    matn<- mat; matn[which(matn>0)]<-0; matn[which(matn<0)]<-1
    #
    ntrip<-0
    for(i in 1:nsp){
        # first, subset two species outcompeting a third
        nei<-sum(matn[i,])
        idnei<-as.numeric(which(matn[i,]==1))
        if(nei>=2){
            # number of search = number of possible pairs * 2= n!/(n-2)!
            nos<- factorial(nei)/(factorial(nei-2))
            counter<-0
            # then, for each pair, look if one is facilitating the other
            for(y in 1:nei){for(yy in 1:nei){
                if(y != yy){
                    counter<-counter+1
                    if(matp[idnei[yy],idnei[y]]==1) ntrip <- ntrip+1
                }
            }}
            if(counter==nos) break}
        }
    return(ntrip)
}
### exploitation competition
expcom<-function(mat){
    nsp<-nrow(mat)
    mat[which(is.na(mat))]<-0
    matp<- mat; matp[which(matp<0)]<-0; matp[which(matp>0)]<-1
```

```
    matn<- mat; matn[which(matn>0)]<-0; matn[which(matn<0)]<-1
    #
    ntrip<-0
    for(i in 1:nsp){
        # cols with at least 2 -1 entries
        nei<-sum(matn[,i])
        idnei<-as.numeric(which(matn[,i]==1))
        if(nei>=2){
            # number of search = number of possible pairs * 2= n!/(n-2)!
            nos<- factorial(nei)/(factorial(nei-2))
            counter<-0
            # then, for each pair, look if one is facilitating the other
            for(y in 1:nei){for(yy in 1:nei){
                if(y != yy){
                    counter<-counter+1
                    if(matp[idnei[yy],idnei[y]]==1) ntrip <- ntrip+1
                }
            }}
            if(counter==nos) break}
    }
    return(ntrip)
}
### facilitation cascade
# a -> +b ; b -> +c ; a -> +c
faccas<-function(mat){
    mat[which(mat<0)]<-0
    gnet<-graph.adjacency(t(mat), weighted=TRUE)
    triads<-triad_census(gnet)[9]
    return(triads)
}
## plus: dominance (12)
##       co-dominance (13)
##       quasi-intransitive (14)
##       complete competition (15)
##       quasi-complete competition (16)

# observed networks
sites.df$intfac_o<-NA
sites.df$appcom_o<-NA
sites.df$expcom_o<-NA
sites.df$faccas_o<-NA
sites.df$indfac_o<-NA
sites.df$twln <- NA
sites.df$thrn <- NA
```

```r
sites.df$frtn <- NA
sites.df$fifn <- NA
sites.df$sxtn <- NA
for(zz in 1:nsites){
    if(sites.df$nsp[zz]>=5){
        sites.df$intfac_o[zz]<-intfac(coef.cush_o[[zz]])
        sites.df$appcom_o[zz]<-appcom(coef.cush_o[[zz]])
        sites.df$expcom_o[zz]<-expcom(coef.cush_o[[zz]])
        sites.df$faccas_o[zz]<-faccas(coef.cush_o[[zz]])
        sites.df$indfac_o[zz]<-indfac(coef.cush_o[[zz]])
    }
    if(sites.df$nsp[zz+nsites]>=5){
        sites.df$intfac_o[zz+nsites]<-
          intfac(coef.open_o[[zz]])
        sites.df$appcom_o[zz+nsites]<-
          appcom(coef.open_o[[zz]])
        sites.df$expcom_o[zz+nsites]<-
          expcom(coef.open_o[[zz]])
        sites.df$faccas_o[zz+nsites]<-
          faccas(coef.open_o[[zz]])
        sites.df$indfac_o[zz+nsites]<-
          indfac(coef.open_o[[zz]])
    }
}
for(i in 1:nrow(sites.df)){
    if(sites.df$nsp[i]>=5){
        print(i)
        sites.df$twln[i] <- triad_census(gnet.neg_o[[i]])[12]
        sites.df$thrn[i] <- triad_census(gnet.neg_o[[i]])[13]
        sites.df$frtn[i] <- triad_census(gnet.neg_o[[i]])[14]
        sites.df$fifn[i] <- triad_census(gnet.neg_o[[i]])[15]
        sites.df$sxtn[i] <- triad_census(gnet.neg_o[[i]])[16]
    }}

# random networks
#### intransitive facilitation
for(zz in 1:nsites){
    print(zz)
    # cushion
    if(sites.df$nsp[zz]>=5){
        vec<-(rep(0,nsimu))
        for(kk in 1:nsimu){
            vec[kk]<-intfac(nullcoef.cush_o[[zz]][[kk]])
        }
```

```
        sites.df$intfac_o_nullmean[zz]<-mean(vec)
        sites.df$iintfac_o_nullsd[zz]<-sd(vec)
        sites.df$intfac_o_pvalue[zz]<-
          pval(sites.df$intcomp_o[zz],vec)
    }
    # open
    if(sites.df$nsp[zz+nsites]>=5){
        vec<-(rep(0,nsimu))
        for(kk in 1:nsimu){
            vec[kk]<-intfac(nullcoef.open_o[[zz]][[kk]])
        }
        sites.df$intfac_o_nullmean[zz+nsites]<-mean(vec)
        sites.df$intfac_o_nullsd[zz+nsites]<-sd(vec)
        sites.df$intfac_o_pvalue[zz+nsites]<-
          pval(sites.df$intfac_o[zz+nsites],vec)
    }
}
### apparent competition
for(zz in 1:nsites){
    print(zz)
    # cushion
    if(sites.df$nsp[zz]>=5){
        vec<-(rep(0,nsimu))
        for(kk in 1:nsimu){
            vec[kk]<-appcom(nullcoef.cush_o[[zz]][[kk]])
        }
        sites.df$appcom_o_nullmean[zz]<-mean(vec)
        sites.df$appcom_o_nullsd[zz]<-sd(vec)
        sites.df$appcom_o_pvalue[zz]<-
          pval(sites.df$appcom_o[zz],vec)
    }
    # open
    if(sites.df$nsp[zz+nsites]>=5){
        vec<-(rep(0,nsimu))
        for(kk in 1:nsimu){
            vec[kk]<-appcom(nullcoef.open_o[[zz]][[kk]])
        }
        sites.df$appcom_o_nullmean[zz+nsites]<-mean(vec)
        sites.df$appcom_o_nullsd[zz+nsites]<-sd(vec)
        sites.df$appcom_o_pvalue[zz+nsites]<-
          pval(sites.df$appcom_o[zz+nsites],vec)
    }
}
### exploit comp
```

```r
for(zz in 1:nsites){
    print(zz)
    # cushion
    if(sites.df$nsp[zz]>=5){
        vec<-(rep(0,nsimu))
        for(kk in 1:nsimu){
            vec[kk]<-expcom(nullcoef.cush_o[[zz]][[kk]])
        }
        sites.df$expcom_o_nullmean[zz]<-mean(vec)
        sites.df$expcom_o_nullsd[zz]<-sd(vec)
        sites.df$expcom_o_pvalue[zz]<-
          pval(sites.df$expcom_o[zz],vec)
    }
    # open
    if(sites.df$nsp[zz+nsites]>=5){
        vec<-(rep(0,nsimu))
        for(kk in 1:nsimu){
            vec[kk]<-expcom(nullcoef.open_o[[zz]][[kk]])
        }
        sites.df$expcom_o_nullmean[zz+nsites]<-mean(vec)
        sites.df$expcom_o_nullsd[zz+nsites]<-sd(vec)
        sites.df$expcom_o_pvalue[zz+nsites]<-
          pval(sites.df$expcom_o[zz+nsites],vec)
    }
}
### facilitation cascade
for(zz in 1:nsites){
    print(zz)
    # cushion
    if(sites.df$nsp[zz]>=5){
        vec<-(rep(0,nsimu))
        for(kk in 1:nsimu){
            vec[kk]<-faccas(nullcoef.cush_o[[zz]][[kk]])
        }
        sites.df$faccas_o_nullmean[zz]<-mean(vec)
        sites.df$faccas_o_nullsd[zz]<-sd(vec)
        sites.df$faccas_o_pvalue[zz]<-
          pval(sites.df$faccas_o[zz],vec)
    }
    # open
    if(sites.df$nsp[zz+nsites]>=5){
        vec<-(rep(0,nsimu))
        for(kk in 1:nsimu){
            vec[kk]<-faccas(nullcoef.open_o[[zz]][[kk]])
```

```
        }
        sites.df$faccas_o_nullmean[zz+nsites]<-mean(vec)
        sites.df$faccas_o_nullsd[zz+nsites]<-sd(vec)
        sites.df$faccas_o_pvalue[zz+nsites]<-
          pval(sites.df$faccas_o[zz+nsites],vec)
    }
}
## twln
for(zz in 1:nsites){
    # cushion
    if(sites.df$nsp[zz]>=5){
        vec<-(rep(0,nsimu))
        for(kk in 1:nsimu){
            mat<- nullcoef.cush_o[[zz]][[kk]]
            mat[which(mat>0)]<-0
            snet<-graph.adjacency(t(mat), weighted=TRUE)
            vec[kk]<-triad_census(snet)[12]
        }
        sites.df$twln_o_nullmean[zz]<-mean(vec)
        sites.df$twln_o_nullsd[zz]<-sd(vec)
        sites.df$twln_o_pvalue[zz]<-
          pval(sites.df$twln[zz],vec)
    }
    # open
    if(sites.df$nsp[zz+nsites]>=5){
        vec<-(rep(0,nsimu))
        for(kk in 1:nsimu){
            mat<- nullcoef.open_o[[zz]][[kk]]
            mat[which(mat>0)]<-0
            snet<-graph.adjacency(t(mat), weighted=TRUE)
            vec[kk]<-triad_census(snet)[12]
        }
        sites.df$twln_o_nullmean[zz+nsites]<-mean(vec)
        sites.df$twln_o_nullsd[zz+nsites]<-sd(vec)
        sites.df$twln_o_pvalue[zz+nsites]<-
          pval(sites.df$twln[zz+nsites],vec)
    }
}
## thrn
for(zz in 1:nsites){
    # cushion
    if(sites.df$nsp[zz]>=5){
        vec<-(rep(0,nsimu))
        for(kk in 1:nsimu){
```

```
            mat<- nullcoef.cush_o[[zz]][[kk]]
            mat[which(mat>0)]<-0
            snet<-graph.adjacency(t(mat), weighted=TRUE)
            vec[kk]<-triad_census(snet)[13]
        }
        sites.df$thrn_o_nullmean[zz]<-mean(vec)
        sites.df$thrn_o_nullsd[zz]<-sd(vec)
        sites.df$thrn_o_pvalue[zz]<-
          pval(sites.df$thrn[zz],vec)
    }
    # open
    if(sites.df$nsp[zz+nsites]>=5){
        vec<-(rep(0,nsimu))
        for(kk in 1:nsimu){
            mat<- nullcoef.open_o[[zz]][[kk]]
            mat[which(mat>0)]<-0
            snet<-graph.adjacency(t(mat), weighted=TRUE)
            vec[kk]<-triad_census(snet)[13]
        }
        sites.df$thrn_o_nullmean[zz+nsites]<-mean(vec)
        sites.df$thrn_o_nullsd[zz+nsites]<-sd(vec)
        sites.df$thrn_o_pvalue[zz+nsites]<-
          pval(sites.df$thrn[zz+nsites],vec)
    }
}

## frtn
for(zz in 1:nsites){
    # cushion
    if(sites.df$nsp[zz]>=5){
        vec<-(rep(0,nsimu))
        for(kk in 1:nsimu){
            mat<- nullcoef.cush_o[[zz]][[kk]]
            mat[which(mat>0)]<-0
            snet<-graph.adjacency(t(mat), weighted=TRUE)
            vec[kk]<-triad_census(snet)[14]
        }
        sites.df$frtn_o_nullmean[zz]<-mean(vec)
        sites.df$frtn_o_nullsd[zz]<-sd(vec)
        sites.df$frtn_o_pvalue[zz]<-
          pval(sites.df$frtn[zz],vec)
    }
    # open
    if(sites.df$nsp[zz+nsites]>=5){
```

```
        vec<-(rep(0,nsimu))
        for(kk in 1:nsimu){
            mat<- nullcoef.open_o[[zz]][[kk]]
            mat[which(mat>0)]<-0
            snet<-graph.adjacency(t(mat), weighted=TRUE)
            vec[kk]<-triad_census(snet)[14]
        }
        sites.df$frtn_o_nullmean[zz+nsites]<-mean(vec)
        sites.df$frtn_o_nullsd[zz+nsites]<-sd(vec)
        sites.df$frtn_o_pvalue[zz+nsites]<-
          pval(sites.df$frtn[zz+nsites],vec)
    }
}

## fifn
for(zz in 1:nsites){
    # cushion
    if(sites.df$nsp[zz]>=5){
        vec<-(rep(0,nsimu))
        for(kk in 1:nsimu){
            mat<- nullcoef.cush_o[[zz]][[kk]]
            mat[which(mat>0)]<-0
            snet<-graph.adjacency(t(mat), weighted=TRUE)
            vec[kk]<-triad_census(snet)[15]
        }
        sites.df$fifn_o_nullmean[zz]<-mean(vec)
        sites.df$fifn_o_nullsd[zz]<-sd(vec)
        sites.df$fifn_o_pvalue[zz]<-
          pval(sites.df$fifn[zz],vec)
    }
    # open
    if(sites.df$nsp[zz+nsites]>=5){
        vec<-(rep(0,nsimu))
        for(kk in 1:nsimu){
            mat<- nullcoef.open_o[[zz]][[kk]]
            mat[which(mat>0)]<-0
            snet<-graph.adjacency(t(mat), weighted=TRUE)
            vec[kk]<-triad_census(snet)[15]
        }
        sites.df$fifn_o_nullmean[zz+nsites]<-mean(vec)
        sites.df$fifn_o_nullsd[zz+nsites]<-sd(vec)
        sites.df$fifn_o_pvalue[zz+nsites]<-
          pval(sites.df$fifn[zz+nsites],vec)
    }
```

```r
}

## sxtn
for(zz in 1:nsites){
    # cushion
    if(sites.df$nsp[zz]>=5){
        vec<-(rep(0,nsimu))
        for(kk in 1:nsimu){
            mat<- nullcoef.cush_o[[zz]][[kk]]
            mat[which(mat>0)]<-0
            snet<-graph.adjacency(t(mat), weighted=TRUE)
            vec[kk]<-triad_census(snet)[16]
        }
        sites.df$sxtn_o_nullmean[zz]<-mean(vec)
        sites.df$sxtn_o_nullsd[zz]<-sd(vec)
        sites.df$sxtn_o_pvalue[zz]<-
          pval(sites.df$sxtn[zz],vec)
    }
    # open
    if(sites.df$nsp[zz+nsites]>=5){
        vec<-(rep(0,nsimu))
        for(kk in 1:nsimu){
            mat<- nullcoef.open_o[[zz]][[kk]]
            mat[which(mat>0)]<-0
            snet<-graph.adjacency(t(mat), weighted=TRUE)
            vec[kk]<-triad_census(snet)[16]
        }
        sites.df$sxtn_o_nullmean[zz+nsites]<-mean(vec)
        sites.df$sxtn_o_nullsd[zz+nsites]<-sd(vec)
        sites.df$sxtn_o_pvalue[zz+nsites]<-
          pval(sites.df$sxtn[zz+nsites],vec)
    }
}
```

```r
# prevalence of network modules worldwide
length(which(sites.df$intfac_o_pvalue<0.05))/siti*100
```

```
## [1] 3.703704
```

```r
length(which(sites.df$appcom_o_pvalue<0.05))/siti*100
```

```
## [1] 23.45679
```

```r
length(which(sites.df$expcom_o_pvalue<0.05))/siti*100
```

```
## [1] 17.90123
```

```r
length(which(sites.df$faccas_o_pvalue<0.05))/siti*100
```

```
## [1] 30.8642
```

```r
length(which(sites.df$twln_o_pvalue<0.05))/siti*100
```

```
## [1] 20.37037
```

```r
length(which(sites.df$thrn_o_pvalue<0.05))/siti*100
```

```
## [1] 9.876543
```

```r
length(which(sites.df$frtn_o_pvalue<0.05))/siti*100
```

```
## [1] 6.790123
```

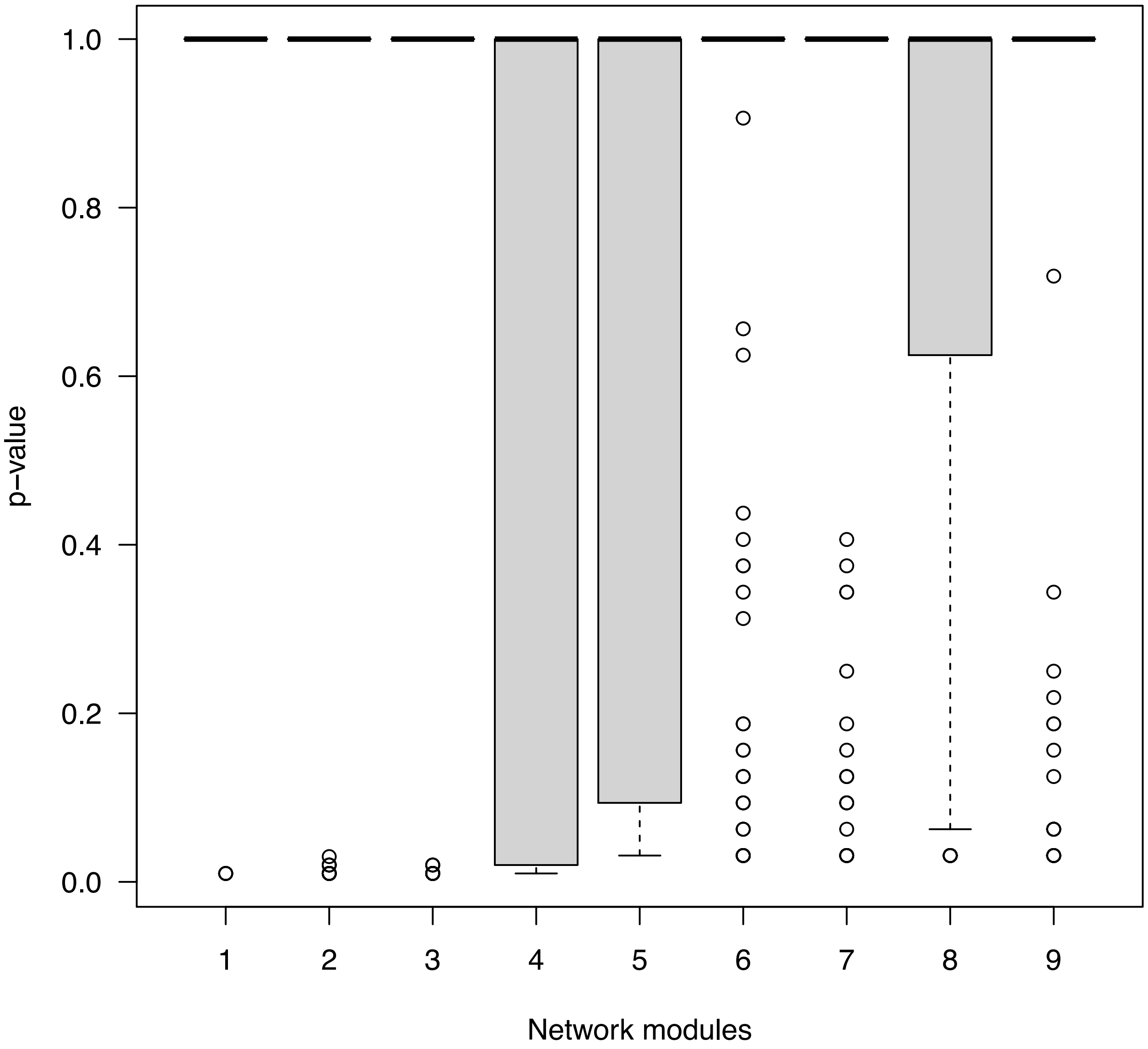
```r
length(which(sites.df$fifn_o_pvalue<0.05))/siti*100
```

```
## [1] 9.259259
```

```r
length(which(sites.df$sxtn_o_pvalue<0.05))/siti*100
```

```
## [1] 5.555556
```

```r
boxplot(cbind(sites.df$intfac_o_pvalue,
              sites.df$appcom_o_pvalue,
              sites.df$expcom_o_pvalue,
              sites.df$faccas_o_pvalue,
              sites.df$twln_o_pvalue,
              sites.df$thrn_o_pvalue,
              sites.df$frtn_o_pvalue,
              sites.df$fifn_o_pvalue,
              sites.df$sxtn_o_pvalue),
        las = 1, xlab = 'Network modules', ylab = 'p-value')
```



Results indicate that these network modules are poorly represented in the examined alpine plant communities at the global scale. In fact, their prevalence is significantly lower than the previously explored motifs (overrepresentation higher than 34%). These results corroborate our findings about the importance of integrating competition and facilitation.

Statistics

We include environmental data of temperature, aridity and biodiversity (from Cavieres *et al.* 2014).

```
sites.df2 <- merge(sites.df, biodiv,
                by.x=c("habitat","site"),
                by.y=c("habitat","code"), sort=FALSE)
sites.df2 <- merge(sites.df2,envdata,
                by.x="site",
                by.y="Site", sort=FALSE, all.x=TRUE)
sites.df2$site.name <-NULL
sites.df2$temp <-NA
sites.df2$temp <- ifelse(sites.df2$TMAX.JUNE<=18, "low", sites.df2$temp)
sites.df2$temp <- ifelse(sites.df2$TMAX.JUNE>18&sites.df2$TMAX.JUNE<=26,
                         "med", sites.df2$temp)
sites.df2$temp <- ifelse(sites.df2$TMAX.JUNE>26, "high",
                         sites.df2$temp)
sites.df2$temp <- factor(sites.df2$temp, order=T,
                         levels=c("low","med","high"))
sites.df2$arid <- 1/sites.df2$P.TSUMMER
sites.df2$aridity <- NA
sites.df2$aridity <- ifelse(sites.df2$arid<=0.2, "low", sites.df2$aridity)
sites.df2$aridity <- ifelse(sites.df2$arid>0.2&sites.df2$arid <=0.5,
                            "med", sites.df2$aridity)
sites.df2$aridity <- ifelse(sites.df2$arid>0.5, "high", sites.df2$aridity)
sites.df2$aridity <- factor(sites.df2$aridity, order=T,
                            levels=c("low","med","high"))
sites.df2 <- sites.df2[which(sites.df2$nsp>=5&sites.df2$nass>=3),]
```

Distribution of alpha

We examine how pairwise species associations α_{ij} vary across environmnetal conditions (i.e., temperature, aridity and habitat).

```
aij.df <- data.frame(matrix(nrow=length(unlist(aij_o)),ncol=5))
colnames(aij.df) <- c("aij","site","habitat","temp","aridity")
aij.df$aij <- unlist(aij_o)

riga.a<-1
riga.b<-0

for(i in 1:nsites){
  if(sites.df$nsp[i]>=5&sites.df$nass_o[i]>0){
    aij<-which(coef.cush_o[[i]]!=0)
    toadd<-length(aij)
```

```
    riga.a<-riga.b+1
    riga.b<-riga.a+toadd-1
    aij.df$site[riga.a:riga.b]<-
      rep(sites.df$site[i],toadd)
    aij.df$habitat[riga.a:riga.b]<-
      rep(as.character(sites.df$habitat[i]),toadd)
  }
  if(sites.df$nsp[i+nsites]>=5&sites.df$nass_o[i+nsites]>0){
    aij<-which(coef.open_o[[i]]!=0)
    toadd<-length(aij)
    riga.a<-riga.b+1
    riga.b<-riga.a+toadd-1
    aij.df$site[riga.a:riga.b]<-
      rep(sites.df$site[i+nsites],toadd)
    aij.df$habitat[riga.a:riga.b]<-
      rep(as.character(sites.df$habitat[i+nsites]),toadd)
  }
}

for(i in 1:nrow(aij.df)){
  aij.df$temp[i] <-
    as.character(sites.df2$temp[which(sites.df2$site==aij.df$site[i]&
    as.character(sites.df2$habitat)==aij.df$habitat[i])])
  aij.df$aridity[i] <-
    as.character(sites.df2$aridity[which(sites.df2$site==aij.df$site[i]&
    as.character(sites.df2$habitat)==aij.df$habitat[i])])
}

aij.df$site    <-factor(aij.df$site)
aij.df$habitat <-factor(aij.df$habitat,
                        levels=c("open","cush"))
aij.df$temp    <-factor(aij.df$temp, ordered=TRUE,
                        levels=c("low","med","high"))
aij.df$aridity <-factor(aij.df$aridity, ordered=TRUE,
                        levels=c("low","med","high"))

for(i in 1:nsites){
  sites.df$mediass_o[i]       <- median(aij_o[[i]])
  sites.df$mediass_o[i+nsites]<- median(aij_o[[i+nsites]])
}
```

We do so using mixed-effects models (`lme` function in `lme4`) (Bates *et al.* 2015) to examine the mean and variance of α_{ij} (two separate models) in response to temperature, aridity and habitat; site was included as random effect. Results show weak relationships between pairwise species associations and environmental conditions.

```r
mod.aij <- lmer(aij ~ temp + aridity + habitat  + (1|site),
                      data=aij.df, na.action=na.exclude)
Anova(mod.aij)
```

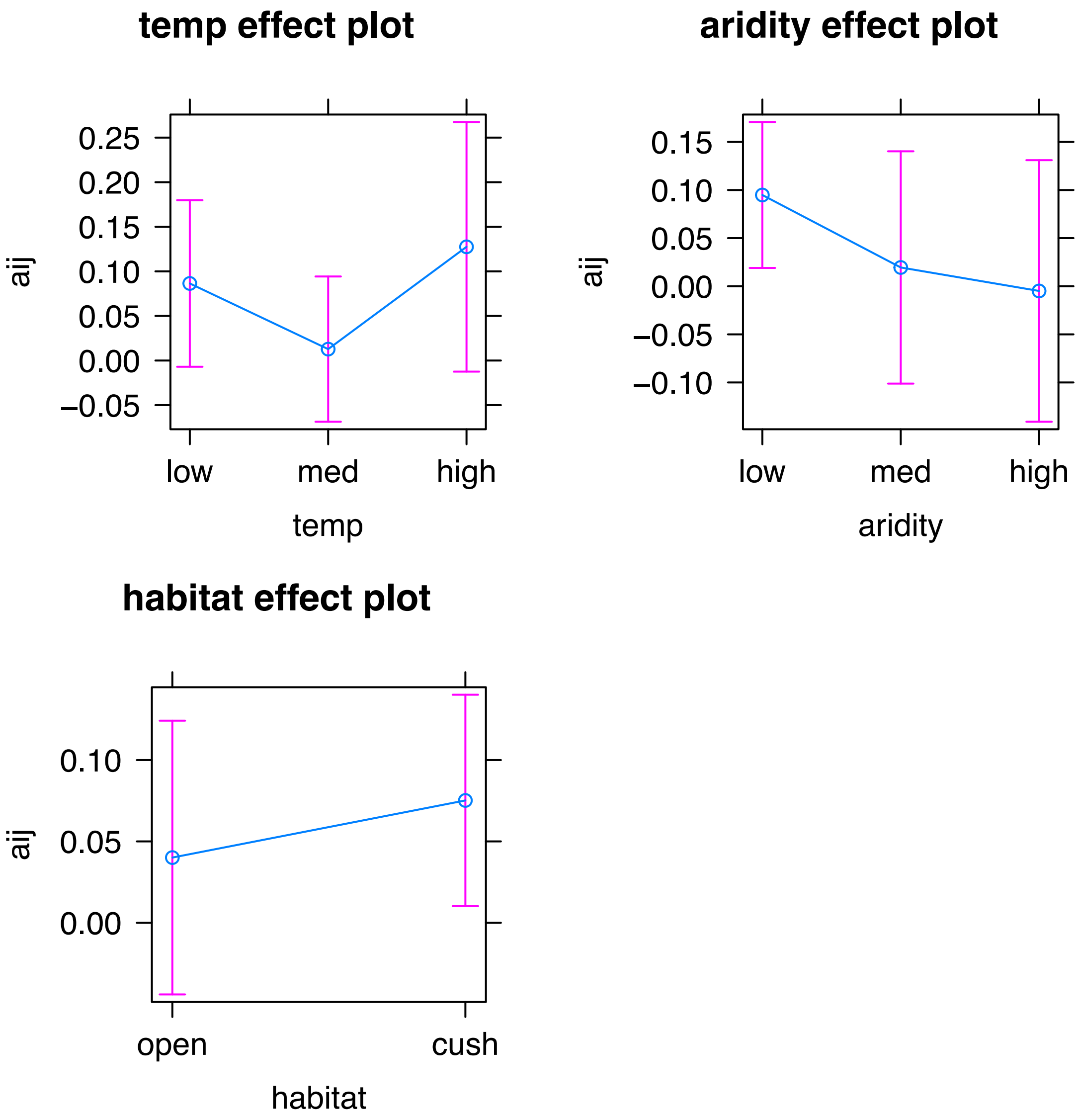
```
## Analysis of Deviance Table (Type II Wald chisquare tests)
##
## Response: aij
##           Chisq Df Pr(>Chisq)
## temp     2.3256  2     0.3126
## aridity 1.6591  2     0.4363
## habitat 0.4532  1     0.5008
```

```r
summary(mod.aij)
```

```
## Linear mixed model fit by REML. t-tests use Satterthwaite's method [
## lmerModLmerTest]
## Formula: aij ~ temp + aridity + habitat + (1 | site)
##    Data: aij.df
##
## REML criterion at convergence: 43327.7
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -35.223  -0.175   0.033   0.251  17.941
##
## Random effects:
##  Groups   Name        Variance Std.Dev.
##  site     (Intercept) 0.004857 0.06969
##  Residual             5.797109 2.40772
## Number of obs: 9423, groups:  site, 75
##
## Fixed effects:
##                Estimate Std. Error         df t value Pr(>|t|)
## (Intercept)     0.02773    0.04579  248.79560   0.606    0.545
## temp.L          0.02905    0.06257   79.13148   0.464    0.644
## temp.Q          0.07690    0.05046   52.11150   1.524    0.134
## aridity.L      -0.07052    0.05945   96.87699  -1.186    0.238
## aridity.Q       0.02077    0.05771   88.20146   0.360    0.720
## habitatcush     0.03511    0.05215 5405.07241   0.673    0.501
##
## Correlation of Fixed Effects:
##              (Intr) temp.L temp.Q ardt.L ardt.Q
## temp.L        0.020
## temp.Q        0.028  0.339
```

```
## aridity.L    0.237 -0.458 -0.318
## aridity.Q   -0.100  0.185  0.284  0.117
## habitatcush -0.737 -0.068  0.018  0.023  0.077
```

```
plot(allEffects(mod.aij), multiline=TRUE)
```



```
mod.varaij <- lmer(varass_o ~ temp + aridity + habitat + (1|site.y),
                              data=sites.df2, na.action=na.exclude)
Anova(mod.varaij)
```

```
## Analysis of Deviance Table (Type II Wald chisquare tests)
##
## Response: varass_o
##          Chisq Df Pr(>Chisq)
## temp    2.2924  2     0.3178
## aridity 4.2265  2     0.1208
```

```
## habitat 0.4737  1     0.4913
```

```r
summary(mod.varaij, digits=3)
```

```
## Warning in summary.merMod(as(object, "lmerMod"), ...): additional arguments
## ignored
```

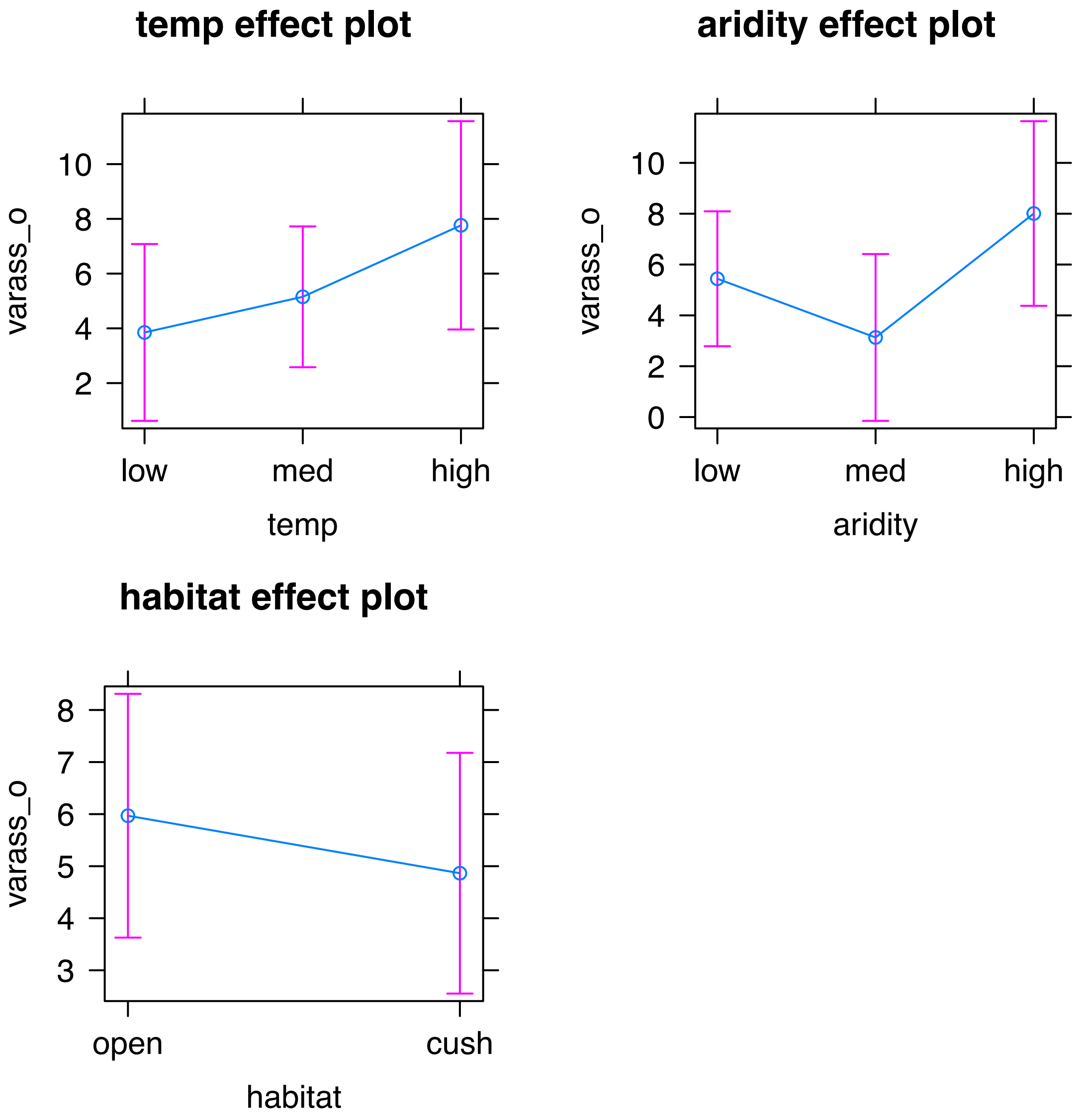
```
## Linear mixed model fit by REML. t-tests use Satterthwaite's method [
## lmerModLmerTest]
## Formula: varass_o ~ temp + aridity + habitat + (1 | site.y)
##    Data: sites.df2
##
## REML criterion at convergence: 1079.5
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -1.0403 -0.3325 -0.1533  0.0871  8.4278
##
## Random effects:
##  Groups   Name        Variance Std.Dev.
##  site.y   (Intercept)  4.571   2.138
##  Residual             95.163   9.755
## Number of obs: 148, groups:  site.y, 55
##
## Fixed effects:
##             Estimate Std. Error      df t value Pr(>|t|)
## (Intercept)   6.2656     1.2219 96.7001   5.128  1.5e-06 ***
## temp.L        2.7679     1.8307 34.1196   1.512   0.1398
## temp.Q        0.5346     1.5120 47.9671   0.354   0.7252
## aridity.L     1.8157     1.6946 43.9921   1.071   0.2898
## aridity.Q     2.9339     1.6036 47.5207   1.830   0.0736 .
## habitatcush  -1.1043     1.6045 95.3409  -0.688   0.4929
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##             (Intr) temp.L temp.Q ardt.L ardt.Q
## temp.L      -0.024
## temp.Q       0.133  0.181
## aridity.L    0.144 -0.389 -0.294
## aridity.Q   -0.018  0.074  0.201  0.075
## habitatcush -0.663  0.003 -0.003  0.009 -0.006
```

```r
confint(mod.varaij)
```

```
## Computing profile confidence intervals ...
```

```
##                   2.5 %    97.5 %
## .sig01       0.0000000  4.597514
## .sigma       8.4825587 10.991930
## (Intercept)  3.9472413  8.610401
## temp.L      -0.6991102  6.226946
## temp.Q      -2.3370359  3.388682
## aridity.L   -1.3971605  5.014239
## aridity.Q   -0.1151037  5.960674
## habitatcush -4.2757145  2.053384
```

```
plot(allEffects(mod.varaij), multiline=TRUE)
```



Network motifs

We examine the overrepresentation of three-species association modules – intransitive competition, facilitation-mediated competition, competition-mediated facilitation type-1 and type-2

– across alpine plant communities globally. First, we examine the significance of observed modules. Results show that intransitive competition is very rare (prevalent in only 3.1% of communities) while network motifs that include both competition and facilitation are prevalent in at least 34% of communities.

```
round(length(
  sites.df2$intcomp_o_pvalue[which(sites.df2$intcomp_o_pvalue<=0.05)])
      /nrow(sites.df2)*100,1)
```

```
## [1] 3.1
```

```
round(length(
  sites.df2$facmc_o_pvalue[which(sites.df2$facmc_o_pvalue<=0.05)])
      /nrow(sites.df2)*100,1)
```

```
## [1] 39.5
```

```
round(length(
  sites.df2$commfac1_o_pvalue[which(sites.df2$commfac1_o_pvalue<=0.05)])
      /nrow(sites.df2)*100,1)
```

```
## [1] 38.3
```

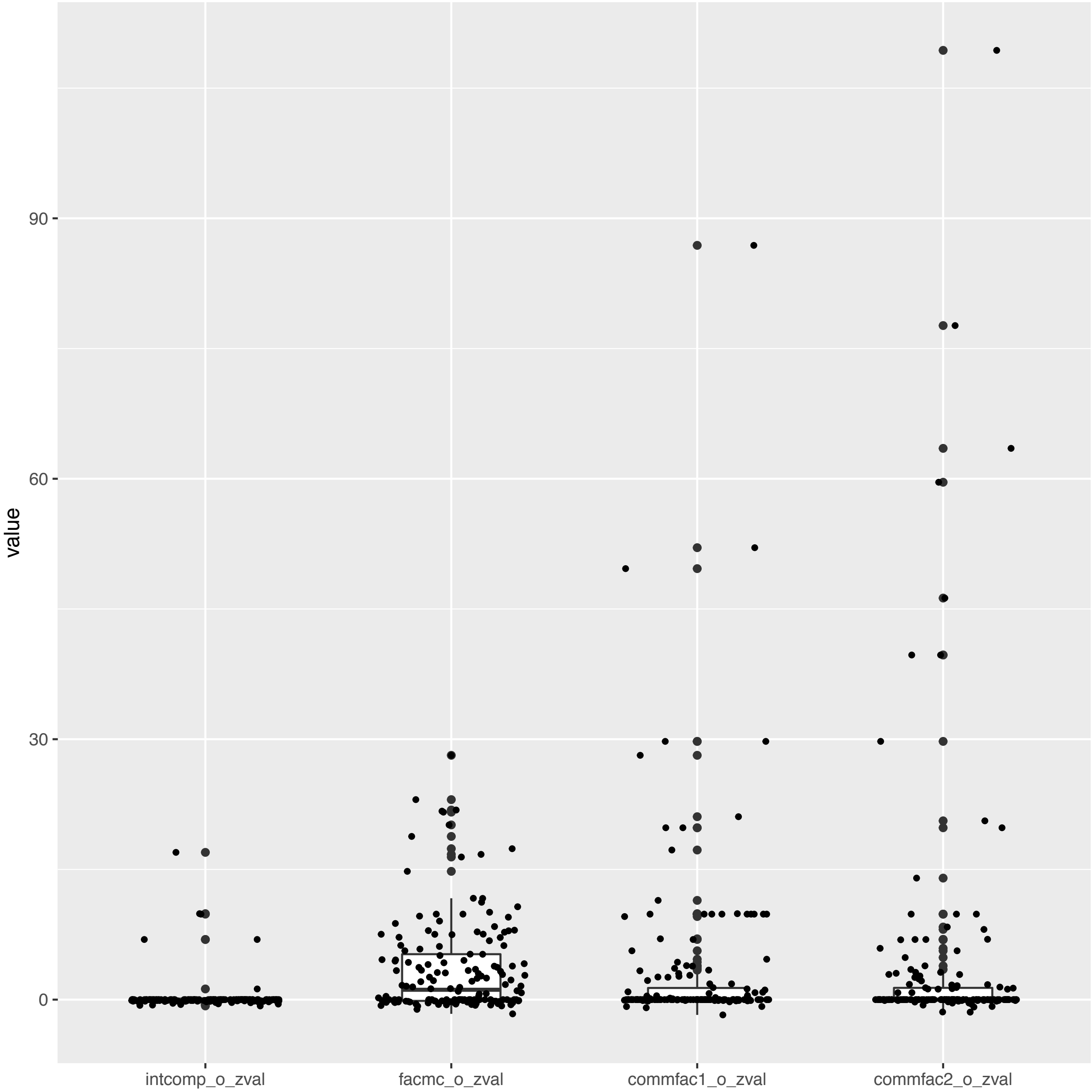
```
round(length(
  sites.df2$commfac2_o_pvalue[which(sites.df2$commfac2_o_pvalue<=0.05)])
      /nrow(sites.df2)*100,1)
```

```
## [1] 34.6
```

Second, we examine the overrepresentation of observed modules across environmental conditions (i.e., temperature, aridity and habitat) using mixed-effects models. Results show that the frequency of intransitive competition is as similar as expected by chances ($\mu = 0.24$, -1.44–1.93) while facilitation-mediated competition, competition-mediated facilitation 1 and 2 were prevalent ($\mu = 3.23$, 1.55–4.92; $\mu = 2.64$, 0.96–4.33; $\mu = 3.01$, 1.37–4.74, respectively), indicating the latters are network motifs occurring in alpine plant communities. Furthermore, results indicate that the overrepresentation of network motifs can change with environmental conditions, particularly competition-mediated facilitation 2 decreases with aridity and increases in the presence of ecosystem engineers.

```
dfmotif <- melt(sites.df2, measure.vars=
           c("intcomp_o_zval",   # z-score intransitive competition
             "facmc_o_zval",     # z-score facilitation-m competition
             "commfac1_o_zval",  # z-score competition- facilitation 1
             "commfac2_o_zval"), # z-score competition- facilitation 1
           variable_name="motif")

ggplot(dfmotif, aes(x = motif, y = value)) +
  geom_boxplot(width = 0.4, fill = "white") +
  geom_jitter(width = 0.3, size = 1) +
```

```
  scale_color_manual(values = modcol) +
  labs(x = NULL)
```



```
mod.motifs <- lmer(value ~ -1+motif/(temp + aridity + habitat) + (1|site.y),
                                data=dfmotif, na.action=na.exclude)

lsmeans(mod.motifs, ~ motif)
```

```
##  motif            lsmean    SE  df lower.CL upper.CL
##  intcomp_o_zval    0.244 0.853 207   -1.438     1.93
##  facmc_o_zval      3.234 0.853 207    1.552     4.92
##  commfac1_o_zval   2.643 0.853 207    0.962     4.33
```

```
##  commfac2_o_zval  3.055 0.853 207    1.373     4.74
##
## Results are averaged over the levels of: habitat, aridity, temp
## Degrees-of-freedom method: kenward-roger
## Confidence level used: 0.95
```

```r
Anova(mod.motifs)
```

```
## Analysis of Deviance Table (Type II Wald chisquare tests)
##
## Response: value
##                 Chisq Df Pr(>Chisq)
## motif         46.1695  4  2.271e-09 ***
## motif:temp     2.3829  8    0.96697
## motif:aridity 16.4951  8    0.03582 *
## motif:habitat 14.7407  4    0.00527 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
summary(mod.motifs)
```

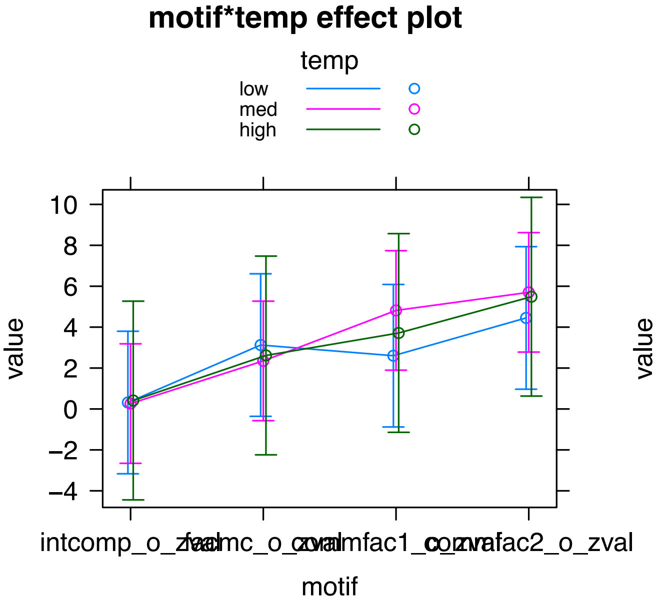
```
## Linear mixed model fit by REML. t-tests use Satterthwaite's method [
## lmerModLmerTest]
## Formula: value ~ -1 + motif/(temp + aridity + habitat) + (1 | site.y)
##    Data: dfmotif
##
## REML criterion at convergence: 4354.3
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -2.2702 -0.3332 -0.1034  0.1121 10.8925
##
## Random effects:
##  Groups   Name        Variance Std.Dev.
##  site.y   (Intercept)  8.053   2.838
##  Residual             73.069   8.548
## Number of obs: 616, groups:  site.y, 55
##
## Fixed effects:
##                                  Estimate Std. Error        df t value
## motifintcomp_o_zval               0.27675    1.09582 426.32142   0.253
## motiffacmc_o_zval                 2.25315    1.09582 426.32142   2.056
## motifcommfac1_o_zval              1.51326    1.09582 426.32142   1.381
## motifcommfac2_o_zval              0.87435    1.09582 426.32142   0.798
## motifintcomp_o_zval:temp.L        0.06862    1.70411 255.21764   0.040
## motiffacmc_o_zval:temp.L         -0.35978    1.70411 255.21764  -0.211
```

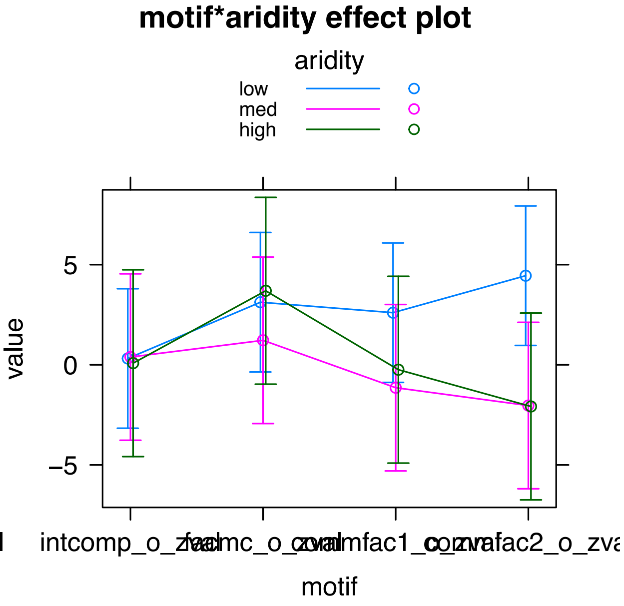
```
## motifcommfac1_o_zval:temp.L        0.78608    1.70411 255.21764   0.461
## motifcommfac2_o_zval:temp.L        0.73300    1.70411 255.21764   0.430
## motifintcomp_o_zval:temp.Q         0.08243    1.37958 306.22829   0.060
## motiffacmc_o_zval:temp.Q           0.42581    1.37958 306.22829   0.309
## motifcommfac1_o_zval:temp.Q       -1.35347    1.37958 306.22829  -0.981
## motifcommfac2_o_zval:temp.Q       -0.59808    1.37958 306.22829  -0.434
## motifintcomp_o_zval:aridity.L     -0.16626    1.56705 294.44596  -0.106
## motiffacmc_o_zval:aridity.L        0.40487    1.56705 294.44596   0.258
## motifcommfac1_o_zval:aridity.L    -2.01342    1.56705 294.44596  -1.285
## motifcommfac2_o_zval:aridity.L    -4.61781    1.56705 294.44596  -2.947
## motifintcomp_o_zval:aridity.Q     -0.15442    1.48153 327.90073  -0.104
## motiffacmc_o_zval:aridity.Q        1.78692    1.48153 327.90073   1.206
## motifcommfac1_o_zval:aridity.Q     1.89724    1.48153 327.90073   1.281
## motifcommfac2_o_zval:aridity.Q     2.62853    1.48153 327.90073   1.774
## motifintcomp_o_zval:habitatcush   -0.06553    1.37764 552.79032  -0.048
## motiffacmc_o_zval:habitatcush      1.96106    1.37764 552.79032   1.423
## motifcommfac1_o_zval:habitatcush   2.26044    1.37764 552.79032   1.641
## motifcommfac2_o_zval:habitatcush   4.36081    1.37764 552.79032   3.165
##                                   Pr(>|t|)
## motifintcomp_o_zval                0.80074
## motiffacmc_o_zval                  0.04038 *
## motifcommfac1_o_zval               0.16802
## motifcommfac2_o_zval               0.42537
## motifintcomp_o_zval:temp.L         0.96791
## motiffacmc_o_zval:temp.L           0.83296
## motifcommfac1_o_zval:temp.L        0.64499
## motifcommfac2_o_zval:temp.L        0.66746
## motifintcomp_o_zval:temp.Q         0.95240
## motiffacmc_o_zval:temp.Q           0.75780
## motifcommfac1_o_zval:temp.Q        0.32733
## motifcommfac2_o_zval:temp.Q        0.66494
## motifintcomp_o_zval:aridity.L      0.91558
## motiffacmc_o_zval:aridity.L        0.79630
## motifcommfac1_o_zval:aridity.L     0.19986
## motifcommfac2_o_zval:aridity.L     0.00347 **
## motifintcomp_o_zval:aridity.Q      0.91705
## motiffacmc_o_zval:aridity.Q        0.22864
## motifcommfac1_o_zval:aridity.Q     0.20124
## motifcommfac2_o_zval:aridity.Q     0.07696 .
## motifintcomp_o_zval:habitatcush    0.96208
## motiffacmc_o_zval:habitatcush      0.15516
## motifcommfac1_o_zval:habitatcush   0.10141
## motifcommfac2_o_zval:habitatcush   0.00163 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

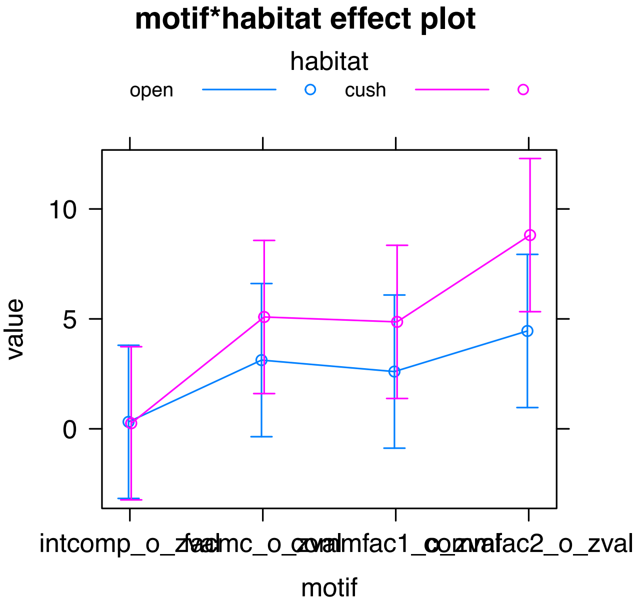
```
confint(mod.motifs)
```

```
##                                        2.5 %    97.5 %
## .sig01                             1.8514025  3.646838
## .sigma                             7.9321578  8.906183
## motifintcomp_o_zval               -1.8229687  2.379188
## motiffacmc_o_zval                  0.1534323  4.355589
## motifcommfac1_o_zval              -0.5864563  3.615700
## motifcommfac2_o_zval              -1.2253642  2.976793
## motifintcomp_o_zval:temp.L        -3.1939244  3.326056
## motiffacmc_o_zval:temp.L          -3.6223292  2.897651
## motifcommfac1_o_zval:temp.L       -2.4764698  4.043510
## motifcommfac2_o_zval:temp.L       -2.5295422  3.990438
## motifintcomp_o_zval:temp.Q        -2.5567716  2.725097
## motiffacmc_o_zval:temp.Q          -2.2133905  3.068478
## motifcommfac1_o_zval:temp.Q       -3.9926634  1.289206
## motifcommfac2_o_zval:temp.Q       -3.2372786  2.044590
## motifintcomp_o_zval:aridity.L     -3.1715429  2.828607
## motiffacmc_o_zval:aridity.L       -2.6004111  3.399738
## motifcommfac1_o_zval:aridity.L    -5.0187006  0.981449
## motifcommfac2_o_zval:aridity.L    -7.6230894 -1.622940
## motifintcomp_o_zval:aridity.Q     -2.9896035  2.684138
## motiffacmc_o_zval:aridity.Q       -1.0482570  4.625485
## motifcommfac1_o_zval:aridity.Q    -0.9379373  4.735805
## motifcommfac2_o_zval:aridity.Q    -0.2066480  5.467094
## motifintcomp_o_zval:habitatcush   -2.7220332  2.590974
## motiffacmc_o_zval:habitatcush     -0.6954452  4.617562
## motifcommfac1_o_zval:habitatcush  -0.3960653  4.916942
## motifcommfac2_o_zval:habitatcush   1.7043064  7.017313
```

```
plot(allEffects(mod.motifs),
     lines=list(multiline=T),
     confint=list(style="bars"))
```







Motif–Biodiversity relationship

We examine the relationship between network motifs and biodiversity across alpine plant communities globally. We use a mixed-effects model with α-diversity as response, the overrepresentation of three-species associations as predictor afer taking into account environmental factors (i.e., temperature, aridity and habitat) as well as global network descriptors (i.e., median species associations α_{ij}, network connectance, and ratio of positive over negative association frequency) as first set of predictors; site was included as random effect. Results indicate that network motifs including competition and facilitation predict biodiversity globally (facilitation-mediated competition β (95% c.i.): 0.075 (0.016–0.133); competition-mediated facilitation type-1: 0.079 (0.027–0.130); competition-mediated facilitation type-2: 0.083 (0.016–0.150)), whereas this is not the case for intransitive competition (-1.020 (-2.556–

0.514)). Morever, the strength and direction of such relationships was rather consistent over environmental conditions.

```r
sites.df2$conn <- sites.df2$nass_o/((sites.df2$nsp^2)-sites.df2$nsp)

sites.df2$ratioass <- (sites.df2$nposass_o+10^-3)/
                      (sites.df2$nnegass_o+10^-3)

dfmotif<-melt(sites.df2,
              measure.vars=
              c("intcomp_o_zval","facmc_o_zval",
              "commfac1_o_zval","commfac2_o_zval"),
              variable_name="motif")

mod.alpha <- lmer(alpha ~ (temp + aridity + habitat) *
                         (mediass_o + scale(conn) + ratioass +
                         motif:value) +
                         (1|site.y),
                         data=dfmotif,
                         na.action=na.exclude)

Anova(mod.alpha)
```

```
## Analysis of Deviance Table (Type II Wald chisquare tests)
##
## Response: alpha
##                          Chisq Df Pr(>Chisq)
## temp                    4.7817  2  0.0915523 .
## aridity                14.2174  2  0.0008180 ***
## habitat               136.3833  1  < 2.2e-16 ***
## mediass_o               0.0363  1  0.8488053
## scale(conn)            15.8728  1  6.774e-05 ***
## ratioass                2.3121  1  0.1283698
## motif:value            36.0923  4  2.770e-07 ***
## temp:mediass_o         23.0058  2  1.010e-05 ***
## temp:scale(conn)        1.4809  2  0.4768949
## temp:ratioass           1.6349  2  0.4415502
## aridity:mediass_o      29.1539  2  4.670e-07 ***
## aridity:scale(conn)    14.8752  2  0.0005887 ***
## aridity:ratioass        3.5354  2  0.1707234
## habitat:mediass_o       2.4670  1  0.1162603
## habitat:scale(conn)     0.6739  1  0.4116952
## habitat:ratioass        1.3444  1  0.2462675
## temp:motif:value       37.1507  8  1.080e-05 ***
## aridity:motif:value     9.6413  8  0.2911081
```

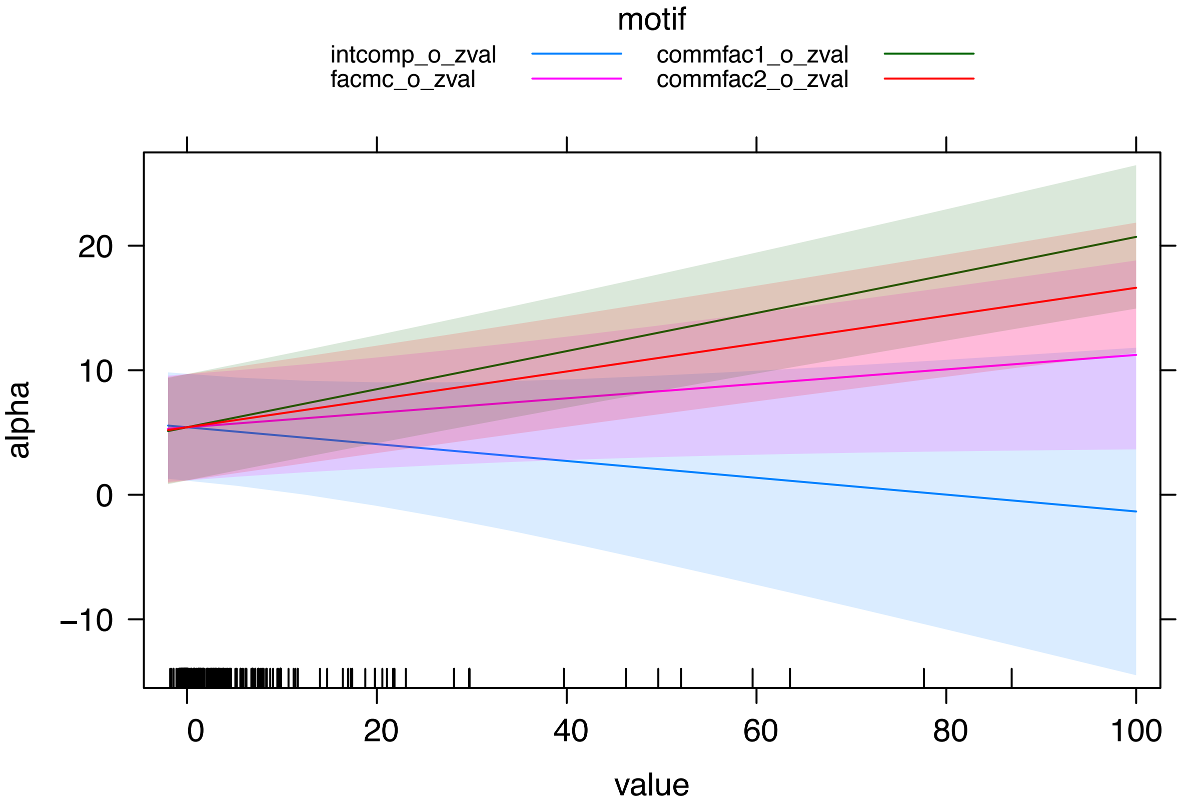
```
## habitat:motif:value  30.3648  4  4.125e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
modalphaci<-round(coef(summary(mod.alpha))[,1],3)
modalphaci<-cbind(modalphaci,round(confint(mod.alpha),3)[-c(1:2),])
print(modalphaci)
```

```
##                                         modalphaci  2.5 % 97.5 %
## (Intercept)                                  3.047  2.455  3.642
## temp.L                                       0.182 -0.967  1.352
## temp.Q                                       1.224  0.328  2.100
## aridity.L                                   -0.920 -1.821  0.001
## aridity.Q                                    0.749  0.075  1.442
## habitatcush                                  1.391  1.165  1.619
## mediass_o                                    0.138 -0.153  0.432
## scale(conn)                                  0.440  0.163  0.715
## ratioass                                     0.034 -0.050  0.117
## motifintcomp_o_zval:value                   -1.020 -2.556  0.514
## motiffacmc_o_zval:value                      0.075  0.016  0.134
## motifcommfac1_o_zval:value                   0.079  0.027  0.130
## motifcommfac2_o_zval:value                   0.083  0.016  0.150
## temp.L:mediass_o                            -1.491 -2.202 -0.782
## temp.Q:mediass_o                            -1.135 -1.624 -0.645
## temp.L:scale(conn)                          -0.339 -0.905  0.224
## temp.Q:scale(conn)                          -0.217 -0.643  0.209
## temp.L:ratioass                              0.073 -0.104  0.250
## temp.Q:ratioass                              0.041 -0.061  0.143
## aridity.L:mediass_o                          1.650  1.046  2.256
## aridity.Q:mediass_o                          0.975  0.515  1.436
## aridity.L:scale(conn)                       -0.444 -1.000  0.114
## aridity.Q:scale(conn)                        0.557  0.155  0.960
## aridity.L:ratioass                          -0.001 -0.002  0.000
## aridity.Q:ratioass                           0.001  0.000  0.002
## habitatcush:mediass_o                       -0.428 -0.942  0.085
## habitatcush:scale(conn)                     -0.117 -0.387  0.152
## habitatcush:ratioass                         0.001  0.000  0.002
## temp.L:motifintcomp_o_zval:value            -0.054 -0.316  0.208
## temp.Q:motifintcomp_o_zval:value            -0.222 -1.123  0.682
## temp.L:motiffacmc_o_zval:value               0.027 -0.046  0.101
## temp.Q:motiffacmc_o_zval:value               0.018 -0.040  0.076
## temp.L:motifcommfac1_o_zval:value            0.023 -0.070  0.117
## temp.Q:motifcommfac1_o_zval:value            0.104  0.046  0.162
## temp.L:motifcommfac2_o_zval:value           -0.090 -0.195  0.015
## temp.Q:motifcommfac2_o_zval:value           -0.013 -0.075  0.049
```

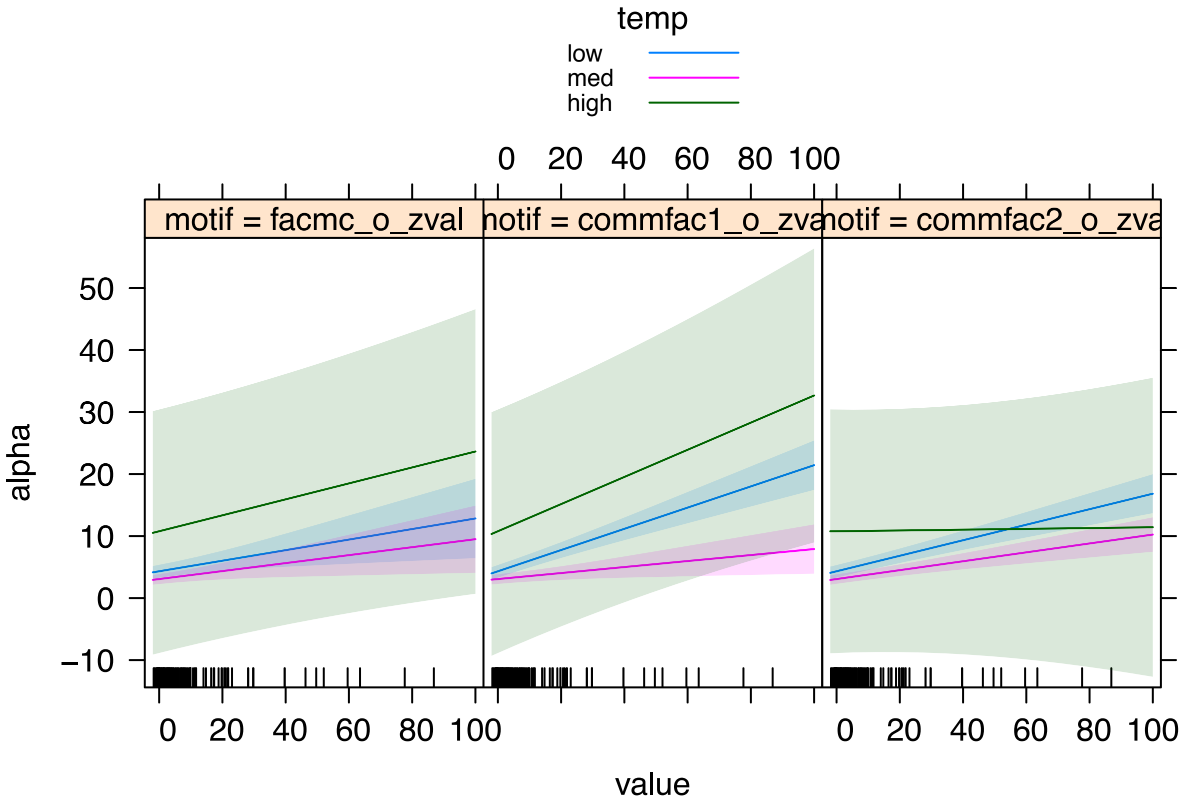
```
## aridity.L:motifintcomp_o_zval:value       -2.199 -5.368  0.966
## aridity.Q:motifintcomp_o_zval:value       -1.348 -3.183  0.484
## aridity.L:motiffacmc_o_zval:value         -0.021 -0.090  0.048
## aridity.Q:motiffacmc_o_zval:value         -0.048 -0.124  0.029
## aridity.L:motifcommfac1_o_zval:value      -0.075 -0.161  0.011
## aridity.Q:motifcommfac1_o_zval:value      -0.011 -0.104  0.083
## aridity.L:motifcommfac2_o_zval:value      -0.030 -0.154  0.094
## aridity.Q:motifcommfac2_o_zval:value      -0.124 -0.250  0.001
## habitatcush:motifintcomp_o_zval:value     -0.191 -1.285  0.904
## habitatcush:motiffacmc_o_zval:value       -0.072 -0.139 -0.005
## habitatcush:motifcommfac1_o_zval:value    -0.037 -0.076  0.002
## habitatcush:motifcommfac2_o_zval:value    -0.071 -0.098 -0.044
```

```
# We plot the relationship between modules and biodiversity
# Fig. 3
plot(Effect(c("motif","value"),mod.alpha),
     lines=list(multiline=TRUE),
     confint=list(style="bands"),
     main="")
```

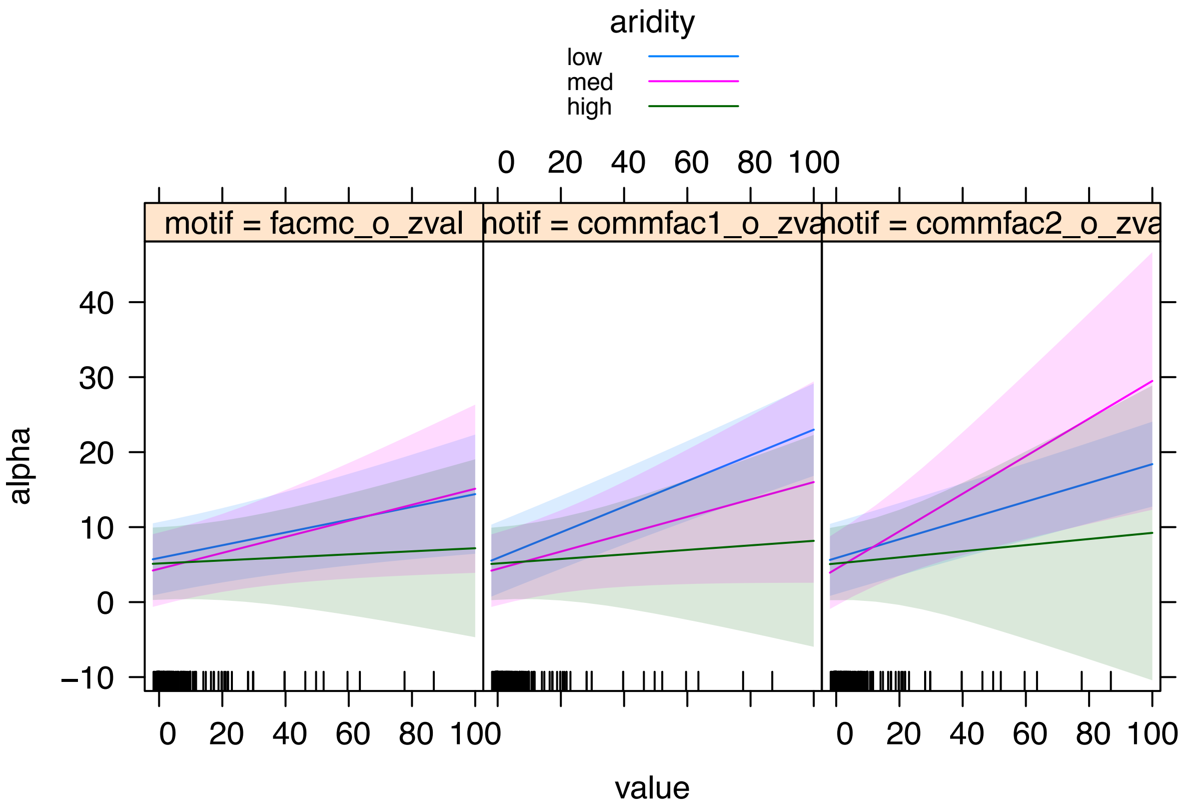


```
# For plotting the interaction terms,
# we dropped intransitive competition because
# it makes unclear and messy plots due to its very wide bands
# plot(allEffects(mod.alpha),
```

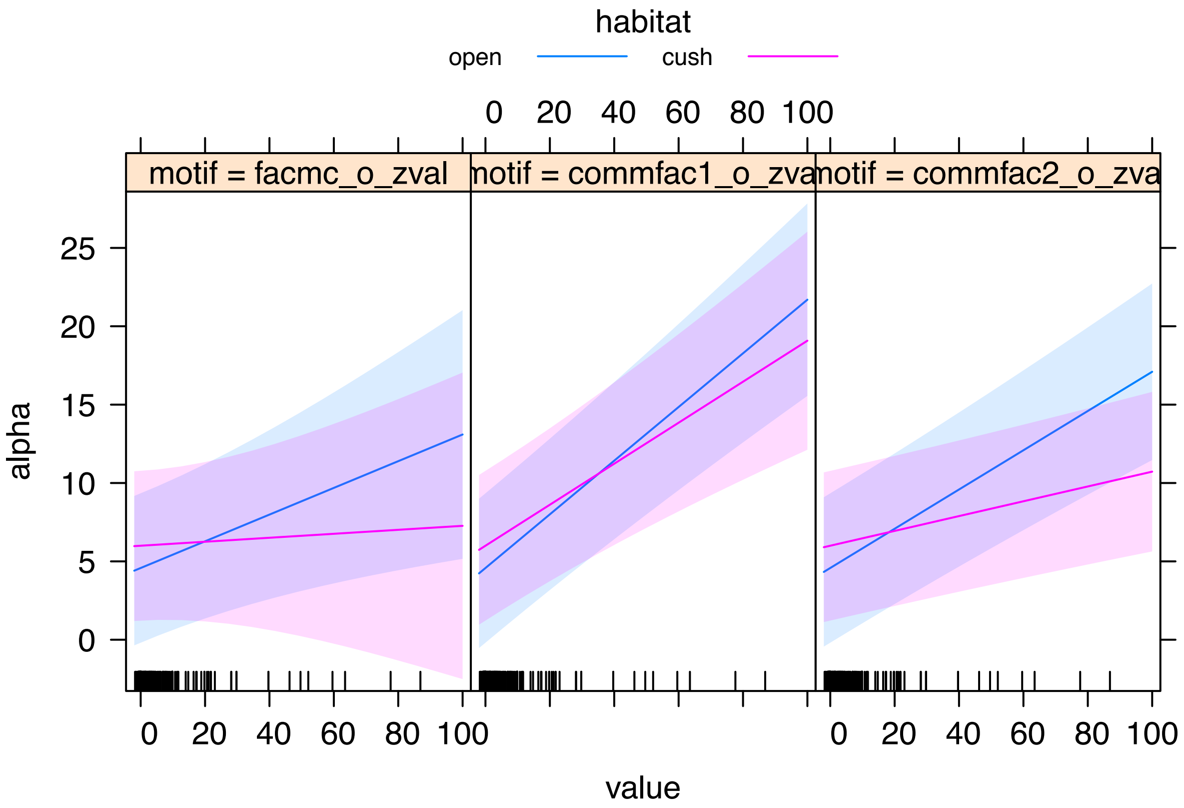
```
#       lines=list(multiline=TRUE),
#       confint=list(style="bands"))
# For Extended Data Fig. 3, we omit 'intcomp'.

dfmotif2noitc <- subset(dfmotif2, motif != "intcomp_o_zval")

mod.alpha2 <- lmer(alpha ~ (temp + aridity + habitat)*
                           (mediass_o + scale(conn) + ratioass +
                           motif:value) +
                           (1|site.y),
                           data=dfmotif2noitc,
                           na.action=na.exclude)

plot(Effect(c("temp","motif","value"),mod.alpha2),
     lines=list(multiline=TRUE),
     confint=list(style="bands"),
     main="")
```



```
plot(Effect(c("aridity","motif","value"),mod.alpha2),
     lines=list(multiline=TRUE),
     confint=list(style="bands"),
     main="")
```



```
plot(Effect(c("habitat","motif","value"),mod.alpha2),
     lines=list(multiline=TRUE),
     confint=list(style="bands"),
     main="")
```


habitat
open
cush
motif = facmc_o_zval
notif = commfac1_o_zva
notif = commfac2_o_zva
alpha
value
0
5
10
15
20
25
0
20
40
60
80
100

Sensitivity analysis

In order to check the robustness of our results, we compared the Bayesian Poisson framework presented in the main text (Harris 2016, Gelman & Su 2018) to four additional frameworks:

1. testing the robustness of resulting networks by repeating our original analysis for each species but with one plot at a time left out of analysis
2. testing the robustness of our model by fitting a Negative Binomial distribution and Zero-Inflated models on sub-sets of the data that exclude rare species (those occurring at less than four samples, therefore repre-senting species that contribute most to data sparsity and the issue of separation)
3. testing the robustness of this approach by examining the effects of removing rare species on model estimation
4. testing the impact of choosing a 95

With the first analysis, we show that the estimated magnitudes and signs of coefficients do not significantly change across the entire dataset when individual samples are left out of analysis. Notably, such consistency of estimates consequently yields to very similar species association networks, therefore highlighting the reliability of our original coefficient estimates and resulting networks.

With the second analysis, we show that removing rare species allowed us fitting different distributions, such as the Negative Binomial and Zero-Inflated Negative Binomial. Results reveal that using different distributions and models does not significantly influence the estimation as coefficients do not change with respect to our original approach.

With the third analysis, we show the robustness of our approach against the removal of species evidencing that removing rare species does not significantly influence the estimation of coefficients for the other species.

With the fourth analysis, we show that changing the CI cutoff to 90% does not change neither the overall distribution of species associations nor the structure of the network as compared to 95%.

Removing samples

In order to assess the robustness of our results, we used an approach of the type 'jackknife sampling', which consisted in comparing the model with all data to subsets with a unique sample left out. Thus, we removed a single sample at the time, for all samples and for each focal species, then estimated the coefficients using the original Bayesian Poisson framework (`bayesglm`) for all communities in all sites.

```r
# observed coefficients
## ecosystem engineering all coefficients
coef.full.cush.list <- rep(list(NA),nsites)
## ecosystem engineering species associations
```

```
matnet.full.cush.list <- rep(list(NA),nsites)

for(k in 1:nsites){
  nsp <- ncol(data.cush.list[[k]])
  print(k)
  if(nsp >= 5){
    coef.full.cush.list[[k]] <-
      matrix(0, ncol=nsp, nrow=nsp)
    matnet.full.cush.list[[k]] <-
      matrix(0, ncol=nsp, nrow=nsp)
    for(i in 1:nsp){
      mod.full <-
        bayesglm(data.cush.list[[k]][,i]~
                   data.cush.list[[k]][,-i],
                   family="poisson")
      coef.full.cush.list[[k]][-i,i] <- coef(mod.full)[-1]
      estimcoef <- coef(mod.full)[-1]
      posterior <- coef(sim(mod.full))[,-1]
      for(z in 1:(nsp-1)){
        if(sign(quantile(posterior[,z],0.025))!=
           sign(quantile(posterior[,z],0.975)))
           estimcoef[z] <- 0
      }
    matnet.full.cush.list[[k]][-i,i] <- estimcoef
    }
  }
}

## open habitat all coefficients
coef.full.open.list <- rep(list(NA),nsites)
## open habitat species associations
matnet.full.open.list <- rep(list(NA),nsites)

for(k in 1:nsites){
  nsp <- ncol(data.open.list[[k]])
  print(k)
  if(nsp >= 5){
    coef.full.open.list[[k]] <-
      matrix(0, ncol=nsp, nrow=nsp)
    matnet.full.open.list[[k]] <-
      matrix(0, ncol=nsp, nrow=nsp)
    for(i in 1:nsp){
      mod.full <-
        bayesglm(data.open.list[[k]][,i]~
```

```
                 data.open.list[[k]][,-i],
                 family="poisson")
      coef.full.open.list[[k]][-i,i] <- coef(mod.full)[-1]
      estimcoef <- coef(mod.full)[-1]
      posterior <- coef(sim(mod.full))[,-1]
      for(z in 1:(nsp-1)){
        if(sign(quantile(posterior[,z],0.025))!=
           sign(quantile(posterior[,z],0.975)))
           estimcoef[z] <- 0
      }
      matnet.full.open.list[[k]][-i,i] <- estimcoef
    }
  }
}

# We then remove one plot at the time
## ecosystem engineering all coefficients
coef.rm.cush.list <- rep(list(NA),nsites)
## ecosystem engineering species associations
matnet.rm.cush.list <- rep(list(NA),nsites)

for(k in 1:nsites){
    nsp <- ncol(data.cush.list[[k]])
    print(k)
    if(nsp >= 5){
        coef.rm.cush.list[[k]] <- rep(
          list(matrix(0, ncol=nsp, nrow=nsp)),
          nrow(data.cush.list[[k]]))
        matnet.rm.cush.list[[k]] <- rep(
          list(matrix(0, ncol=nsp, nrow=nsp)),
          nrow(data.cush.list[[k]]))
        for(j in 1:nrow(data.cush.list[[k]])){
            print(j)
            for(i in 1:nsp){
            mod.rm <-
              bayesglm(data.cush.list[[k]][-j,i]~
                       data.cush.list[[k]][-j,-i],
                       family="poisson")
            coef.rm.cush.list[[k]][[j]][-i,i] <- coef(mod.rm)[-1]
            estimcoef <- coef(mod.rm)[-1]
            posterior <- coef(sim(mod.rm))[,-1]
            for(z in 1:(nsp-1)){
              if(sign(quantile(posterior[,z],0.025))!=
                 sign(quantile(posterior[,z],0.975)))
```

```
                 estimcoef[z] <- 0
            }
            matnet.rm.cush.list[[k]][[j]][-i,i] <- estimcoef
            }
        }
    }
}
## open habitat all coefficients
coef.rm.open.list <- rep(list(NA),nsites)
## open habitat species associations
matnet.rm.open.list <- rep(list(NA),nsites)

for(k in 1:nsites){
    nsp <- ncol(data.open.list[[k]])
    print(k)
    if(nsp >= 5){
        coef.rm.open.list[[k]] <- rep(
          list(matrix(0, ncol=nsp, nrow=nsp)),
          nrow(data.open.list[[k]]))
        matnet.rm.open.list[[k]] <- rep(
          list(matrix(0, ncol=nsp, nrow=nsp)),
          nrow(data.open.list[[k]]))
        for(j in 1:nrow(data.open.list[[k]])){
            print(j)
            for(i in 1:nsp){
                mod.rm <-
                  bayesglm(data.open.list[[k]][-j,i]~
                           data.open.list[[k]][-j,-i],
                           family="poisson")
                coef.rm.open.list[[k]][[j]][-i,i] <- coef(mod.rm)[-1]
                estimcoef <- coef(mod.rm)[-1]
                posterior <- coef(sim(mod.rm))[,-1]
                for(z in 1:(nsp-1)){
                  if(sign(quantile(posterior[,z],0.025))!=
                     sign(quantile(posterior[,z],0.975)))
                    estimcoef[z] <- 0
                }
                matnet.rm.open.list[[k]][[j]][-i,i] <- estimcoef
            }
        }
    }
}
```

Coefficient correlation

First, we compared the original coefficients estimated for the complete, observed dataset with those coefficients estimated for the partial, subset data. We do so by means of Pearson correlation.

```r
cor.obsrem <- rep(list(NA), nsites*2)

for(k in 1:nsites){
  nsp <- ncol(data.cush.list[[k]])
  print(k)
  if(nsp >= 5){
  for(j in 1:nrow(data.cush.list[[k]])){
    cor.obsrem[[k]] <- c(cor.obsrem[[k]],
                    cor.test(coef.full.cush.list[[k]]
                             [-which(coef.full.cush.list[[k]]==0)],
                             coef.rm.cush.list[[k]][[j]]
                             [-which(coef.full.cush.list[[k]]==0)],
                    "pearson", alternative = "two.sided")$estimate)
  }}
  nsp <- ncol(data.open.list[[k]])
  if(nsp >= 5){
      for(j in 1:nrow(data.open.list[[k]])){
        cor.obsrem[[k+nsites]] <- c(cor.obsrem[[k+nsites]],
                          cor.test(coef.full.open.list[[k]]
                                   [-which(coef.full.open.list[[k]]==0)],
                                   coef.rm.open.list[[k]][[j]]
                                   [-which(coef.full.open.list[[k]]==0)],
                          "pearson", alternative = "two.sided")$estimate)
      }}
}

cor.obsrem.all <- unlist(cor.obsrem)
cor.obsrem.all <- cor.obsrem.all[-which(is.na(cor.obsrem.all))]
```

```r
# correlation between observed and removed #
print(mean(cor.obsrem.all))
```

```
## [1] 0.9847607
```

```r
print(median(cor.obsrem.all))
```

```
## [1] 0.9979174
```

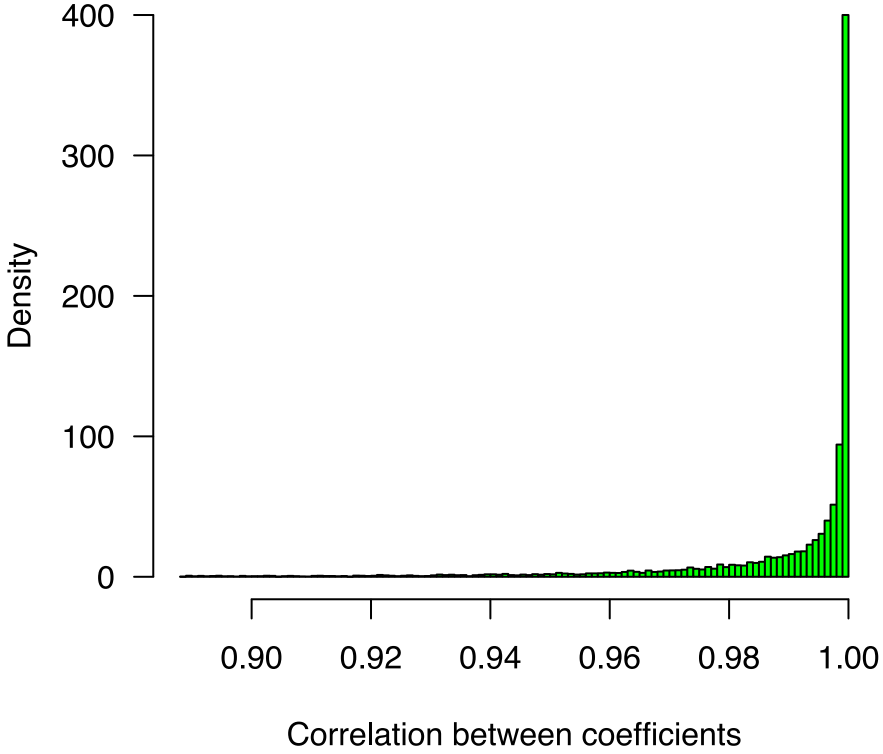
```r
print(sd(cor.obsrem.all))
```

```
## [1] 0.04079851
```

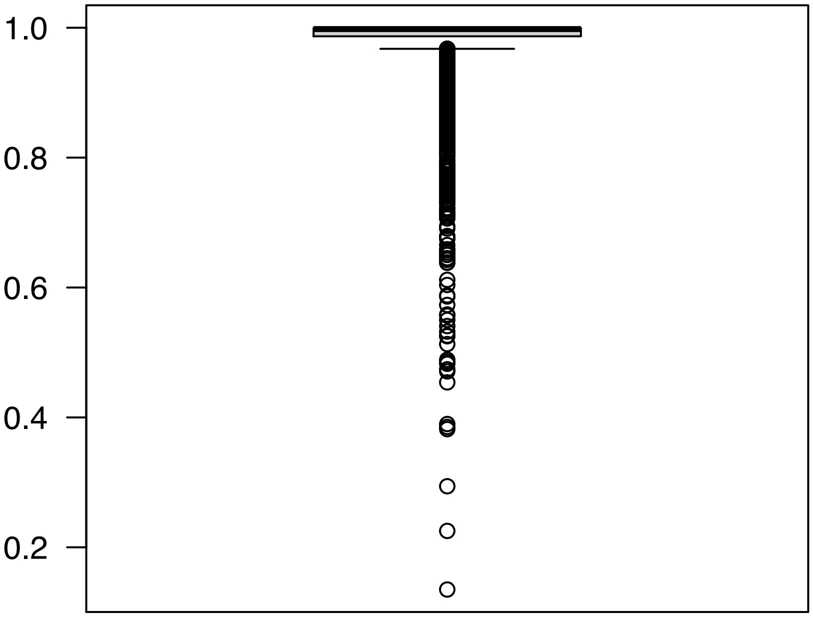
```
print(quantile(cor.obsrem.all, c(0.025, 0.975)))
```

```
##      2.5%     97.5%
## 0.8887052 0.9999774
```

```
hist(cor.obsrem.all
     [which(cor.obsrem.all>0.8887052 & cor.obsrem.all<0.9999774)],
     100, freq=F, las=1, main="", col="green",
     xlab="Correlation between coefficients")
```



```
boxplot(cor.obsrem.all, las=1)
```



Results show that coefficient estimates are not sensitive to data selection and removal of individual observations as both the value and sign of species coefficients remain constant after removing samples from the dataset. Indeed, Pearson correlation of coefficient estimates between the full and partial data was on average 0.985 across all plant communities and sites. This indicates that coefficient estimates are particularly robust against sample removal and are overall reliable for all species.

Differences between coefficients

Having shown that coefficient estimates correlate very well between the full and partial datasets across all plant communities, we measured the differences between coefficients in the two approaches in order to further assess the robustness against sample removal. In particular, we calculated the magnitude of eventual discrepancies as mean differences between coefficients in observed and partial data for each species across all the sites. Furthermore, differences between coefficients in full and partial datasets were standardised with respect to coefficient estimates in the full dataset in order to have a measure of relative deviation from observed values.

```
# mean differences between observed and removed
meandif.list <- rep(list(NA), nsites*2)
# standard differences between observed and removed
stddif.list<- rep(list(NA), nsites*2)

for(k in 1:nsites){
  nsp <- ncol(data.cush.list[[k]])
  print(k)
  if(nsp >= 5){
    meandif.list[[k]] <- matrix(NA, nsp, nsp)
    stddif.list[[k]] <- matrix(NA, nsp, nsp)
```

```
      for(i in 1:nsp){
      for(j in 1:nsp){
        meandif.list[[k]][i,j] <-
          mean(coef.full.cush.list[[k]][i,j]-
              unlist(lapply(coef.rm.cush.list[[k]], "[",i,j)))
        stddif.list[[k]][i,j] <-
          coef.full.cush.list[[k]][i,j]/meandif.list[[k]][i,j]
      }
      }
    diag(meandif.list[[k]]) <- NA
    diag(stddif.list[[k]]) <- NA
  }
  nsp <- ncol(data.open.list[[k]])
  print(k)
  if(nsp >= 5){
    meandif.list[[k+nsites]] <- matrix(NA, nsp, nsp)
    stddif.list[[k+nsites]] <- matrix(NA, nsp, nsp)
      for(i in 1:nsp){
      for(j in 1:nsp){
        meandif.list[[k+nsites]][i,j] <-
          mean(coef.full.open.list[[k]][i,j]-
               unlist(lapply(coef.rm.open.list[[k]], "[",i,j)))
        stddif.list[[k+nsites]][i,j] <-
          coef.full.open.list[[k]][i,j]/meandif.list[[k+nsites]][i,j]
      }
    }
    diag(meandif.list[[k+nsites]]) <- NA
    diag(stddif.list[[k+nsites]]) <- NA
  }
}

meandif.all <- unlist(meandif.list)
meandif.all <- meandif.all[-which(is.na(meandif.all))]

stdmeandif.all <- unlist(stddif.list)
stdmeandif.all <- abs(stdmeandif.all[-which(is.na(stdmeandif.all))])
```

Then, we tested whether those differences were similar to zero. Given that standardised differences greater than two standard deviations suggest that eventual discrepancies are small enough to be negligible, we tested whether those values were overall greater than two.

```
quantile(meandif.all, c(0.025, 0.975))
```

```
##        2.5%       97.5%
## -0.03519170  0.03187464
```

```r
mean(meandif.all)
```

```
## [1] -0.001435195
```

```r
sd(meandif.all)
```

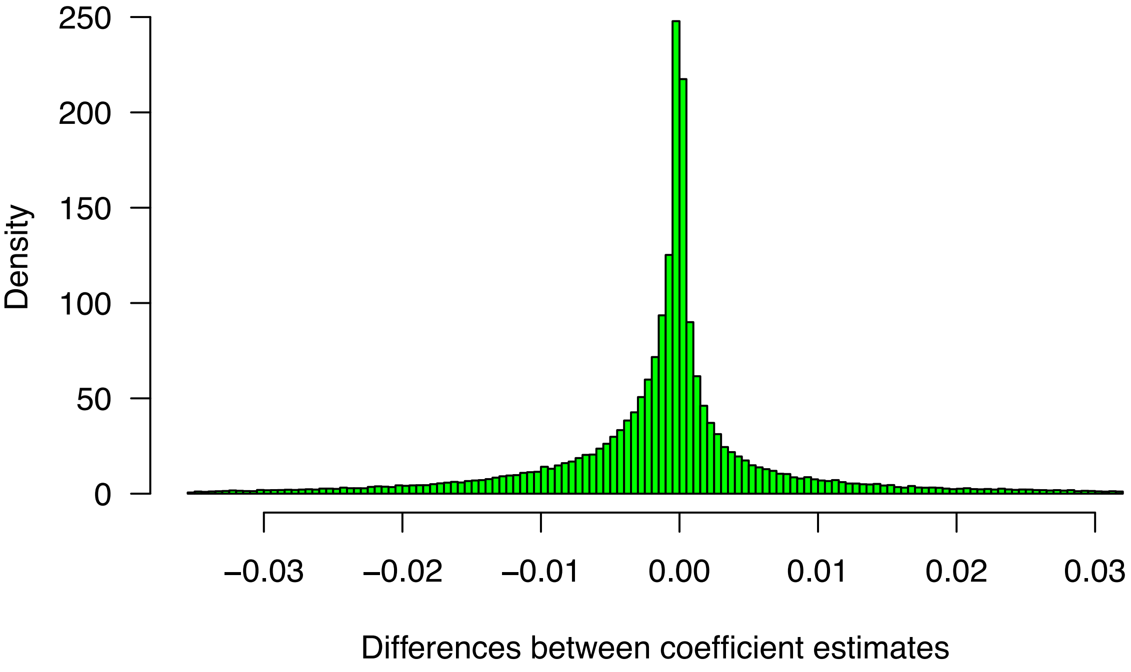
```
## [1] 0.04394773
```

```r
ks.test(meandif.all, 0)
```

```
## Warning in ks.test(meandif.all, 0): p-value will be approximate in the presence
## of ties
```

```
##
##  Two-sample Kolmogorov-Smirnov test
##
## data:  meandif.all and 0
## D = 0.5907, p-value = 0.8764
## alternative hypothesis: two-sided
```

```r
hist(meandif.all
     [which(meandif.all>-0.03519170 & meandif.all<0.03187464)],
     100, freq=F, las=1, main="", col="green",
     xlab="Differences between coefficient estimates")
```



```r
t.test(stdmeandif.all, mu=2, alternative = "greater")
```

```
##
##  One Sample t-test
##
```

```
## data:  stdmeandif.all
## t = 9.8582, df = 100039, p-value < 2.2e-16
## alternative hypothesis: true mean is greater than 2
## 95 percent confidence interval:
##  337.8928      Inf
## sample estimates:
## mean of x
##  405.1613
```

```
quantile(stdmeandif.all, c(0.025, 0.975))
```

```
##        2.5%       97.5%
##    3.590923 1340.896986
```

```
median(stdmeandif.all, na.rm=T)
```

```
## [1] 65.43215
```

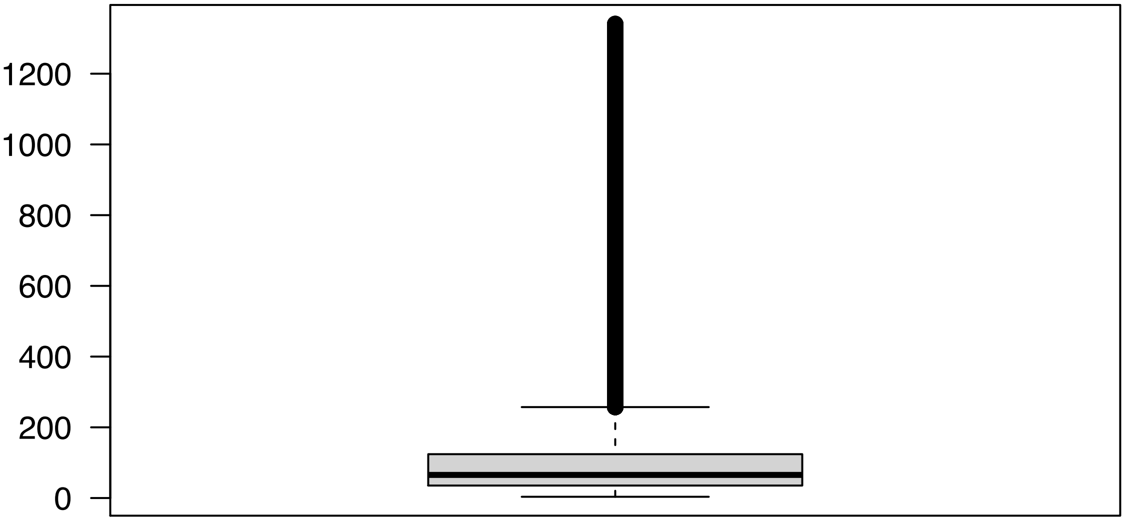
```
mean(stdmeandif.all)
```

```
## [1] 405.1613
```

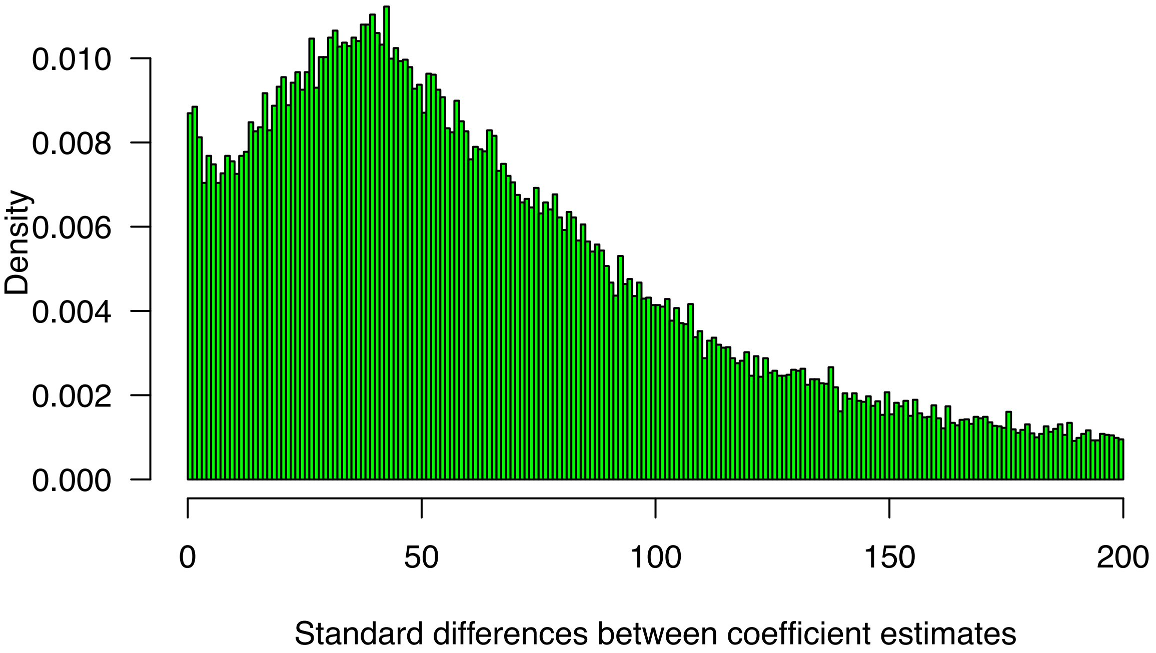
```
sd(stdmeandif.all)
```

```
## [1] 12935.02
```

```
boxplot(stdmeandif.all
        [which(stdmeandif.all> 3.590923 & stdmeandif.all< 1340.896986)],
        las=1)
```



```
hist(stdmeandif.all[which(stdmeandif.all< 200)],
  200, freq=F, las=1, main="", col="green",
  xlab="Standard differences between coefficient estimates")
```



We found that differences between coefficient estimates obtained with the full and partial datasets were very small (on average -0.0014 ± 0.0439 sd), and were not significantly different from zero (95% CI = -0.0352 – 0.0319, $p = 0.8764$, Fig. R5a). We also found that standardised differences were significantly greater than two ($p < 0.001$, 95% CI = 3.591 – 1340.897), being the observed values on average 405 times bigger than actual differences. This indicates that sample removal does not significantly affect the estimation of coefficients and, therefore, its effects are negligible on model performance across all samples, species, and sites. These results further highlight the robustness of our approach and support the reliability of our estimates.

In summary, these analyses show that coefficient estimates are robust to sample removal and reliable as differences between coefficients in full and partial datasets are negligible being very close to zero and relatively very small.

Networks similarity

Next, we explored whether networks resulting from the removal of samples were similar to those in the observed dataset in order to assess the effects of sample removal on network construction. To do so, we assessed the similarity between networks built from species associations using the original Bayesian Poisson model with the complete and partial datasets across all sites. We quantified the similarity between the observed network and each network that resulted from the removal of a sample using the beta-diversity of species interactions approach (Poisot 2016).

```r
netdis <- rep(list(NA), nsites*2)
for(k in 1:nsites){
  nsp <- ncol(data.cush.list[[k]])
  print(k)
```

```
  if(nsp >= 5){
    dimnames(matnet.full.cush.list[[k]]) <-
      rep(list(paste("sp",1:nsp, sep=".")),2)
    gnet.full <- graph.adjacency(t(matnet.full.cush.list[[k]]),
                                  weighted = T)
    if(gsize(gnet.full)>=3){
    for(j in 1:nrow(data.cush.list[[k]])){
      dimnames(matnet.rm.cush.list[[k]][[j]]) <-
        rep(list(paste("sp",1:nsp, sep=".")),2)
      gnet.rm <-
        graph.adjacency(t(matnet.rm.cush.list[[k]][[j]]),
                         weighted = T)
      netdis[[k]] <- c(netdis[[k]],
                       1-betalink(gnet.full,gnet.rm)$WN)
    }}}
  nsp <- ncol(data.open.list[[k]])
  print(k)
  if(nsp >= 5){
    dimnames(matnet.full.open.list[[k]]) <-
      rep(list(paste("sp",1:nsp, sep=".")),2)
    gnet.full <- graph.adjacency(t(matnet.full.open.list[[k]]),
                                  weighted = T)
    if(gsize(gnet.full)>=3){
    for(j in 1:nrow(data.open.list[[k]])){
      dimnames(matnet.rm.open.list[[k]][[j]]) <-
        rep(list(paste("sp",1:nsp, sep=".")),2)
      gnet.rm <-
        graph.adjacency(t(matnet.rm.open.list[[k]][[j]]),
                         weighted = T)
      netdis[[k+nsites]] <- c(netdis[[k+nsites]],
                              1-betalink(gnet.full,gnet.rm)$WN)
    }}}
}

netdis.all <- unlist(netdis)
netdis.all <- netdis.all[-which(is.na(netdis.all))]
```

We then checked the distribution of similarity values between observed and partial networks across all sites.

```
quantile(netdis.all, c(0.025, 0.975))
```

```
##      2.5%     97.5%
## 0.6666667 0.9582160
```

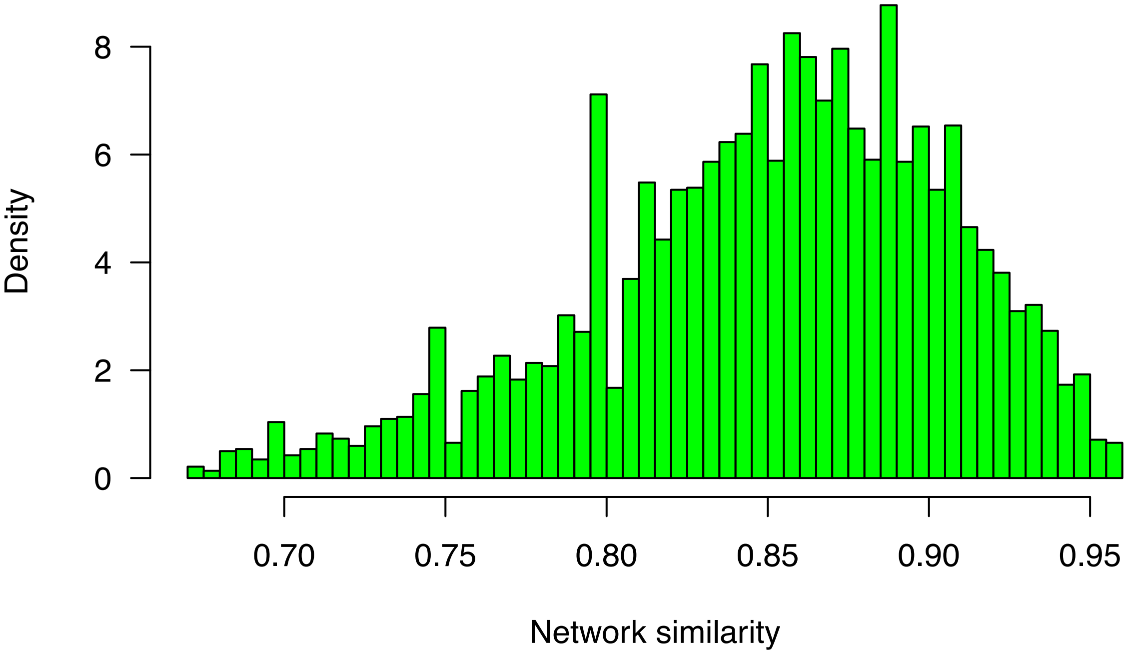
```
mean(netdis.all)
```

```
## [1] 0.8428776
```

```
sd(netdis.all)
```

```
## [1] 0.08278777
```

```
hist(netdis.all
     [which(netdis.all> 0.6666667 & netdis.all< 0.9582160)],
     100, freq=F, las=1, main="", col="green",
     xlab="Network similarity")
```



We found that the similarity between observed and partial networks was very high (mean = 0.84 (0.08 sd)), indicating that networks resulting from the removal of samples were similar to those in the observed dataset (95% c.i. = 0.67 - 0.96) across all sites. In other words, we obtain very similar networks before and after removing samples across all plant communities and sites. This means that excluding samples does not change the structure of resulting networks.

In the end, these results show negligible effects of sample removal on the construction of ecological networks and demonstrate that the final, resulting networks of positive and negative associations among species are robust and reliable.

Changing distribution

In the second analysis, we compare coefficient estimates between the original Bayesian Poisson regression model and a:

1. Bayesian Poisson model with an identity-link function instead of a log-link function

2. Bayesian model with a Negative Binomial distribution instead of a Poisson
3. Template Model Builder (TMB) with a Poisson distribution
4. TMB with a Negative Binomial distribution
5. TMB with a Zero-Inflated Poisson distribution
6. TMB with a Zero-Inflated Negative Binomial distribution
7. TMB with hurdle model

With model 1 we changed the structure of the original Bayesian Poisson model by fitting a linearised form (i.e., the identity-link function), as an alternative to the exponential form. With models 2 and 4 we changed the distribution of the original model by fitting a Negative Binomial, dealing with the issue of data dispersion and sparsity. With models 5 to 7 we also changed the error distribution using Zero-Inflated models as well as separating zero from non-zero counts to deal with underdispersed data (i.e., species with many zeros). To fit these models we subsetted the data to remove those rare species with sparse data that may result in separation. Only species with an occurrence higher than the threshold of four samples were retained in the analysis. We used the data of ecosystem-engineer habitat in a random site (`Cathedral lake`).

```r
mat <- read.table("99CathedralLake2.txt")
mat1 <- mat[-1,-c(1,2)]
for(i in 1:ncol(mat1)){
  mat1[,i] <- as.numeric(as.character(mat1[,i]))
}
# first two columns are useless
mat2 = data.matrix(mat1)
colnames(mat2) = paste("sp",1:ncol(mat2), sep=".")
rownames(mat2)<-NULL
# drop absent species
mat2.cush <- subset(mat2, mat$V2[-1]=="cushion")
mat2.cush <- empty(mat2.cush)
colnames(mat2.cush) = paste("sp",1:ncol(mat2.cush), sep=".")
# remove rare species present in four or less samples
mat2.cush.bin <- ifelse(mat2.cush>0,1,0)
mat2.cush.a <- mat2.cush[,which(colSums(mat2.cush.bin)>4)]

# fitting different models
## poisson ##
coef.a.bayes <- matrix(0, ncol=ncol(mat2.cush.a), nrow=ncol(mat2.cush.a))
for(i in 1:ncol(mat2.cush.a)){
  mod.full <- bayesglm(mat2.cush.a[,i]~mat2.cush.a[,-i],
                       family="poisson")
  coef.a.bayes[-i,i] <- coef(mod.full)[-1]
  fmlamod <- as.formula(paste(paste(colnames(mat2.cush.a)[i], "~"),
```

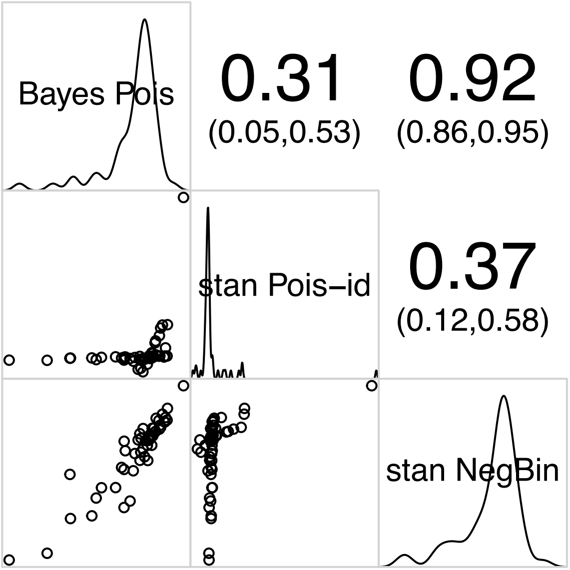
```r
                              paste(colnames(mat2.cush.a)[-i],
                              collapse="+")))
}

## identity link function ##
coef.a.stan2 <- matrix(0, ncol=ncol(mat2.cush.a), nrow=ncol(mat2.cush.a))
for(i in 1:ncol(mat2.cush.a)){
  fmlamod <- as.formula(paste(paste(colnames(mat2.cush.a)[i], "~"),
                              paste(colnames(mat2.cush.a)[-i],
                              collapse="+")))
  mod.full <- stan_glm(fmlamod, family=poisson(link="identity"),
                       data = data.frame(mat2.cush.a),
                       cores=numCores)
  coef.a.stan2[-i,i] <- coef(mod.full)[-1]

## negative binomial
coef.a.stannegb <- matrix(0, ncol=ncol(mat2.cush.a),
                             nrow=ncol(mat2.cush.a))

for(i in 1:ncol(mat2.cush.a)){
  fmlamod <- as.formula(paste(paste(colnames(mat2.cush.a)[i], "~"),
                              paste(colnames(mat2.cush.a)[-i],
                              collapse="+")))
  mod.full <- stan_glm.nb(fmlamod, data = data.frame(mat2.cush.a),
                          iter=5000, chains=4, cores=numCores)
  coef.a.stannegb[-i,i] <- coef(mod.full)[-1]
}

## zero-inflated models ##
coef.a.tmb.pois   <- matrix(0, ncol=ncol(mat2.cush.a),
                               nrow=ncol(mat2.cush.a))
coef.a.tmb.nbin2  <- matrix(0, ncol=ncol(mat2.cush.a),
                               nrow=ncol(mat2.cush.a))
coef.a.tmb.zipois <- matrix(0, ncol=ncol(mat2.cush.a),
                               nrow=ncol(mat2.cush.a))
coef.a.tmb.zinb2  <- matrix(0, ncol=ncol(mat2.cush.a),
                               nrow=ncol(mat2.cush.a))
coef.a.tmb.zitrnb <- matrix(0, ncol=ncol(mat2.cush.a),
                               nrow=ncol(mat2.cush.a))

for(i in 1:ncol(mat2.cush.a)){
  fmlamod <- as.formula(paste(paste(colnames(mat2.cush.a)[i], "~"),
                        paste(colnames(mat2.cush.a)[-i],
                        collapse="+")))
```

```
  # TMB Poisson
  mod.full <- glmmTMB(fmlamod, data = data.frame(mat2.cush.a),
                      family = poisson)
  coef.a.tmb.pois[i,-i] <- fixef(mod.full)$cond[-1]
  # TMB Negative Binomial
  mod.full <- glmmTMB(fmlamod, data = data.frame(mat2.cush.a),
                      family = nbinom2)
  coef.a.tmb.nbin2[i,-i]  <- fixef(mod.full)$cond[-1]
  # TMB Zero-Inflated Poisson
  mod.full <- glmmTMB(fmlamod, data = data.frame(mat2.cush.a),
                      family = poisson, ziformula = ~.)
  coef.a.tmb.zipois[i,-i]  <- fixef(mod.full)$cond[-1]
  # TMB Zero-Inflated Negative Binomial
  mod.full <- glmmTMB(fmlamod, data = data.frame(mat2.cush.a),
                      family = nbinom2, ziformula = ~.)
  coef.a.tmb.zinb2[i,-i]   <- fixef(mod.full)$cond[-1]
  # TMB Hurdle
  mod.full <- glmmTMB(fmlamod, data = data.frame(mat2.cush.a),
                      family = truncated_nbinom1,
                      ziformula = ~.)
  coef.a.tmb.zitrnb1[i,-i]  <- fixef(mod.full)$cond[-1]

# we assemble all of them in two dataframes
# we exclude the main diagonal as we don't have alpha_ii
cordf1 <- data.frame(cbind(
  as.vector(coef.a.bayes[which(coef.a.bayes!=0)]),
  as.vector(coef.a.stan2[which(coef.a.bayes!=0)]),
  as.vector(coef.a.stannegb[which(coef.a.bayes!=0)])
  ))

colnames(cordf1) <- c(
  "Bayes Pois",
  "stan Pois-id",
  "stan NegBin"
  )

cordf2 <- data.frame(cbind(
  as.vector(coef.a.tmb.pois[which(coef.a.bayes!=0)]),
  as.vector(coef.a.tmb.nbin2[which(coef.a.bayes!=0)]),
  as.vector(coef.a.tmb.zipois[which(coef.a.bayes!=0)]),
  as.vector(coef.a.tmb.zinb2[which(coef.a.bayes!=0)]),
  as.vector(coef.a.tmb.zitrnb1[which(coef.a.bayes!=0)])
  ))
```

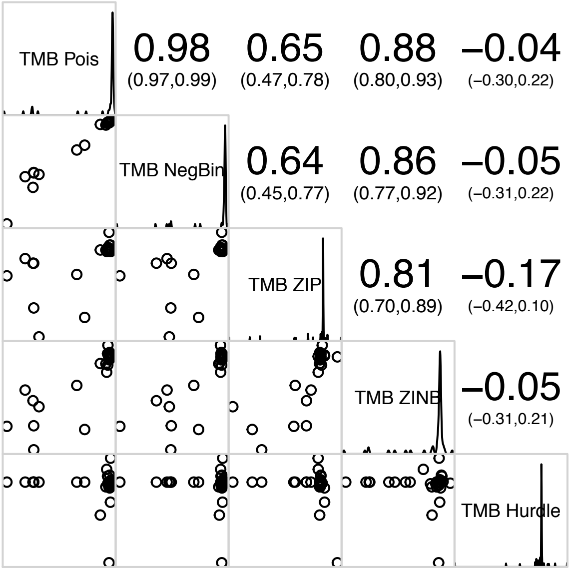
```
colnames(cordf1) <- c(
  "TMB Pois",
  "TMB NegBin",
  "TMB ZIP",
  "TMB ZINB",
  "TMB Hurdle"
  )
```

We then tested the correlations among coefficients across all eight models. We used Pearson correlation test and report results of correlation analysis with confidence intervals in the following figure.

```
corrgram(cordf1,
         lower.panel = panel.pts,
         upper.panel=panel.conf,
         diag.panel=panel.density,
         col="blue")
```



```
corrgram(cordf2,
         lower.panel = panel.pts,
         upper.panel=panel.conf,
         diag.panel=panel.density,
         col="blue")
```



The above figure shows a correlation matrix of coefficient estimates across the eight regression models. On the main diagonal, the different models. The first is the original Bayesian Poisson model, the others are explained in the main above. In the upper panels, the Pearson correlation between estimates in the different models; in red the significant correlations; important comparisons are highlighted with bigger font size. In the lower panels, the data points are coefficient estimates.

Results indicate that all but model 7 (TMB hurdle) and partially model 1 (Bayesian Poisson identity-link) produce very consistent and similar outcomes. Specifically, the original Bayesian Poisson model yield highly significant correlations of coefficient estimates with all the other models, except the TMB hurdle model. Very high correlations were also observed between the original model and a Bayesian Negative Binomial (cor = 0.91, $p < 0.001$), a TMB Poisson (cor = 0.72, $p < 0.001$), a TMB Negative Binomial (cor = 0.70, $p < 0.001$), and a TMB Zero-Inflated Negative Binomial (cor = 0.77, $p < 0.001$). Notably, using a linear form (i.e., identity-link function) instead of an exponential one (i.e., log-link function) produced significantly correlated estimates (cor = 0.31, $p = 0.021$). Yet, such linear form poorly correlated with all the other models, contrary to the original exponential one. It is also worth noticing the very high correlations between a TMB model with Poisson distribution and a Negative Binomial (cor = 0.98, $p < 0.001$), a Zero-Inflated Poisson (cor = 0.65, $p < 0.001$), and a Zero-Inflated Negative Binomial (cor = 0.88, $p < 0.001$).

In summary, these results indicate that Poisson and Negative Binomial distributions produce equivalent and very consistent coefficient estimates. Bayesian models with Poisson or Negative Binomial distribution are the most appropriate for the purpose of this study than the TMB for their statistical plausibility and versatility. We highlight that changing model structure and distribution has little impact on coefficient estimates, and the results therein led to no change in the qualitative conclusions of our study. Therefore, with this analysis we have addressed the concerns of the Referee regarding model structure and solved the issues of sparse data, separation, and dispersion.

Removing rare species

In this third analysis, we test the robustness of our original Bayesian Poisson model against the removal of rare species (i.e., species that occur in four or less samples across the dataset) and, therefore, check that excluding rare species has no consequences for estimating coefficients among the remaining species. We compare the coefficients of the non-rare species estimated for both a full dataset (i.e., with rare species) and a dataset without rare species. We do so for two equivalent Bayesian Poisson models (i.e., `bayesglm` of `arm` as well as `stan_glm` of `rstanarm`).

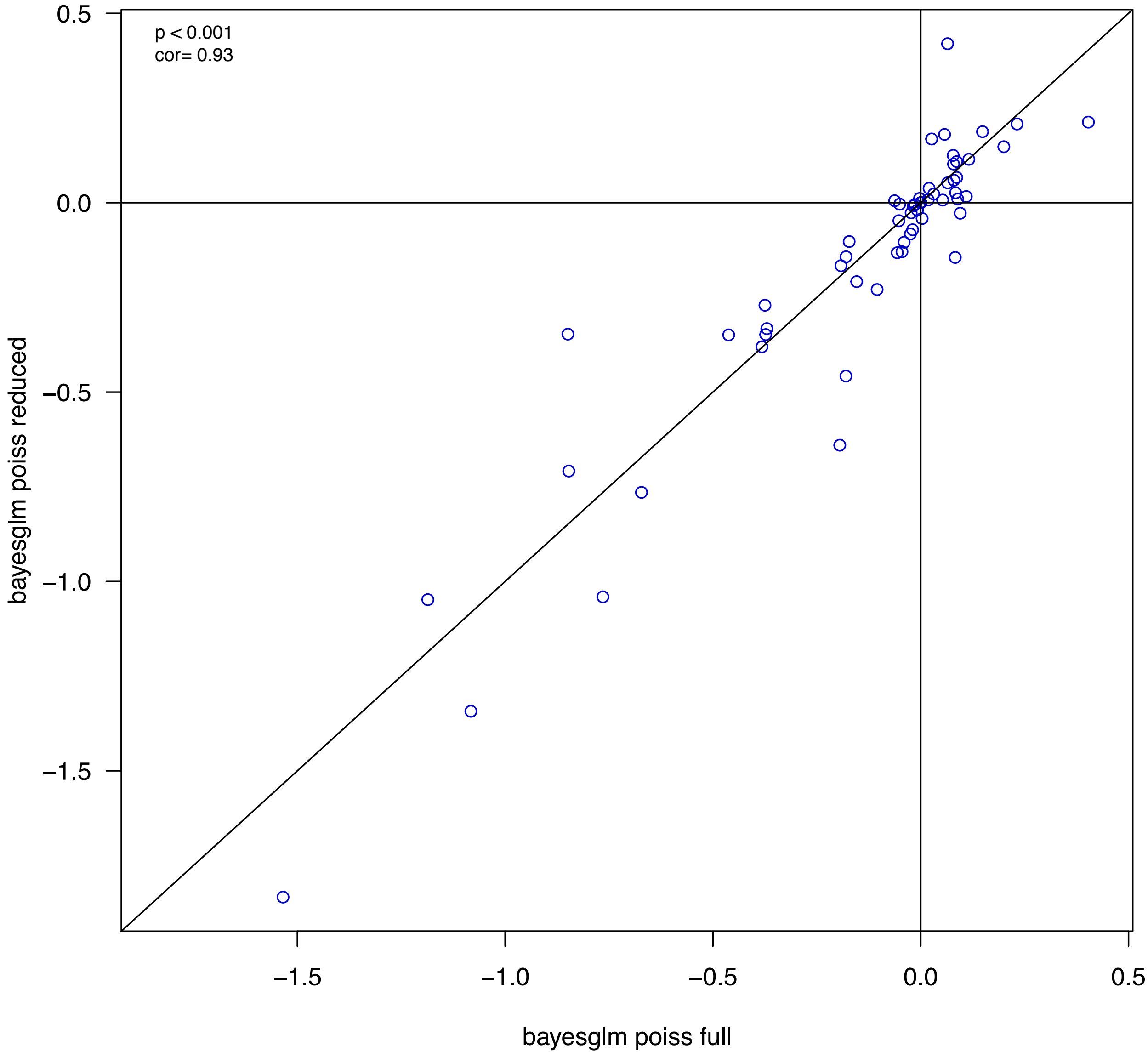
```
# coefficients in full dataset
coef.full         <- matrix(0,
                       ncol=ncol(mat2.cush), nrow=ncol(mat2.cush))
coef.full.stanglm <- matrix(0,
                       ncol=ncol(mat2.cush), nrow=ncol(mat2.cush))

for(i in 1:ncol(mat2.cush)){
  # bayesglm
  mod.full <- bayesglm(mat2.cush[,i]~mat2.cush[,-i],
                        family="poisson")
  coef.full[-i,i] <- coef(mod.full)[-1]
  # stanglm
  fmlamod <- as.formula(paste(paste(colnames(mat2.cush)[i], "~"),
                               paste(colnames(mat2.cush)[-i],
                               collapse="+")))
  mod.full <- stan_glm(fmlamod, family=poisson,
                        data = data.frame(mat2.cush))
  coef.full.stanglm[-i,i] <- coef(mod.full)[-1]
}

# we make a new matrix for the coefficients of
# non-rare species estimated in full dataset
# by removing rare species from the coefficient matrix

coef.r.bayes <- coef.full[which(colSums(mat2.cush.bin)>4),
                          which(colSums(mat2.cush.bin)>4)]
coef.r.stanglm <- coef.full.stanglm[which(colSums(mat2.cush.bin)>4),
                          which(colSums(mat2.cush.bin)>4)]

# coefficients estimated after removing species for analysis 2
# are in the following matrices
coef.a.bayes
coef.a.stan
```

We run pearson correlation tests for species coefficients estimated with the full data and those estimated with partial data without rare species (y-axis) for the two Bayesian Poisson models} are reported at the top-left of the following plots.

```
# bayesglm
corr <- cor.test(coef.r.bayes,coef.a.bayes,
                 "pearson", alternative = "two.sided")
print(corr)
```

```
##
##  Pearson's product-moment correlation
##
## data:  coef.r.bayes and coef.a.bayes
## t = 20.521, df = 62, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.8926864 0.9592870
## sample estimates:
##       cor
## 0.9336289
```

```
plot(coef.r.bayes,coef.a.bayes, las=1, col="blue3",
     xlab="bayesglm poiss full",
     ylab = "bayesglm poiss reduced",
     xlim = range(c(coef.r.bayes,coef.a.bayes)),
     ylim = range(c(coef.r.bayes,coef.a.bayes)))
abline(c(0, 1))
abline(h=0)
abline(v=0)
legend("topleft", legend=c("p < 0.001",
       paste("cor=", round(corr$estimate,2))),
       bty="n", inset=0, cex=.75)
```

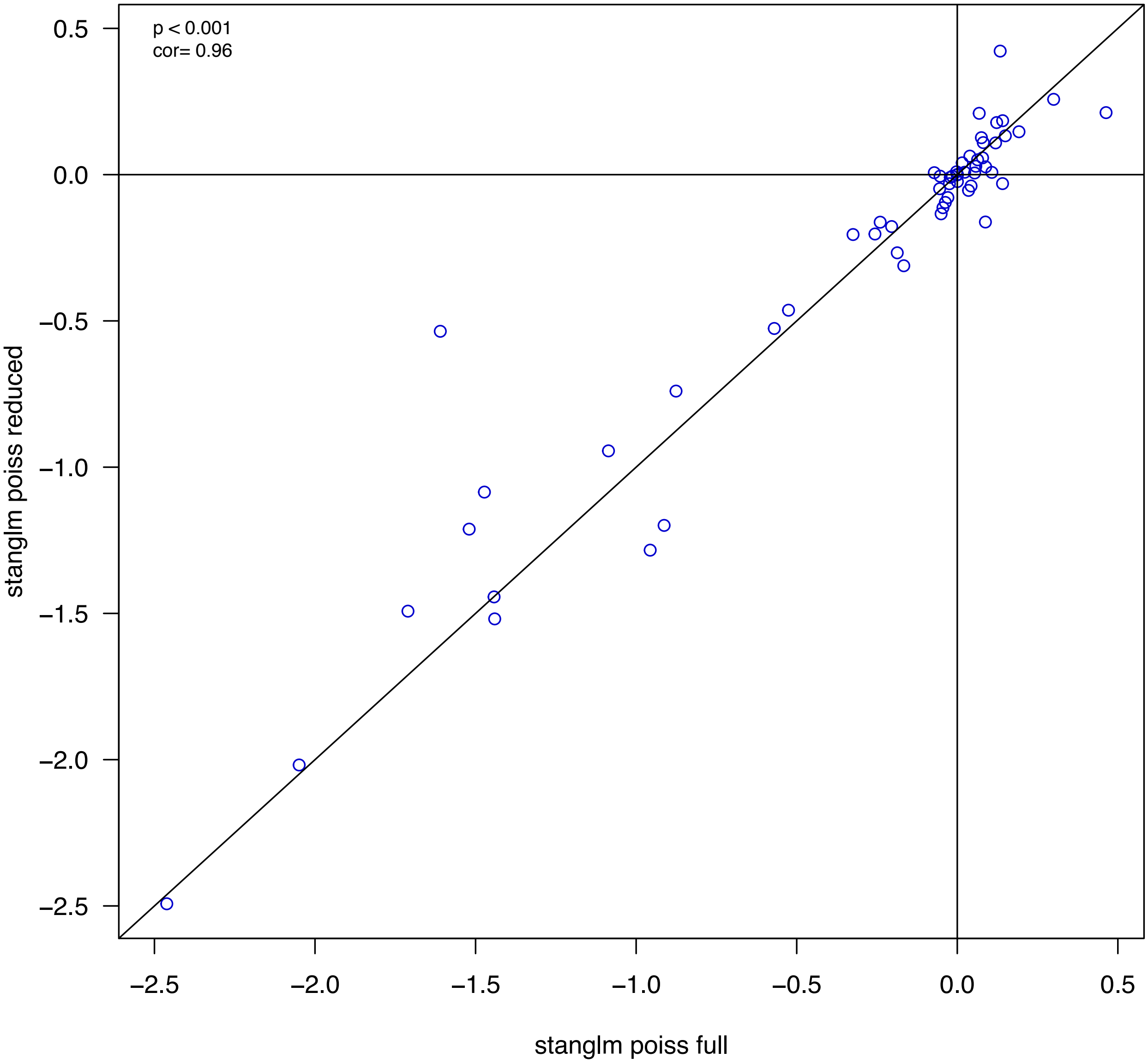


```
# stanglm
corr <- cor.test(coef.r.stanglm,coef.a.stan, "pearson",
                 alternative = "two.sided")
print(corr)
```

```
##
##  Pearson's product-moment correlation
##
## data:  coef.r.stanglm and coef.a.stan
## t = 26.24, df = 62, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.9312485 0.9742425
## sample estimates:
##       cor
```

```
## 0.9578067
```

```
plot(coef.r.stanglm,coef.a.stan, las=1, col="blue3",
     xlab="stanglm poiss full",
     ylab = "stanglm poiss reduced",
     xlim = range(c(coef.r.stanglm,coef.a.stan)),
     ylim = range(c(coef.r.stanglm,coef.a.stan)))
abline(c(0, 1))
abline(h=0)
abline(v=0)
legend("topleft", legend=c("p < 0.001",
       paste("cor=", round(corr$estimate,2))),
       bty="n", inset=0, cex=.75)
```



Results show that coefficient estimates are highly significantly and positively correlated in the presence and absence of rare species in the dataset. Indeed, estimated coefficients do not significantly change when removing rare species that contribute to data sparsity and separation. These results prove that the removal of rare species had no consequences on the estimation of coefficients for the remaining non-rare species, and further demonstrate both the robustness of our original Bayesian Poisson approach, in this case against removing species.

Changing credible interval cutoff

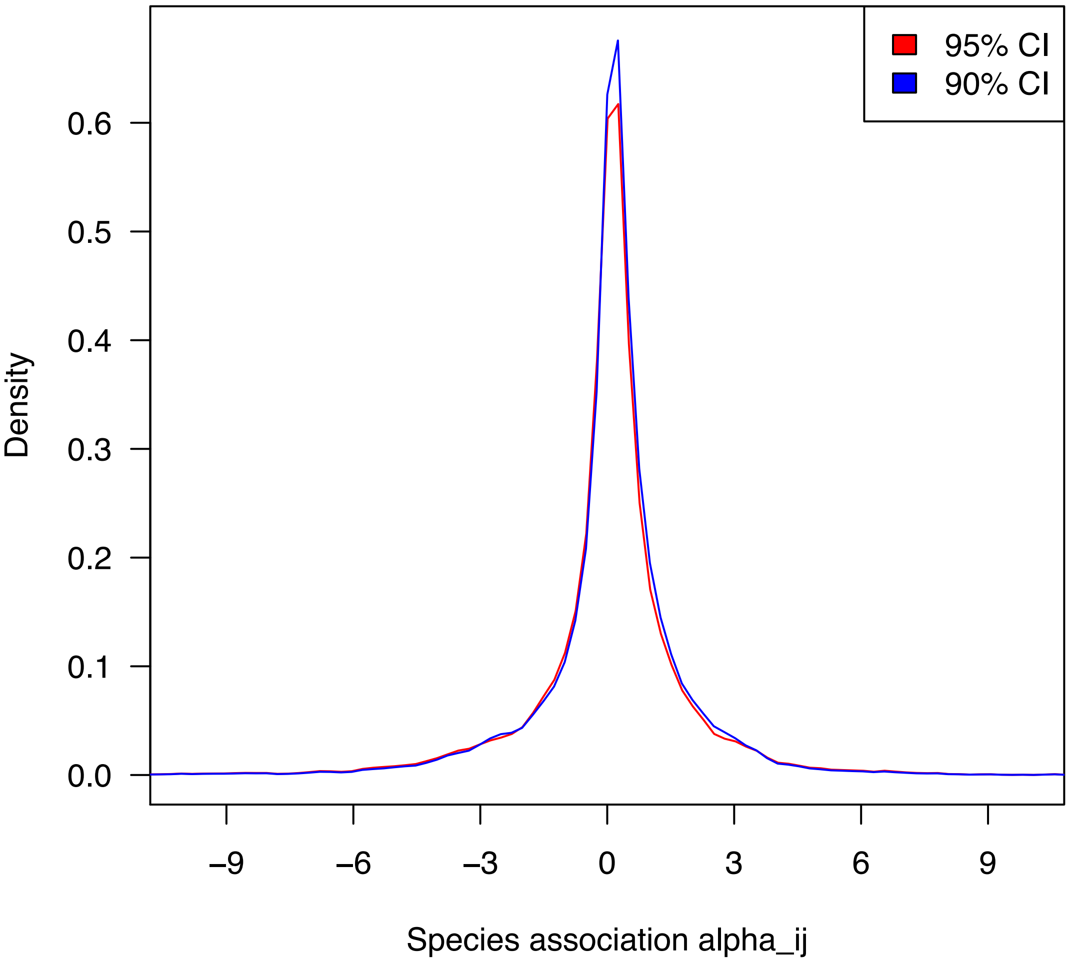
With this fourth analysis, we test the effects of changing the CI cutoff from 95% to 90%. We therefore analyzed our data using a 90% CI. Then, we compared these new results with previous ones.

```r
## function for inferring species associations with 90% CI
statmod90 <- function(dataj, datai, j, nsp){
    mod<-bayesglm(dataj~datai, family="poisson")
    estimcoef <- coef(mod)[-1]
    posterior <- coef(sim(mod))[,-1]
    for(k in 1:(nsp-1)){
        if(sign(quantile(posterior[,k],0.05))!=
           sign(quantile(posterior[,k],0.95)))
        estimcoef[k] <- 0
    }
    cicoef <- rep(0,nsp)
    cicoef[-j] <- estimcoef
    return(cicoef)
}

## cushion
coef.cush_o90 <- foreach(zz=1:nsites, .inorder = TRUE) %dopar% {
    nsp <- ncol(data.cush.list[[zz]])
    if(nsp >=5){
        coef.site <- foreach(j=1:nsp, .combine=rbind, .inorder = TRUE)
        %dopar%{statmod90(data.cush.list[[zz]][,j],
                          data.cush.list[[zz]][,-j], j, nsp)
        }}
    else NA
}

## open
coef.open_o90 <- foreach(zz=1:nsites, .inorder = TRUE) %dopar% {
    nsp <- ncol(data.open.list[[zz]])
    if(nsp >=5){
        coef.site <- foreach(j=1:nsp, .combine=rbind, .inorder = TRUE)
        %dopar%{statmod90(data.open.list[[zz]][,j],
```

```
                           data.open.list[[zz]][,-j], j, nsp)
        }}
    else NA
}

# same names
for(kk in 1:nsites){
    if(sites.df$nsp[kk]>=5)
      dimnames(coef.cush_o90[[kk]]) <-
        rep(list(colnames(data.cush.list[[kk]])),2)
    if(sites.df$nsp[kk+nsites]>=5)
      dimnames(coef.open_o90[[kk]]) <-
        rep(list(colnames(data.open.list[[kk]])),2)
}

aij_o90 <- rep(rep(list(NA),nsites),2)
for(zz in 1:nsites){
    aij_o90[[zz]]         <-
      coef.cush_o90[[zz]][which(coef.cush_o90[[zz]]!=0)]
    aij_o90[[zz+nsites]]  <-
      coef.open_o90[[zz]][which(coef.open_o90[[zz]]!=0)]
}
```

First, we compare the distribution of species associations α_{ij} between 95% CI and 90% CI networks.

```
plot(0,0,type="n",xlim=c(-10,10), ylim=c(0,0.68),yaxt="n",
     xaxt="n",xlab="Species association alpha_ij",ylab="Density")
axis(1,at=c(-9,-6,-3,0,3,6,9), las=1)
axis(2,at=c(0,0.1,0.2,0.3,0.4,0.5,0.6), las=1)
lines(density(unlist(aij_o), adjust=1), col='red')
lines(density(unlist(aij_o90), adjust=1), col='blue')
legend('topright', fill = c('red', 'blue'),
       legend = c('95% CI', '90% CI'))
```



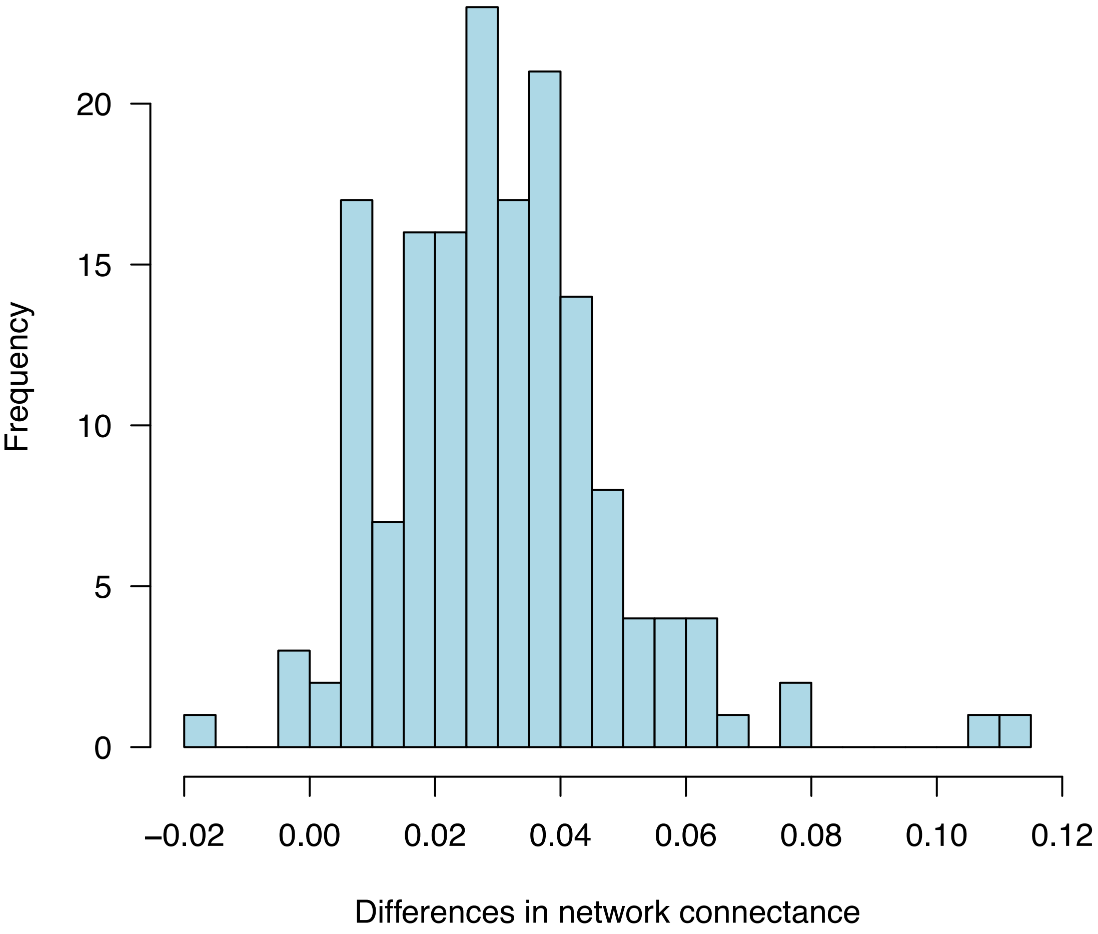
Results indicate that the distribution of inferred species interactions α_{ij} does not change with changing the cutoff.

Second, we compare the overall network structure, and in particular we calculate differences in network connectance.

```
gnet_o90     <- rep(rep(list(NA),nsites),2)
names(gnet_o90)<-c(names(coef.cush_o),names(coef.open_o))
for(i in 1:nsites){
    if(sites.df$nsp[i]>=5){
        # cushion
        gnet_o90[[i]] <-
          graph.adjacency(t(coef.cush_o90[[i]]), weighted=TRUE)
        # open
        gnet_o90[[i+nsites]] <-
          graph.adjacency(t(coef.open_o90[[i]]), weighted=TRUE)
}}
dconn = NA
for(i in 1:(nsites*2)){
    if(sites.df$nsp[i]>=5){
```

```
        dconn = c(dconn,
                  length(E(gnet_o90[[i]]))/length(V(gnet_o90[[i]]))^2 -
                  length(E(gnet_o[[i]]))/length(V(gnet_o[[i]]))^2)
    }}
dconn = dconn[-1]
```

```
hist(dconn, 40, las = 1, col = 'lightblue', main='',
     xlab = 'Differences in network connectance')
```



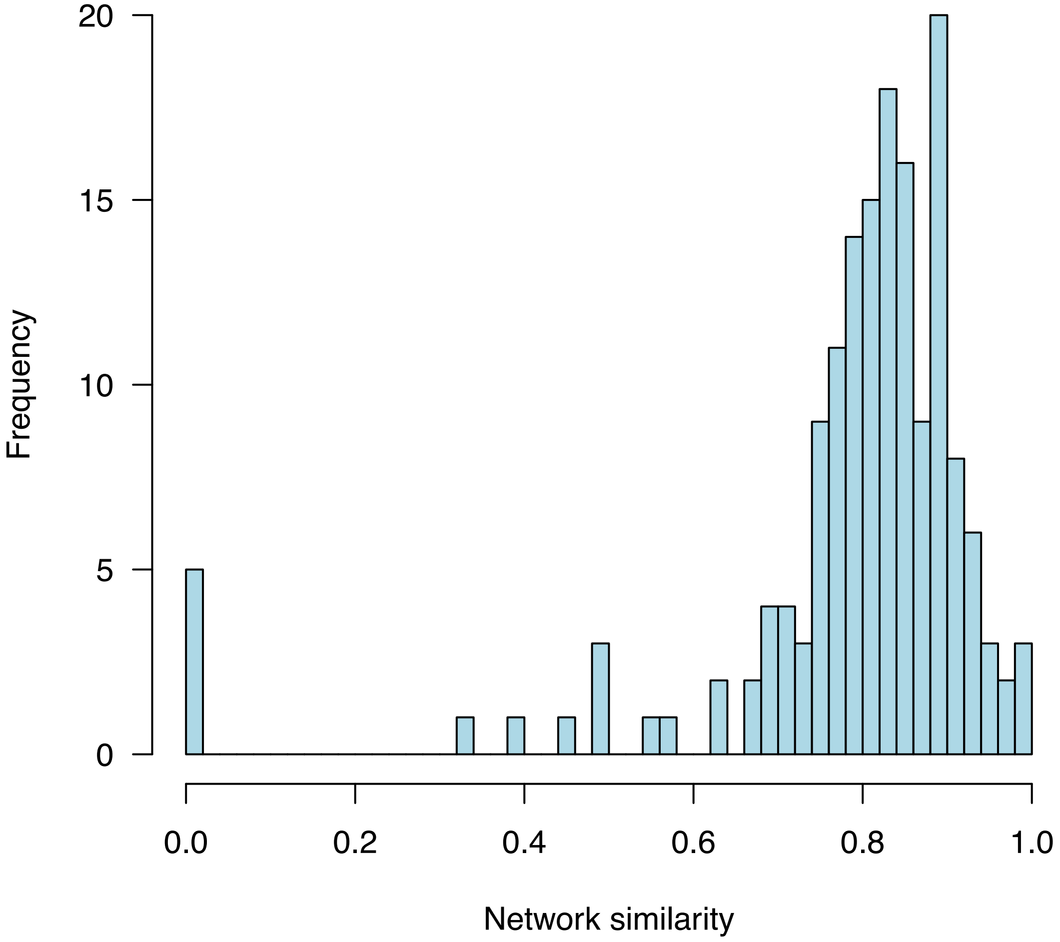
Results show that differences in network connectance are negligible, indicating that the overall network architecture does not change with changing the CI cutoff.

Third, we compare the species-specific patterns of associations (i.e., who is associated to whom) by looking at network similarity.

```
netdis <- NA
for(i in 1:(nsites*2)){
    if(sites.df$nsp[i]>=5){
        print(i)
        netdis = c(netdis,
                  1-betalink(gnet_o[[i]],gnet_o90[[i]])$WN)
```

```
}}
netdis = netdis[-1]
```

```
hist(netdis, 40, las = 1, col = 'lightblue',
     xlab = 'Network similarity', main='')
```



Results show that the 95% CI and 90% CI networks are also similar in terms of specific interactions patterns, indicating that the associations between species pairs do not change either with changing the cutoff.

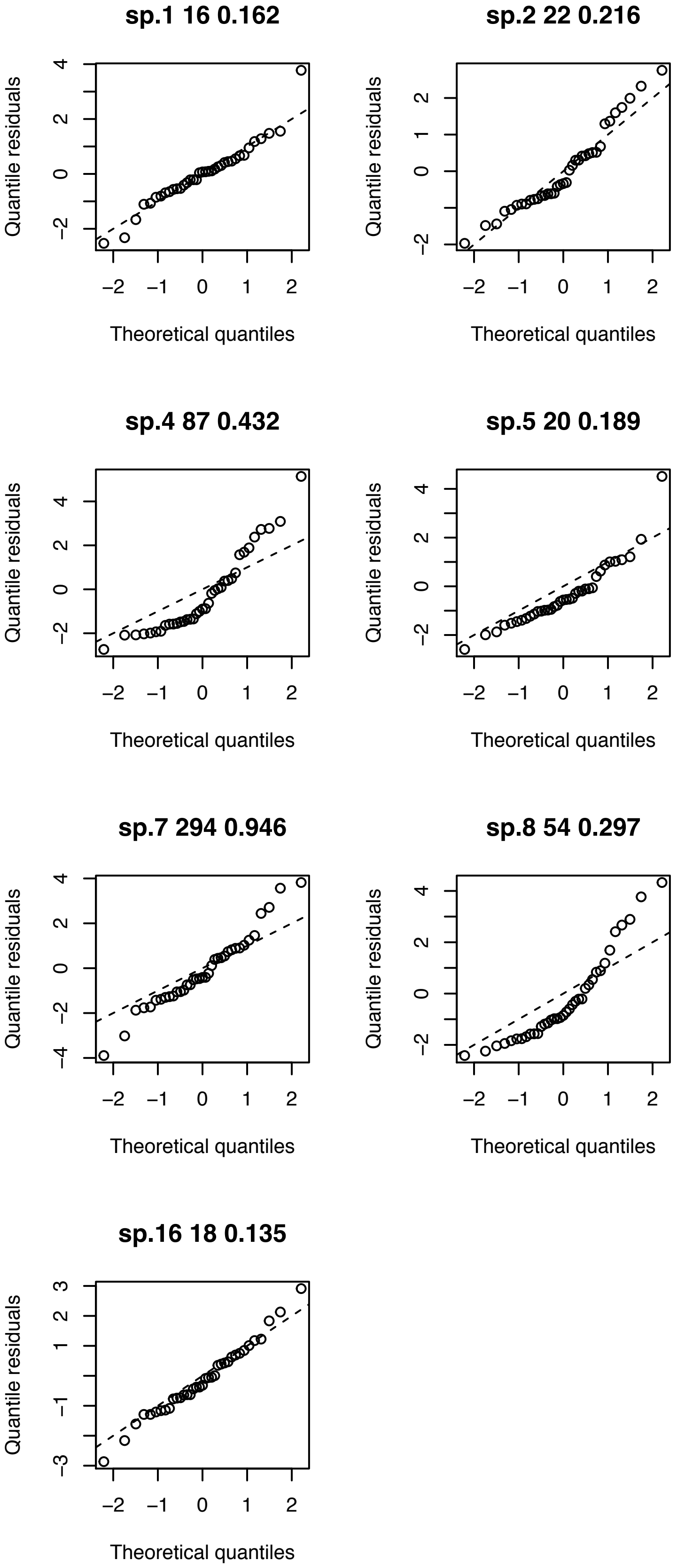
Diagnostics

Q-Q plots

In order to assess the quality of fitted Poisson models, we evaluate the Q-Q plot on the same plant community.

```
# for the full matrix
par(mfrow= c(4,4))
for(i in 1:ncol(mat2.cush)){
mod.full <- bayesglm(mat2.cush[,i]~mat2.cush[,-i], family="poisson")
}
```

```
# for the reduced matrix
par(mfrow= c(4,2))
for(i in 1:8){
mod.full <- bayesglm(mat2.cush.a[,i]~mat2.cush.a[,-i], family="poisson")
if(i!=7)
qqrplot(mod.full, main = paste(colnames(mat2.cush.a)[i],
                               colSums(mat2.cush.a)[i],
        round(length(which(mat2.cush.a[,i]!=0))/nrow(mat2.cush.a), 3)))
}
```



Results indicate that plant abundance data plausibly reproduce a theoretical Poisson distribution, indicating that the assumption of a Poisson distribution in models of species associations is not violated but is a reasonable choice.

We finally highlight two points. First, despite the removal of rare species is justified for statistical reasons (i.e., removing rare species removes warning messages, while its effects is negligible on the estimation of coefficients for the other non-rare species), it is ecologically important to consider rare species anyway. Rare species are a major component of biodiversity and ecosystems, and are the most vulnerable to extinction risk. Thus, maintaining rare species in the analysis comes at no statistical cost, while it substantially increases the ecological foundations and the biological information of the study.

Second, we specify that the original model for the whole database with rare species does not lead us to include potential spurious associations among species in the network. Most critically, estimation of associations for rare species with warning messages are never going to be significant enough to be included in the final networks. Indeed, the sparsity of rare species data makes the Bayesian Credible Interval rather wider than sharpened, thus the resulting species associations are rather excluded than included.

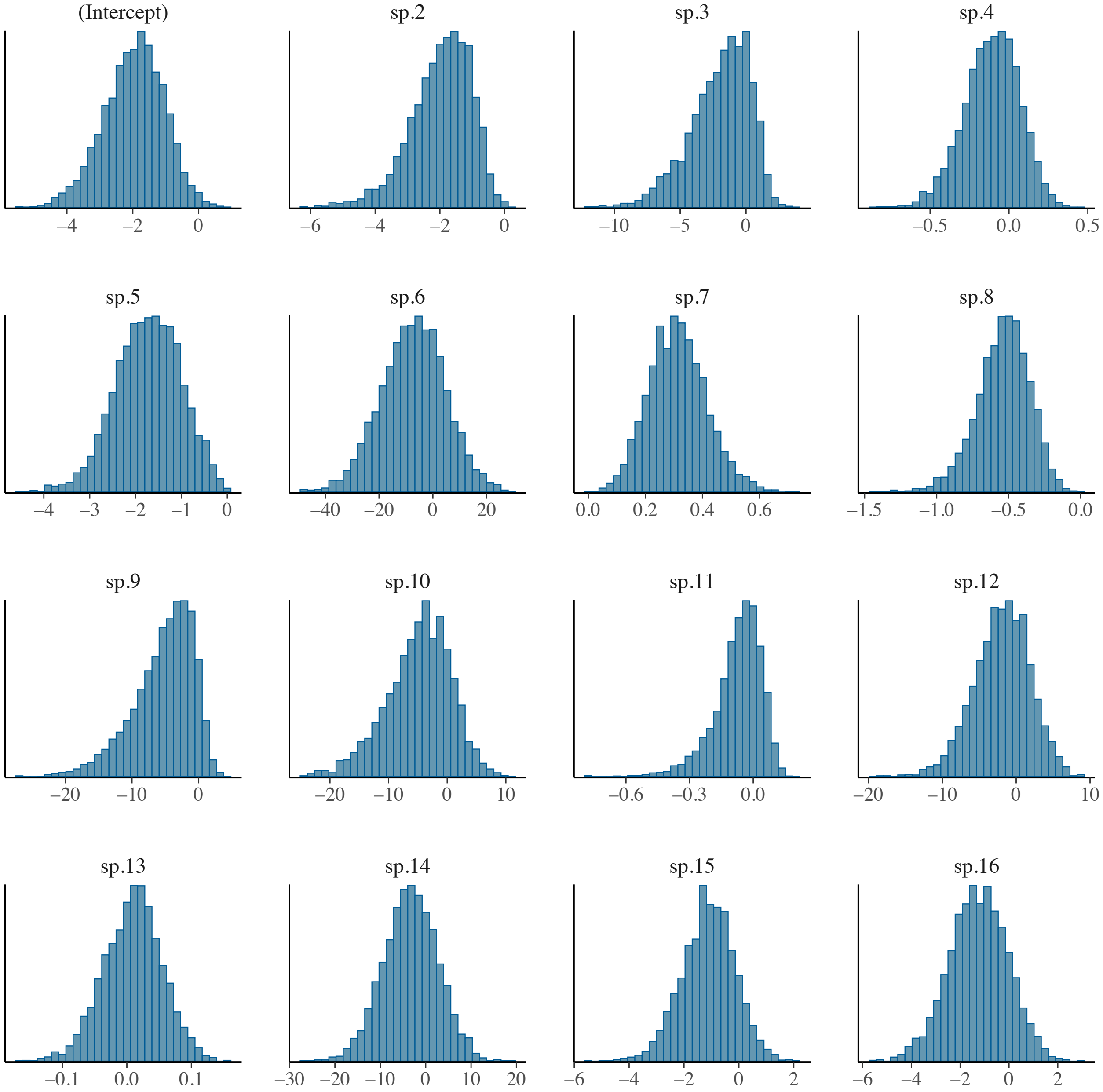
In conclusion, keeping rare species in the model is not only more ecological meaningful but also more statistically conservative. At the end of the day, the Bayesian Poisson model is demonstrated to correctly and accurately infer species associations.

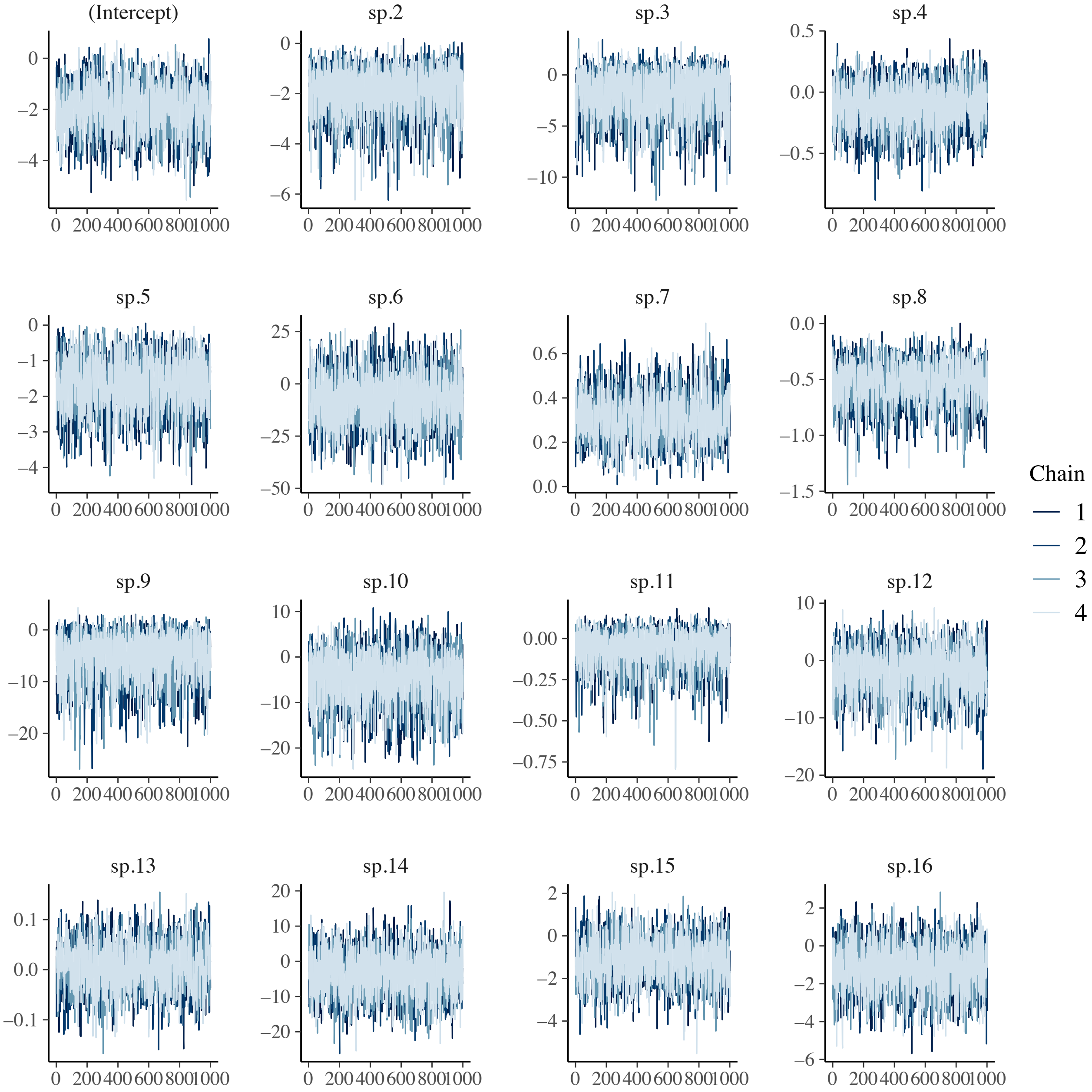
Parameter estimates from MCMC

We carry out visual diagnostic by plotting and inspecting parameter estimates from MCMC draws. First, we look at histograms of marginal posterior distributions, with all Markov chains combined. Second, we look at trace plots, i.e., time series plots of Markov chains.

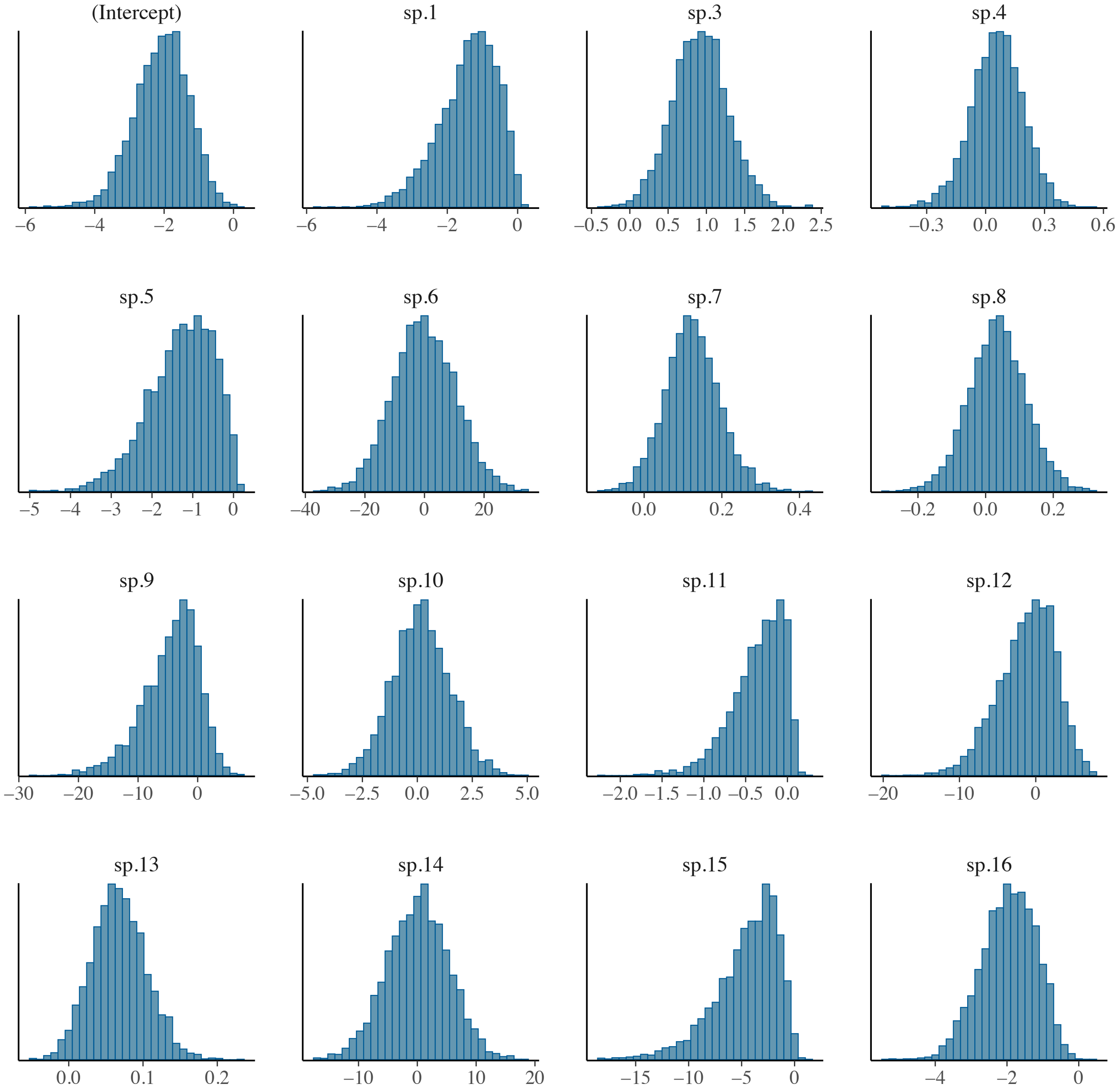
```r
for(i in 1:ncol(mat2.cush)){
    fmlamod <- as.formula(paste(paste(colnames(mat2.cush)[i], "~"),
                                paste(colnames(mat2.cush)[-i], collapse="+")))
  mod.full <- stan_glm(fmlamod, family=poisson,
                       data = data.frame(mat2.cush), cores=numCores)
  posterior <- as.array(mod.full)
    print(mcmc_hist(posterior))
    print(mcmc_trace(posterior))
}
```

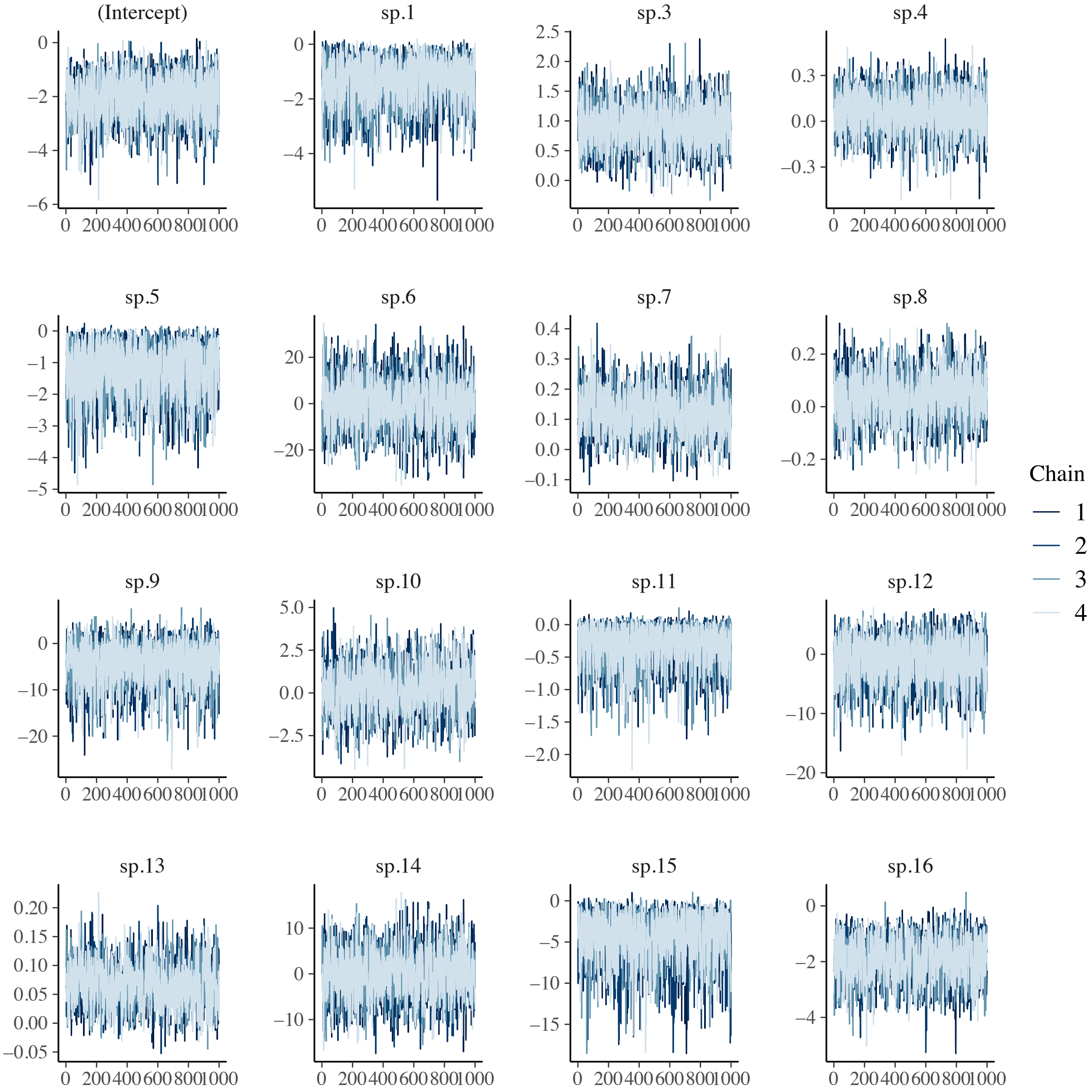
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```


(Intercept)
−4 −2 0
sp.2
−6 −4 −2 0
sp.3
−10 −5 0
sp.4
−0.5 0.0 0.5
sp.5
−4 −3 −2 −1 0
sp.6
−40 −20 0 20
sp.7
0.0 0.2 0.4 0.6
sp.8
−1.5 −1.0 −0.5 0.0
sp.9
−20 −10 0
sp.10
−20 −10 0 10
sp.11
−0.6 −0.3 0.0
sp.12
−20 −10 0 10
sp.13
−0.1 0.0 0.1
sp.14
−30 −20 −10 0 10 20
sp.15
−6 −4 −2 0 2
sp.16
−6 −4 −2 0 2

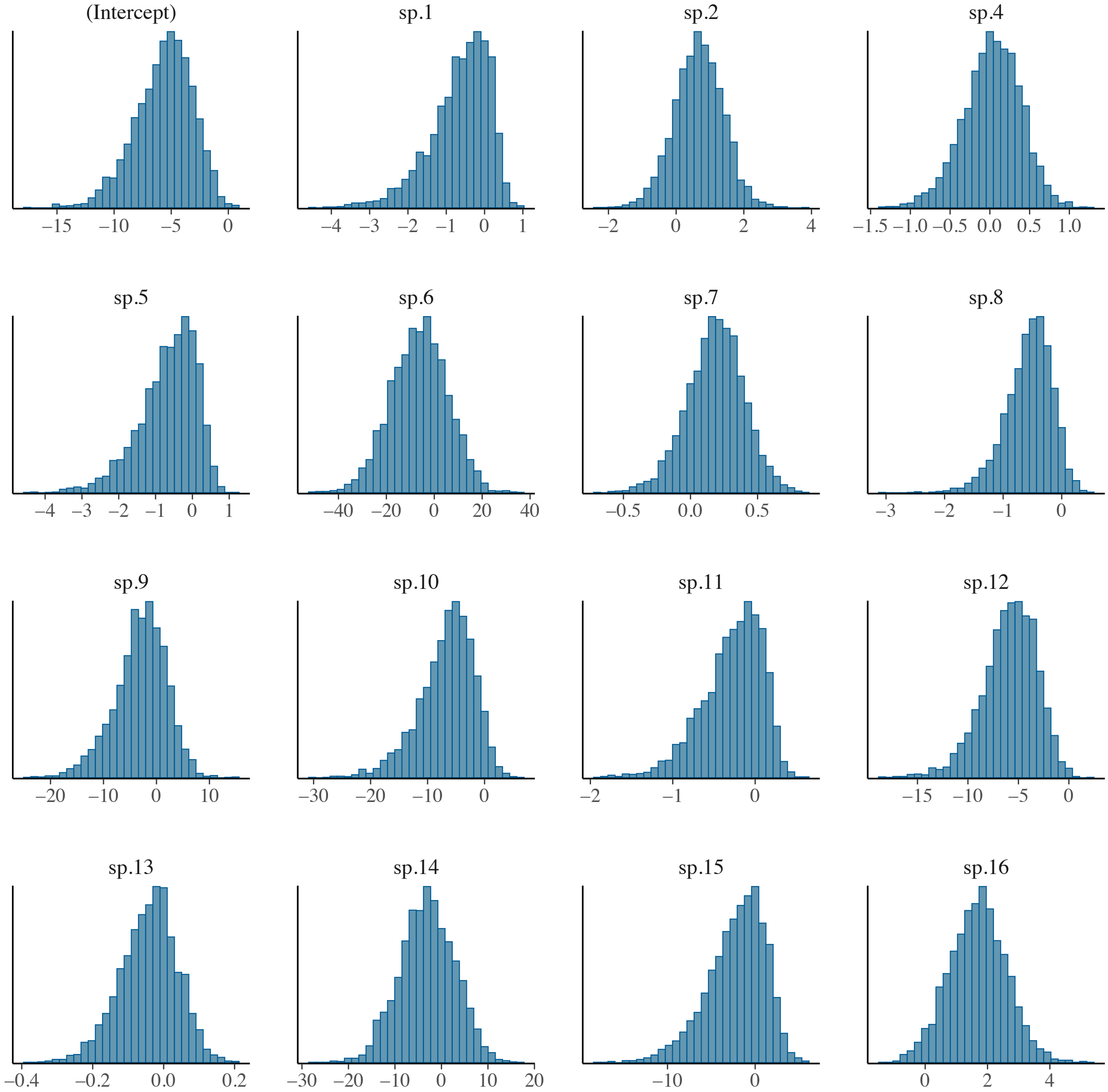


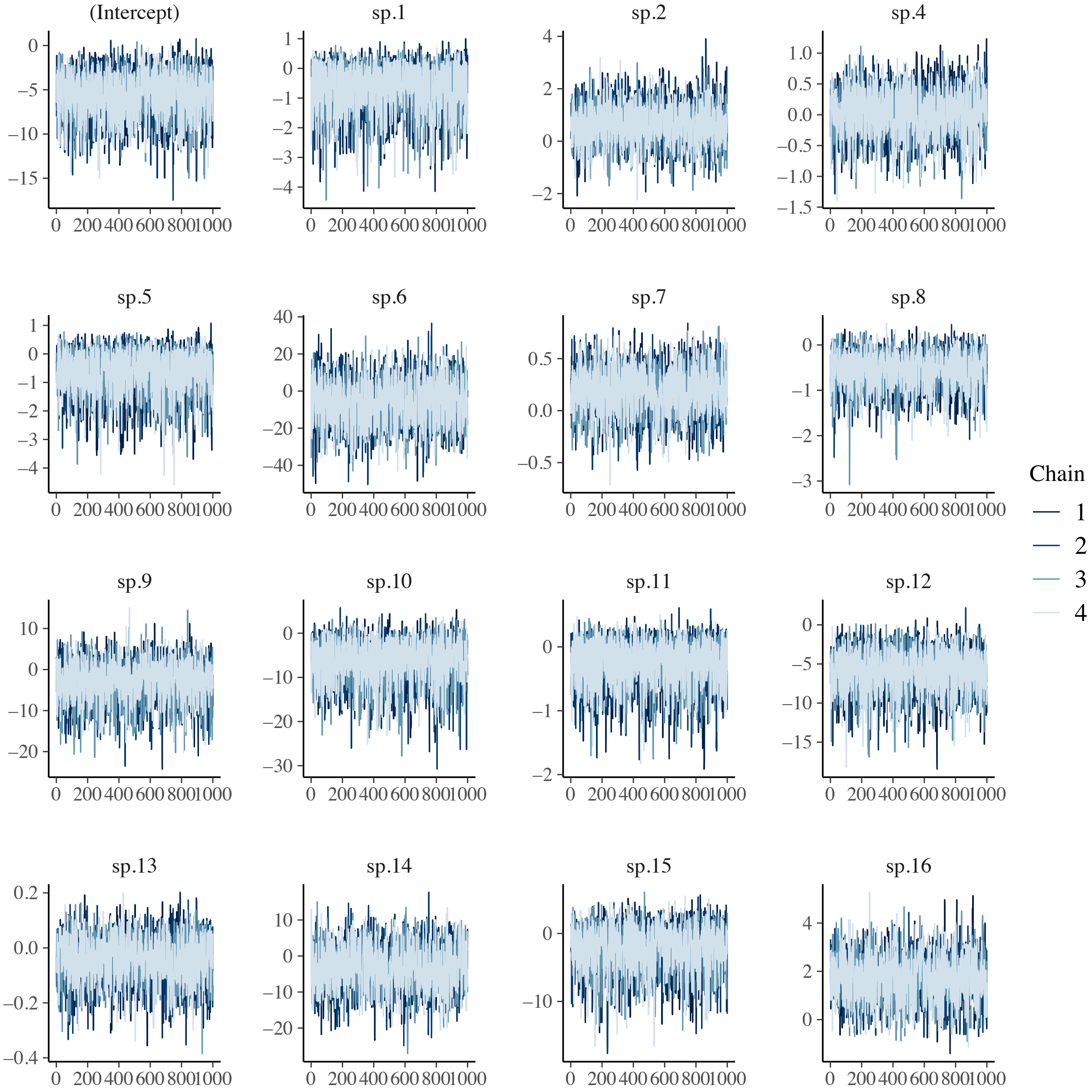
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```


(Intercept)
−6
−4
−2
0
sp.1
−6
−4
−2
0
sp.3
−0.5
0.0
0.5
1.0
1.5
2.0
2.5
sp.4
−0.3
0.0
0.3
0.6
sp.5
−5
−4
−3
−2
−1
0
sp.6
−40
−20
0
20
sp.7
0.0
0.2
0.4
sp.8
−0.2
0.0
0.2
sp.9
−30
−20
−10
0
sp.10
−5.0
−2.5
0.0
2.5
5.0
sp.11
−2.0
−1.5
−1.0
−0.5
0.0
sp.12
−20
−10
0
sp.13
0.0
0.1
0.2
sp.14
−10
0
10
20
sp.15
−15
−10
−5
0
sp.16
−4
−2
0

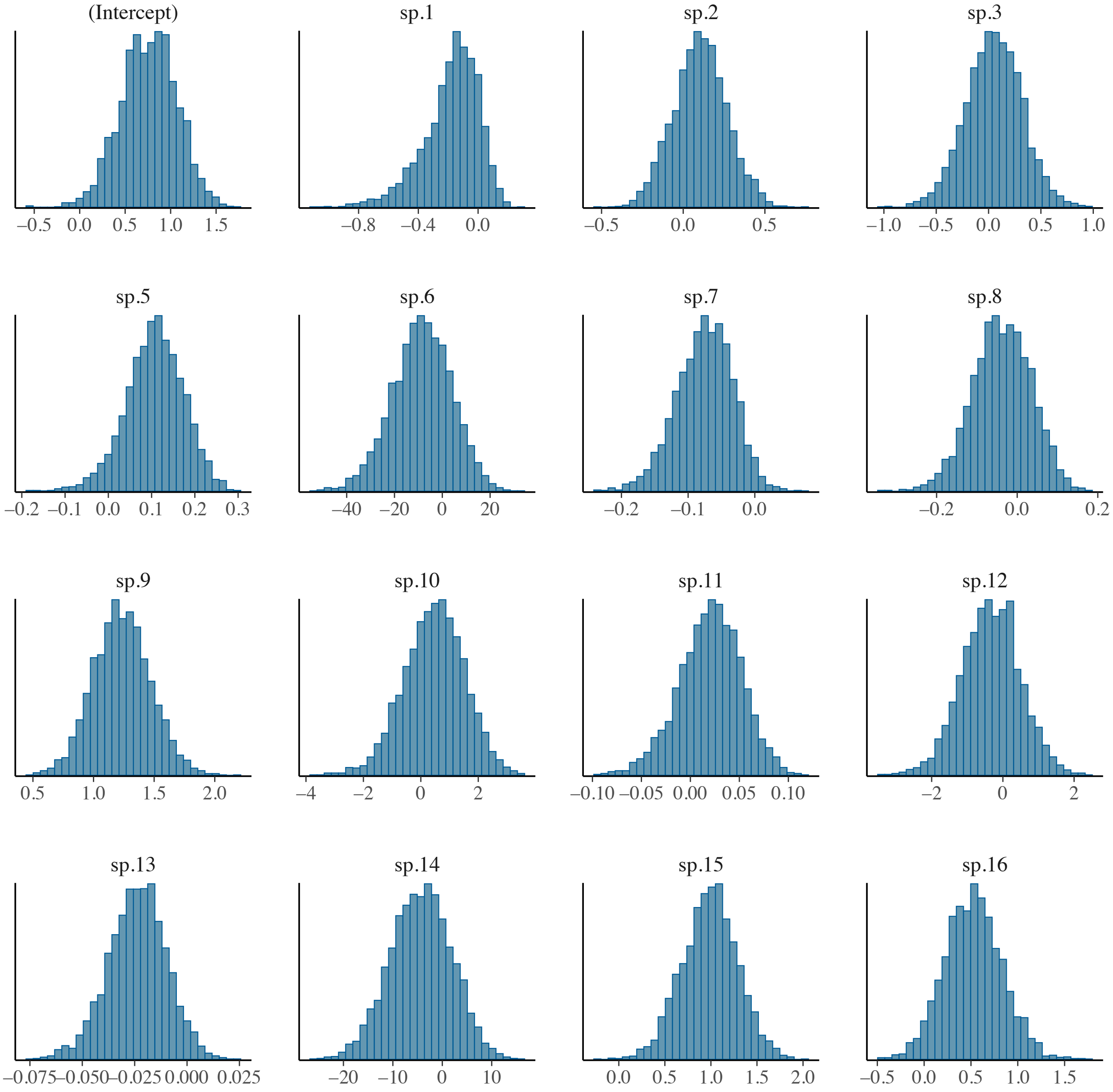


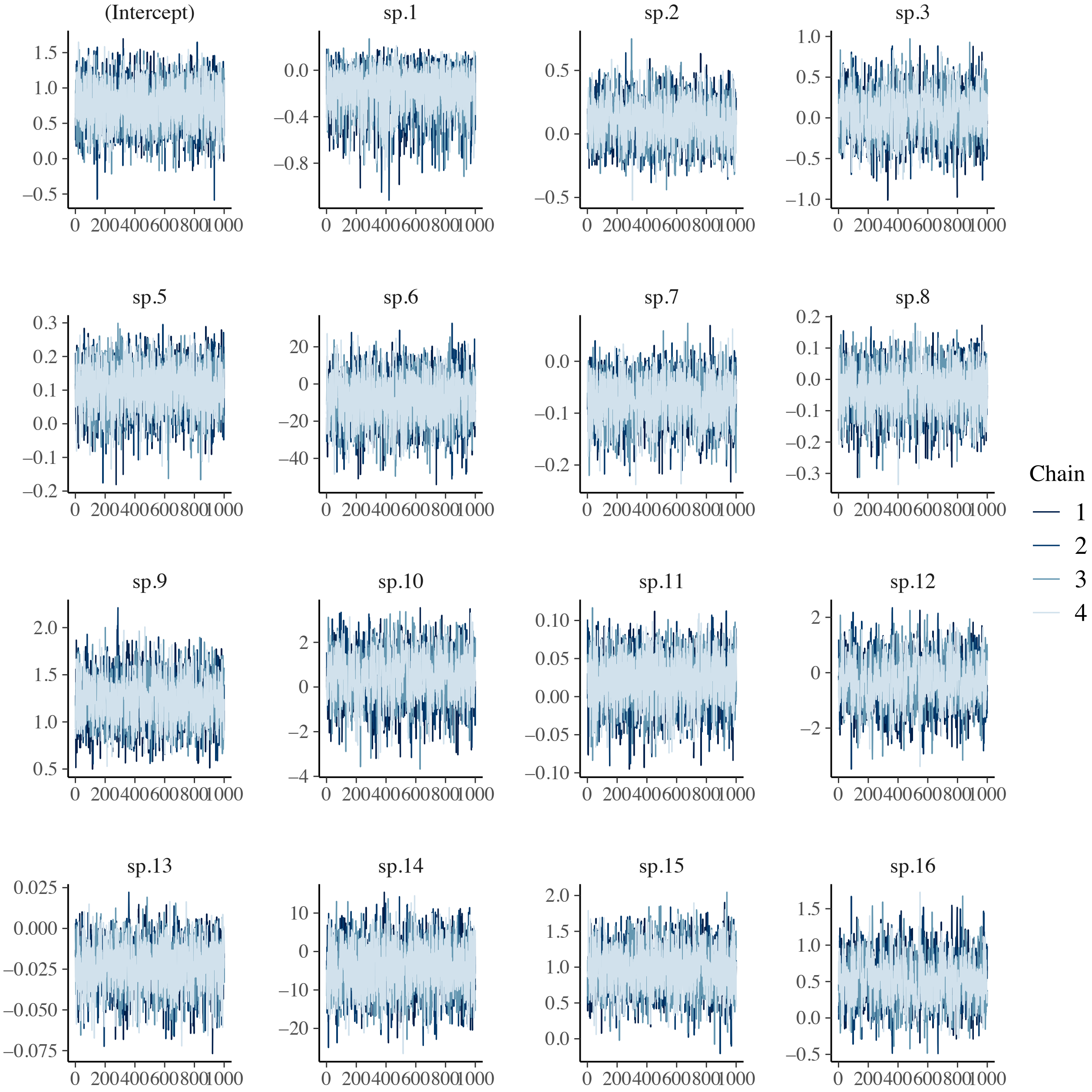
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```


(Intercept)
−15 −10 −5 0
sp.1
−4 −3 −2 −1 0 1
sp.2
−2 0 2 4
sp.4
−1.5 −1.0 −0.5 0.0 0.5 1.0
sp.5
−4 −3 −2 −1 0 1
sp.6
−40 −20 0 20 40
sp.7
−0.5 0.0 0.5
sp.8
−3 −2 −1 0
sp.9
−20 −10 0 10
sp.10
−30 −20 −10 0
sp.11
−2 −1 0
sp.12
−15 −10 −5 0
sp.13
−0.4 −0.2 0.0 0.2
sp.14
−30 −20 −10 0 10 20
sp.15
−10 0
sp.16
0 2 4

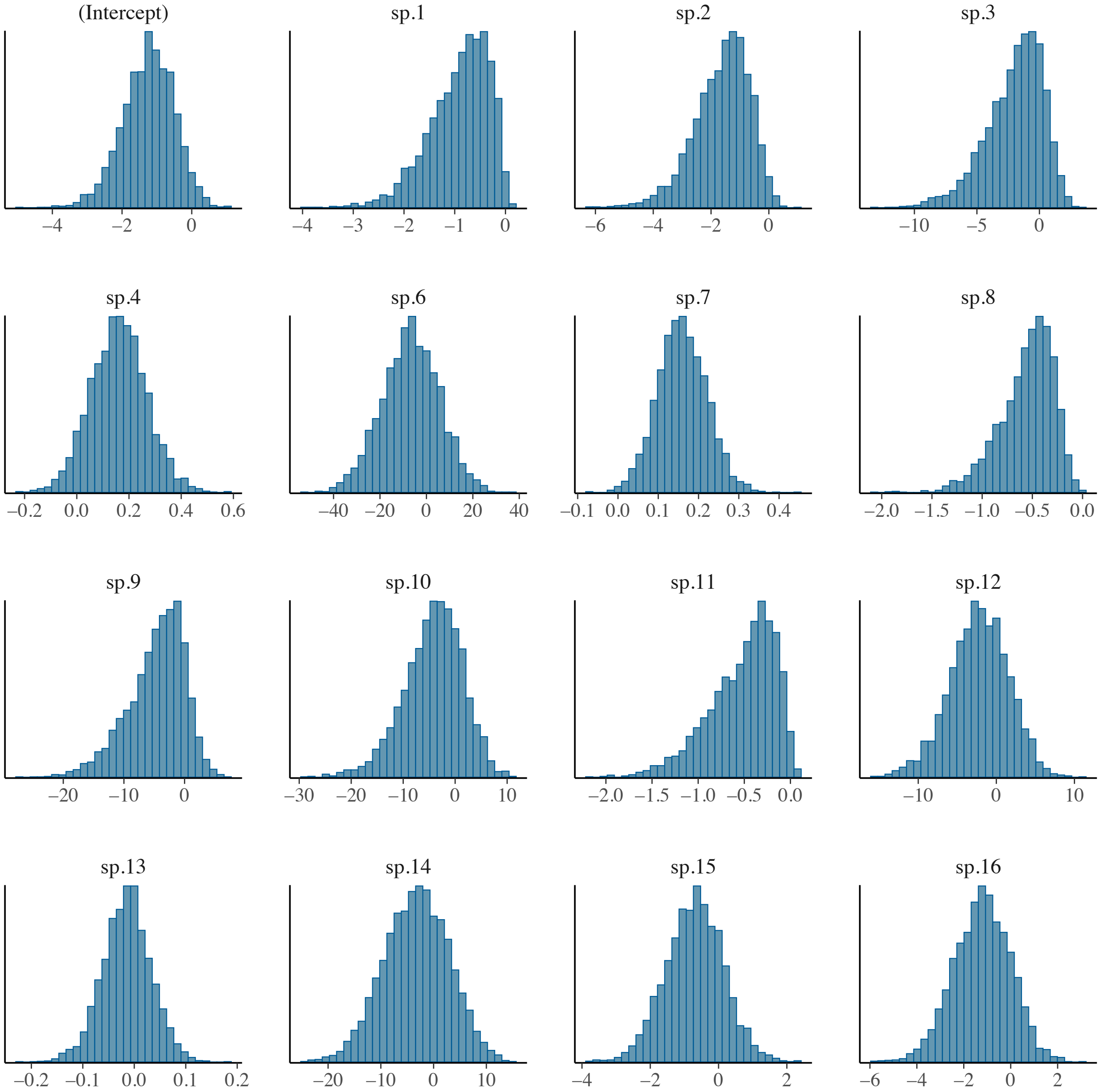


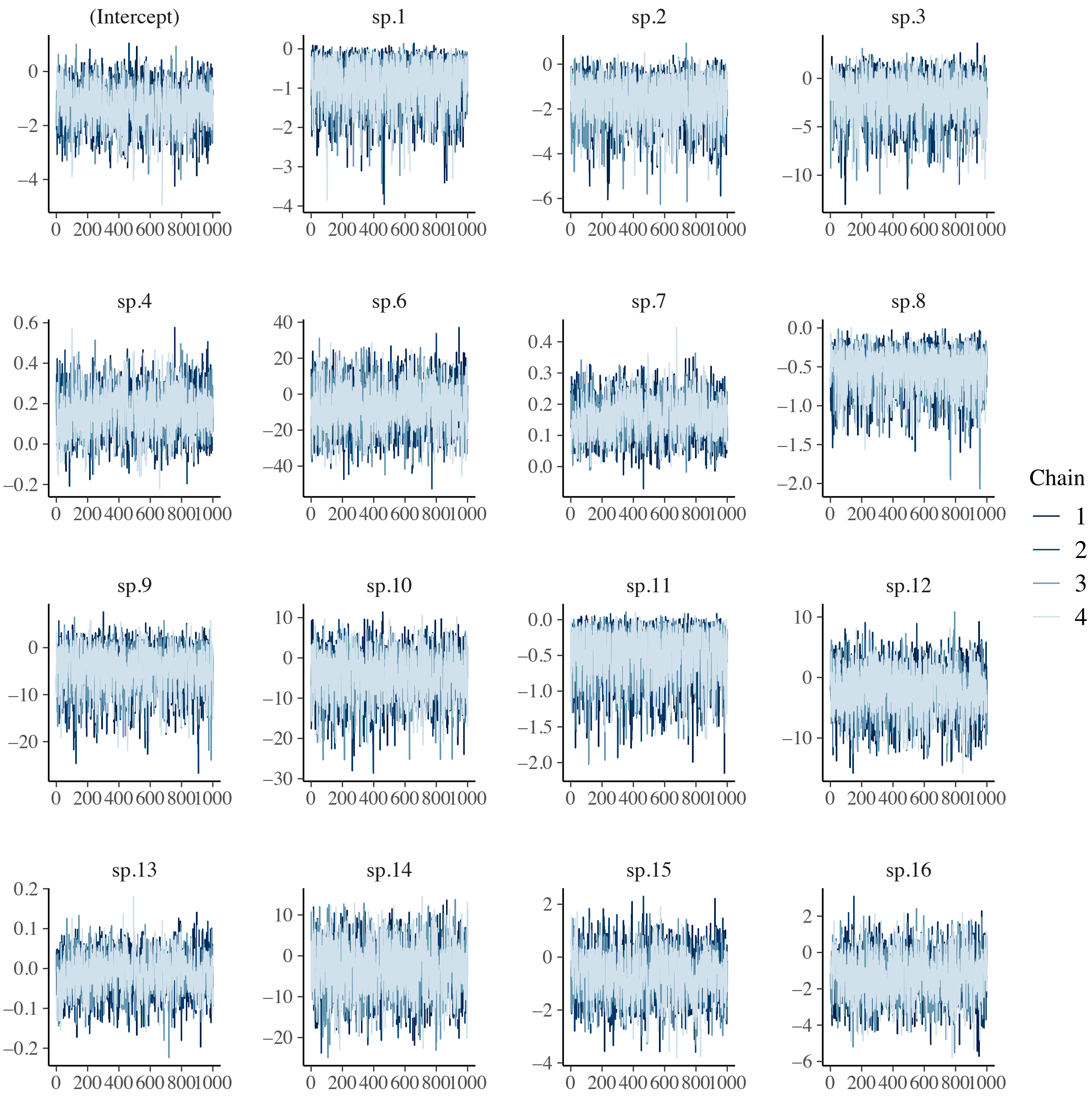
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```


(Intercept)
−0.5 0.0 0.5 1.0 1.5
sp.1
−0.8 −0.4 0.0
sp.2
−0.5 0.0 0.5
sp.3
−1.0 −0.5 0.0 0.5 1.0
sp.5
−0.2 −0.1 0.0 0.1 0.2 0.3
sp.6
−40 −20 0 20
sp.7
−0.2 −0.1 0.0
sp.8
−0.2 0.0 0.2
sp.9
0.5 1.0 1.5 2.0
sp.10
−4 −2 0 2
sp.11
−0.10 −0.05 0.00 0.05 0.10
sp.12
−2 0 2
sp.13
−0.075 −0.050 −0.025 0.000 0.025
sp.14
−20 −10 0 10
sp.15
0.0 0.5 1.0 1.5 2.0
sp.16
−0.5 0.0 0.5 1.0 1.5

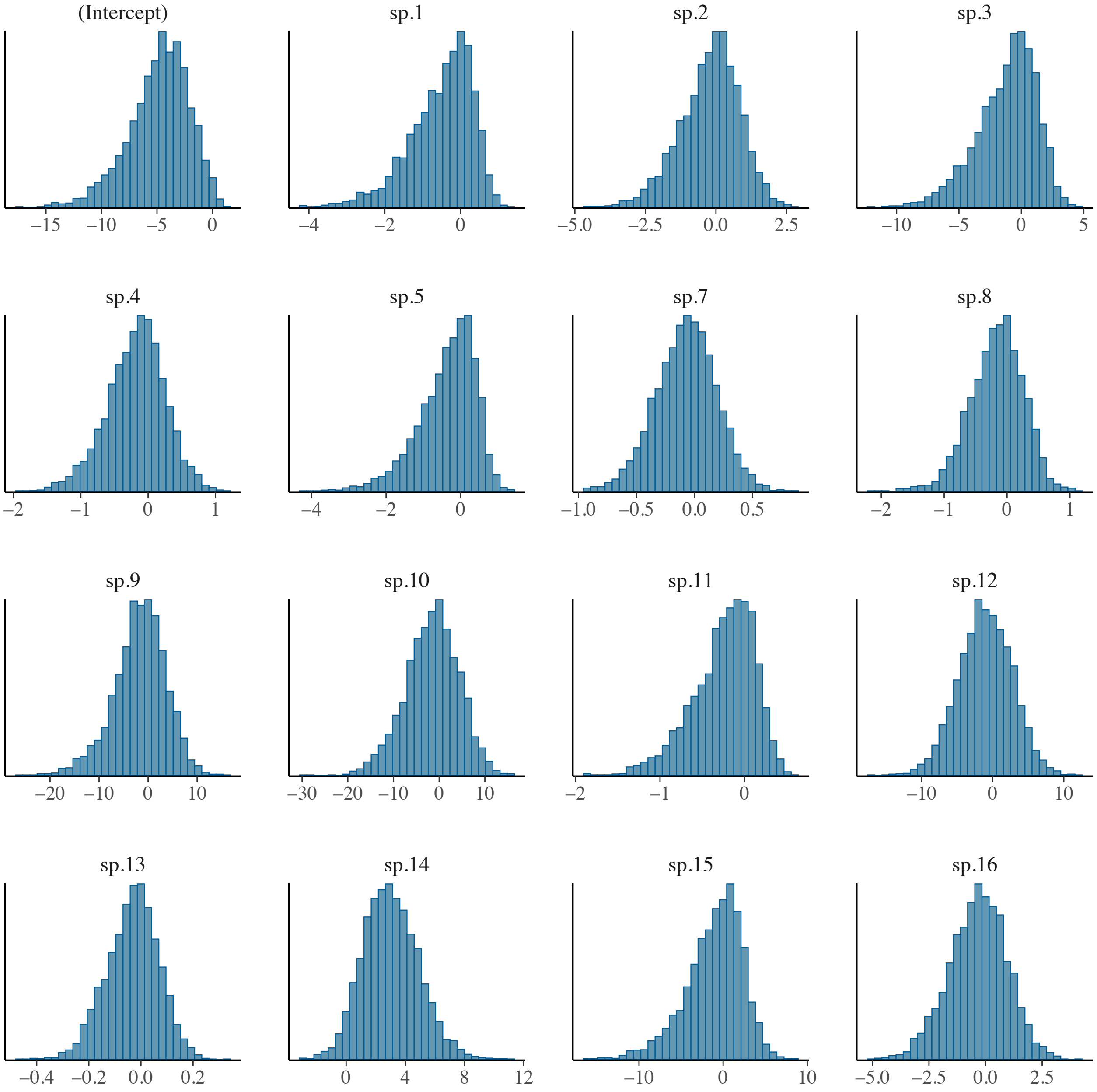


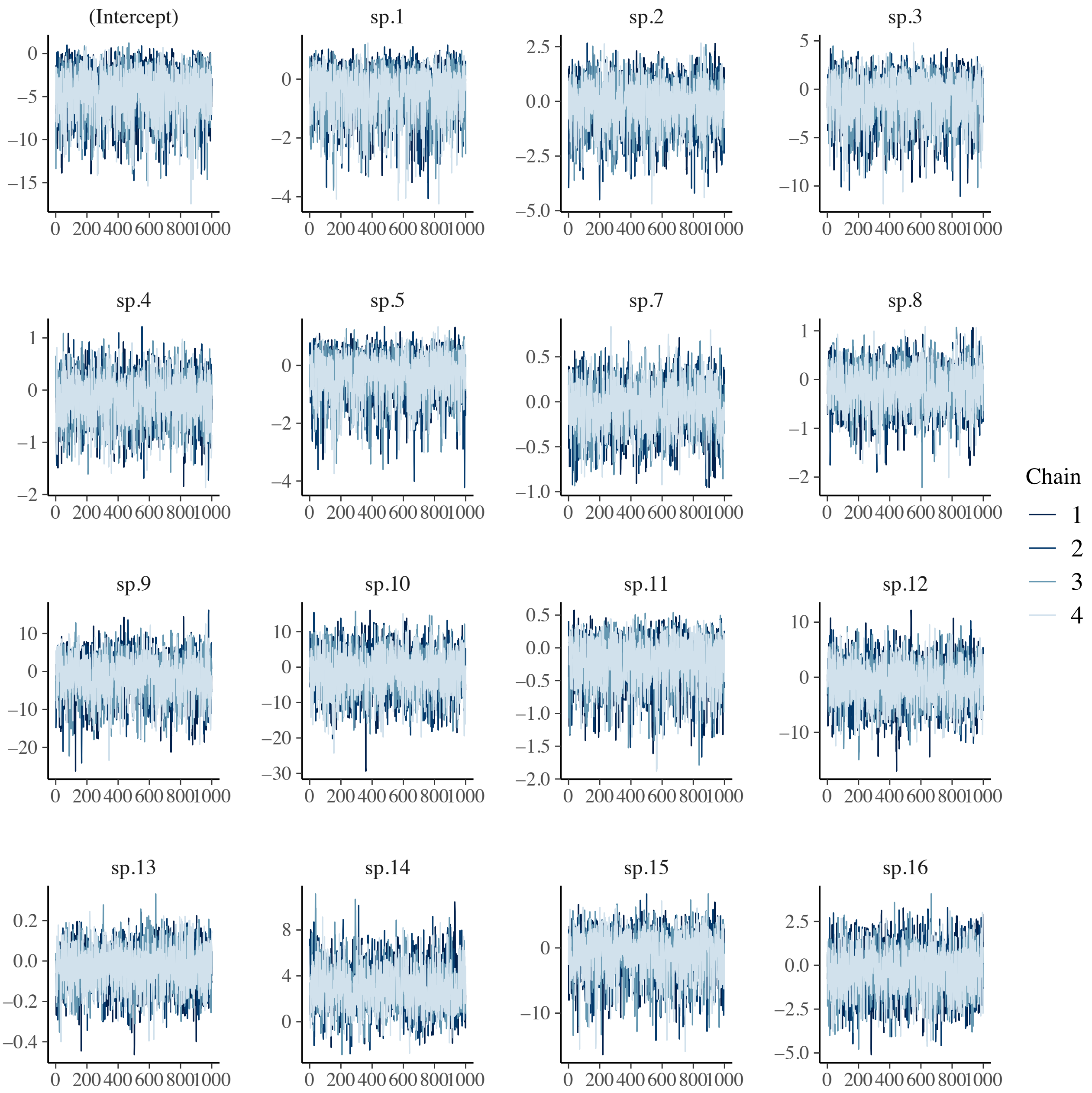
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```


(Intercept)
−4 −2 0
sp.1
−4 −3 −2 −1 0
sp.2
−6 −4 −2 0
sp.3
−10 −5 0
sp.4
−0.2 0.0 0.2 0.4 0.6
sp.6
−40 −20 0 20 40
sp.7
−0.1 0.0 0.1 0.2 0.3 0.4
sp.8
−2.0 −1.5 −1.0 −0.5 0.0
sp.9
−20 −10 0
sp.10
−30 −20 −10 0 10
sp.11
−2.0 −1.5 −1.0 −0.5 0.0
sp.12
−10 0 10
sp.13
−0.2 −0.1 0.0 0.1 0.2
sp.14
−20 −10 0 10
sp.15
−4 −2 0 2
sp.16
−6 −4 −2 0 2

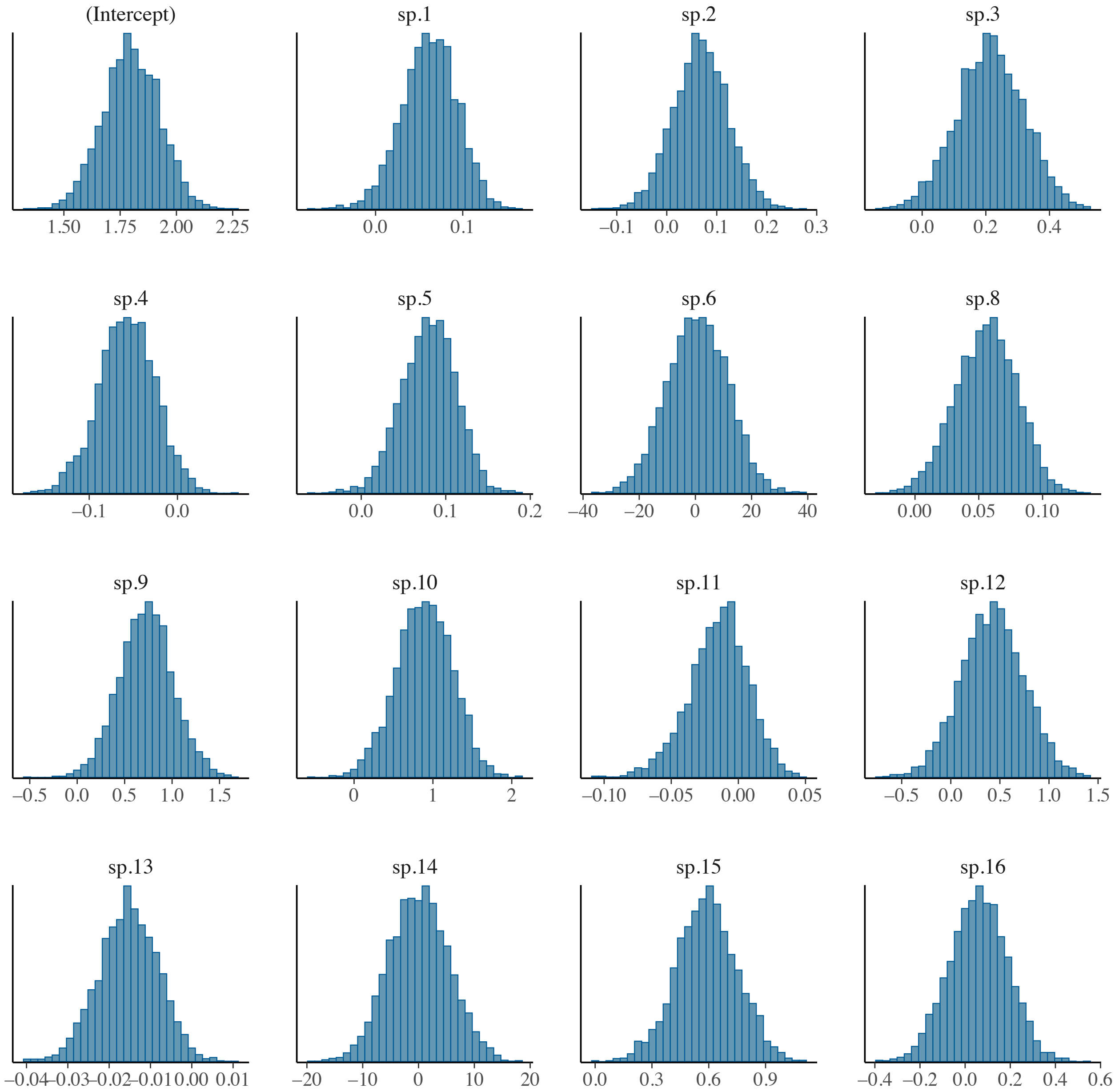


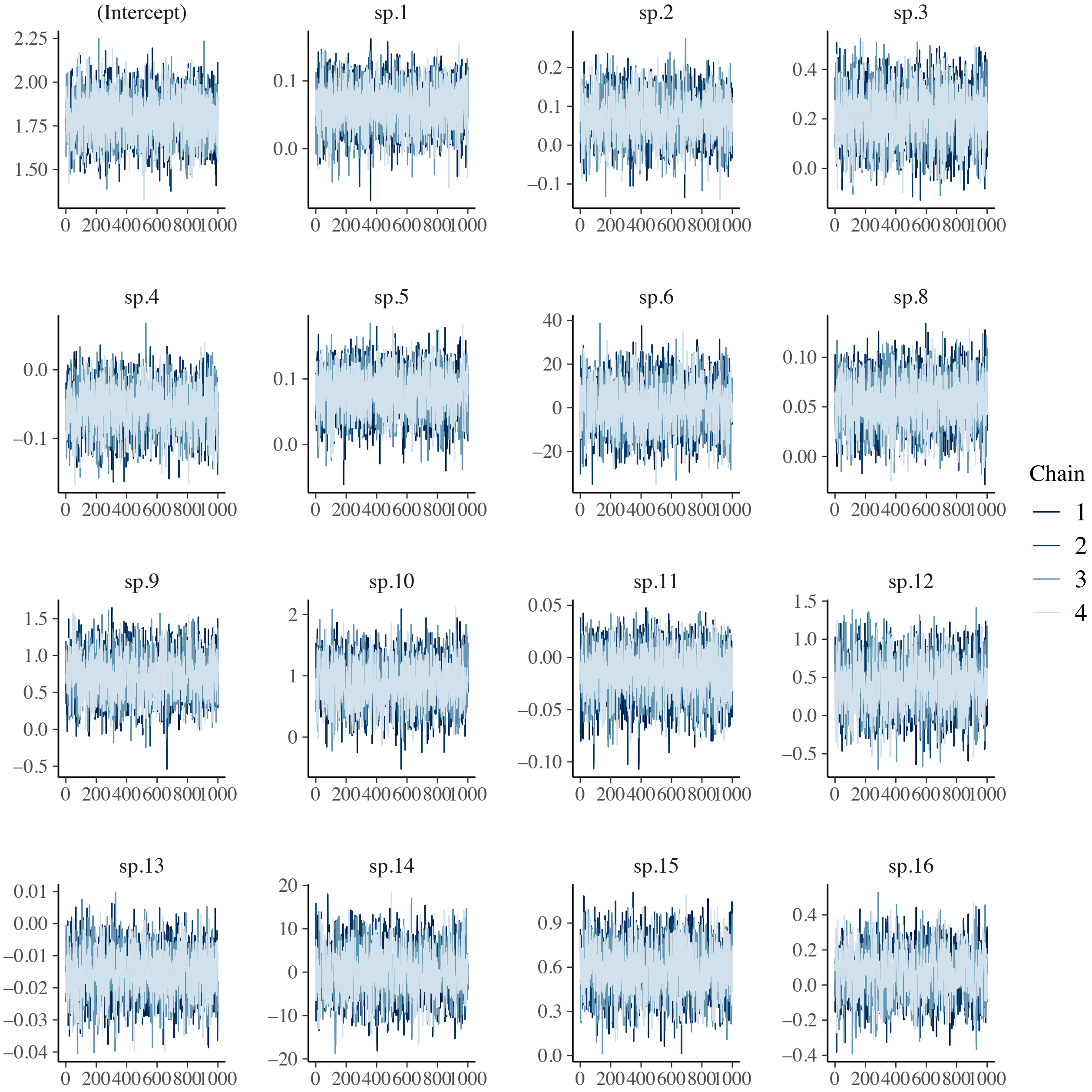
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```


(Intercept)
−15
−10
−5
0
sp.1
−4
−2
0
sp.2
−5.0
−2.5
0.0
2.5
sp.3
−10
−5
0
5
sp.4
−2
−1
0
1
sp.5
−4
−2
0
sp.7
−1.0
−0.5
0.0
0.5
sp.8
−2
−1
0
1
sp.9
−20
−10
0
10
sp.10
−30
−20
−10
0
10
sp.11
−2
−1
0
sp.12
−10
0
10
sp.13
−0.4
−0.2
0.0
0.2
sp.14
0
4
8
12
sp.15
−10
0
10
sp.16
−5.0
−2.5
0.0
2.5

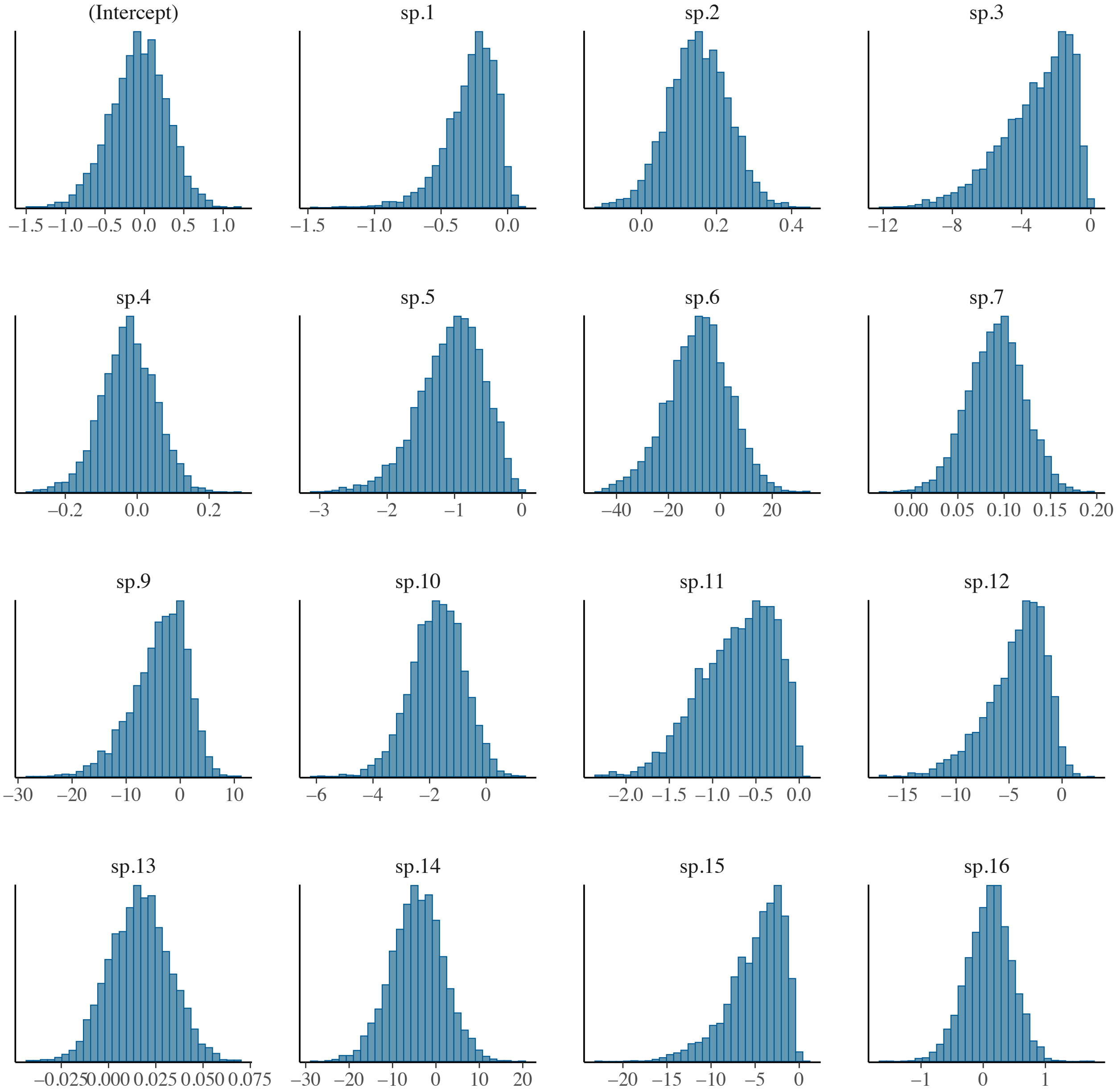


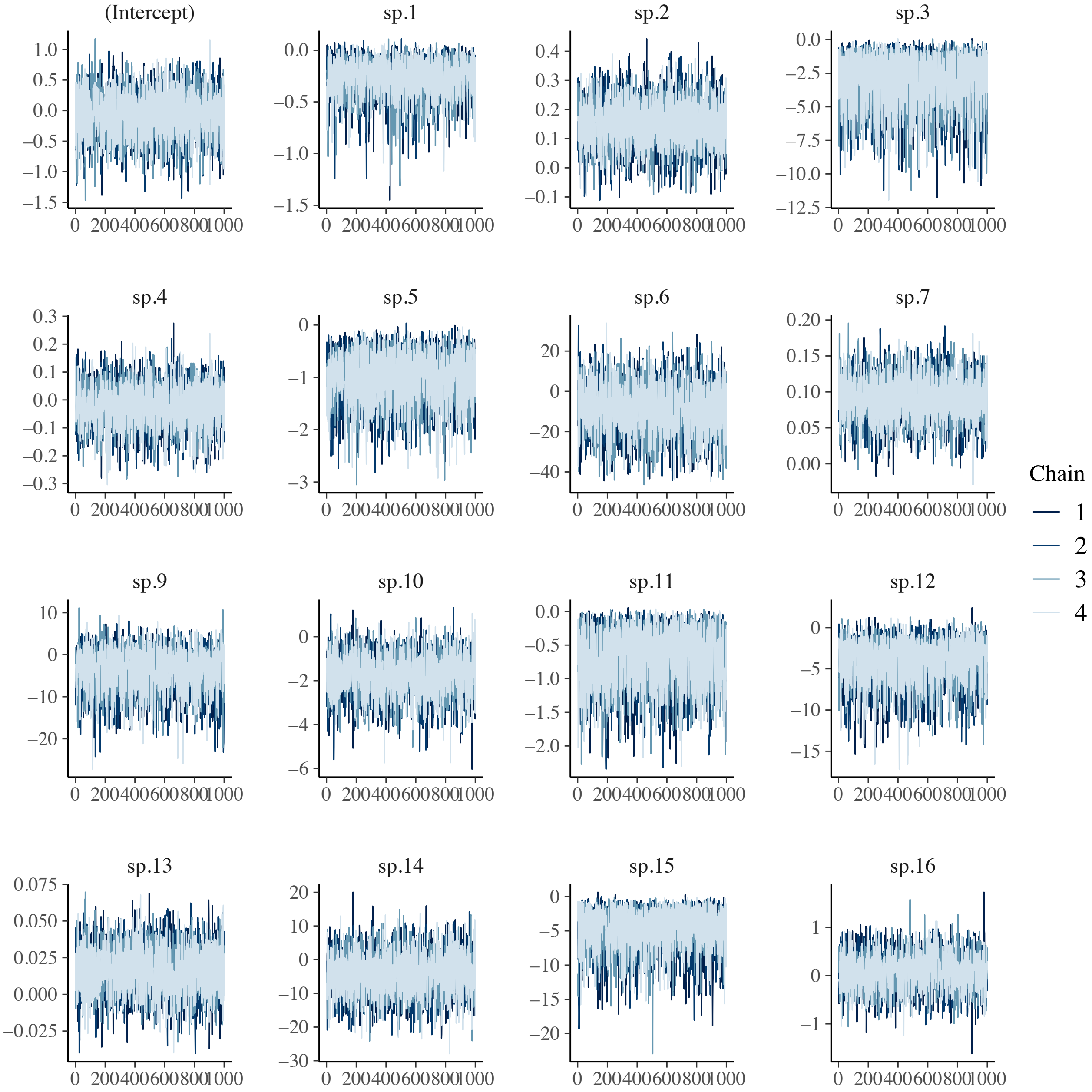
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```


(Intercept)
1.50
1.75
2.00
2.25
sp.1
0.0
0.1
sp.2
−0.1
0.0
0.1
0.2
0.3
sp.3
0.0
0.2
0.4
sp.4
−0.1
0.0
sp.5
0.0
0.1
0.2
sp.6
−40
−20
0
20
40
sp.8
0.00
0.05
0.10
sp.9
−0.5
0.0
0.5
1.0
1.5
sp.10
0
1
2
sp.11
−0.10
−0.05
0.00
0.05
sp.12
−0.5
0.0
0.5
1.0
1.5
sp.13
−0.04
−0.03
−0.02
−0.01
0.00
0.01
sp.14
−20
−10
0
10
20
sp.15
0.0
0.3
0.6
0.9
sp.16
−0.4
−0.2
0.0
0.2
0.4
0.6

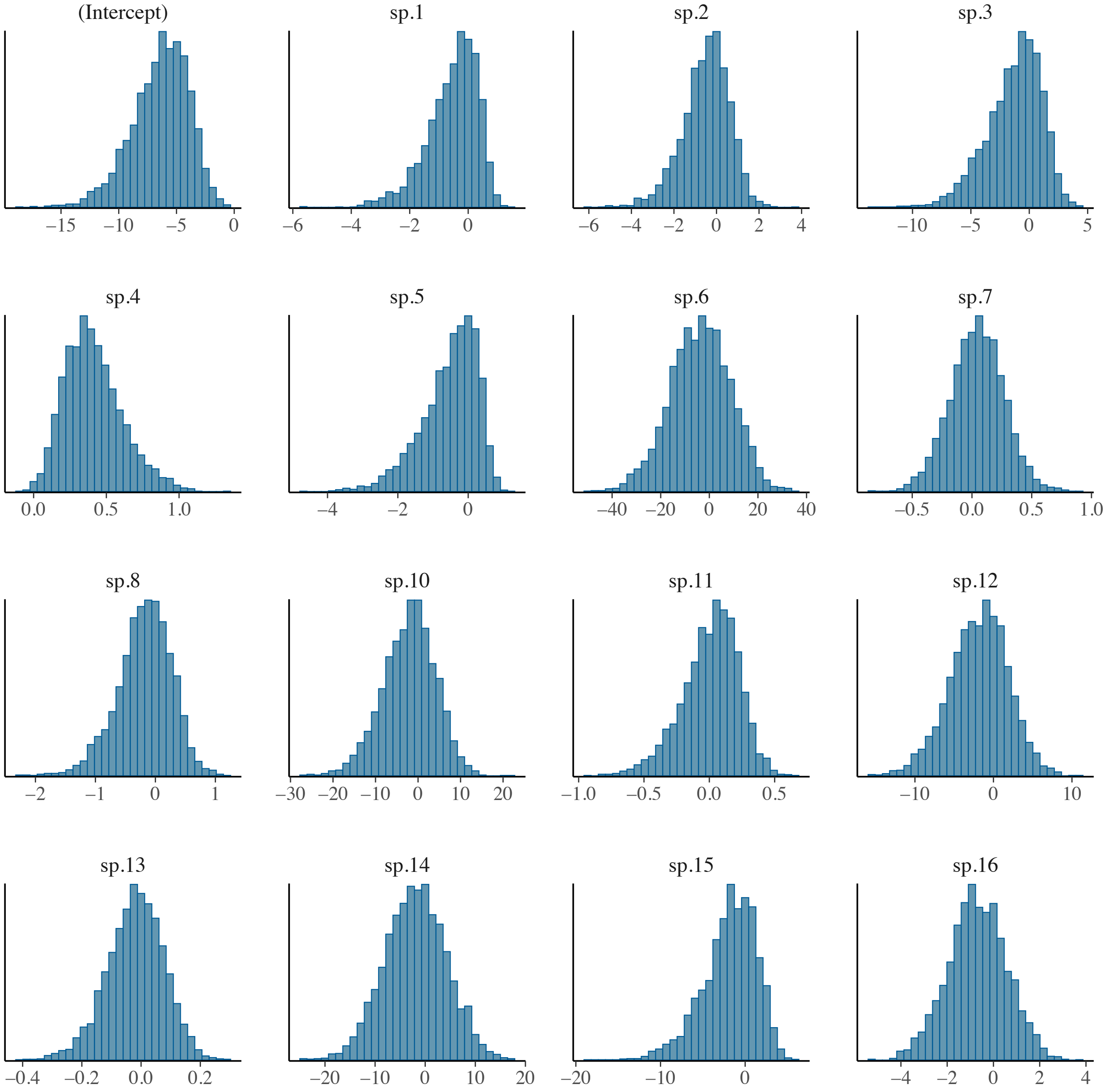


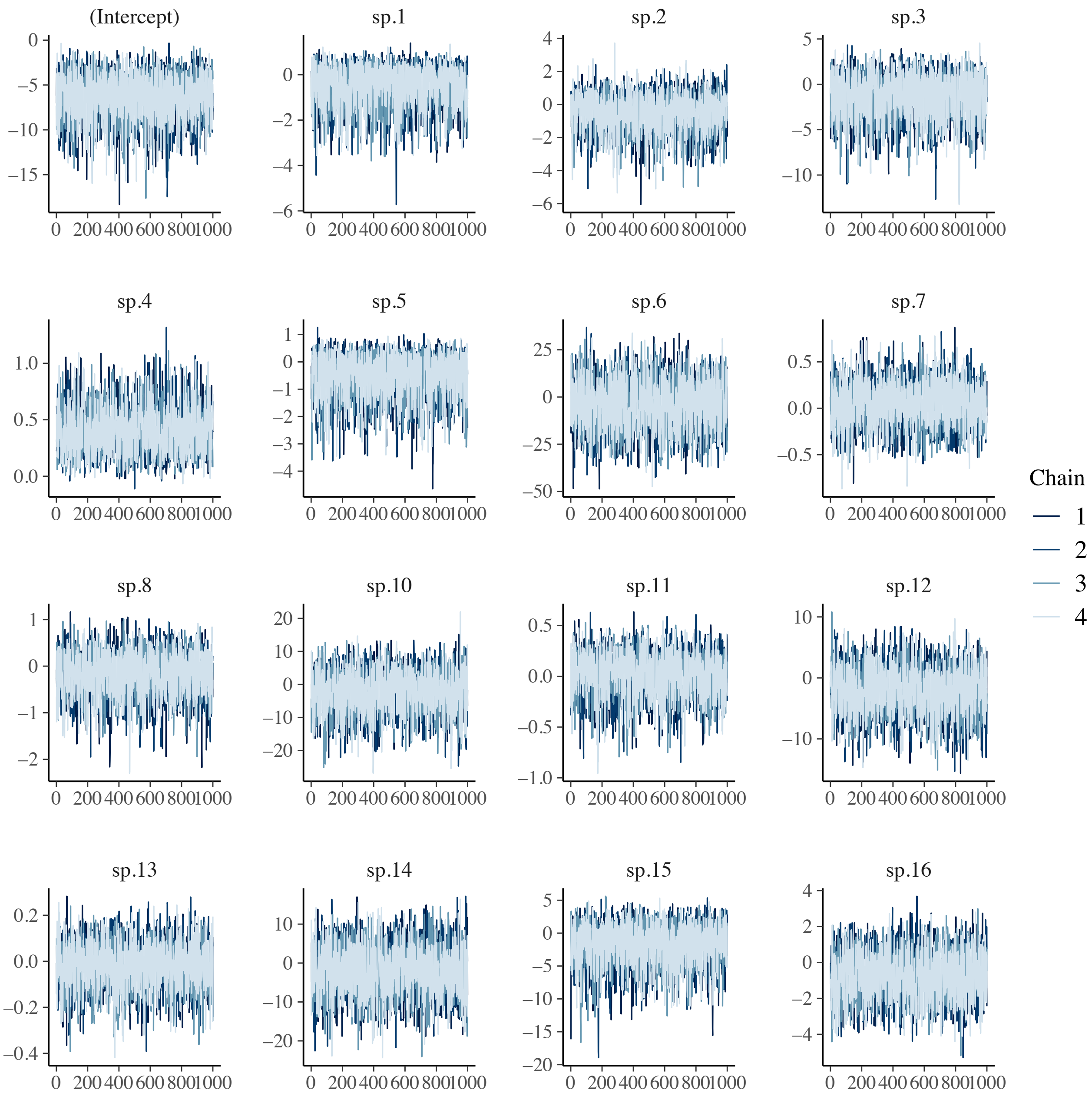
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```


(Intercept)
−1.5 −1.0 −0.5 0.0 0.5 1.0
sp.1
−1.5 −1.0 −0.5 0.0
sp.2
0.0 0.2 0.4
sp.3
−12 −8 −4 0
sp.4
−0.2 0.0 0.2
sp.5
−3 −2 −1 0
sp.6
−40 −20 0 20
sp.7
0.00 0.05 0.10 0.15 0.20
sp.9
−30 −20 −10 0 10
sp.10
−6 −4 −2 0
sp.11
−2.0 −1.5 −1.0 −0.5 0.0
sp.12
−15 −10 −5 0
sp.13
−0.025 0.000 0.025 0.050 0.075
sp.14
−30 −20 −10 0 10 20
sp.15
−20 −15 −10 −5 0
sp.16
−1 0 1

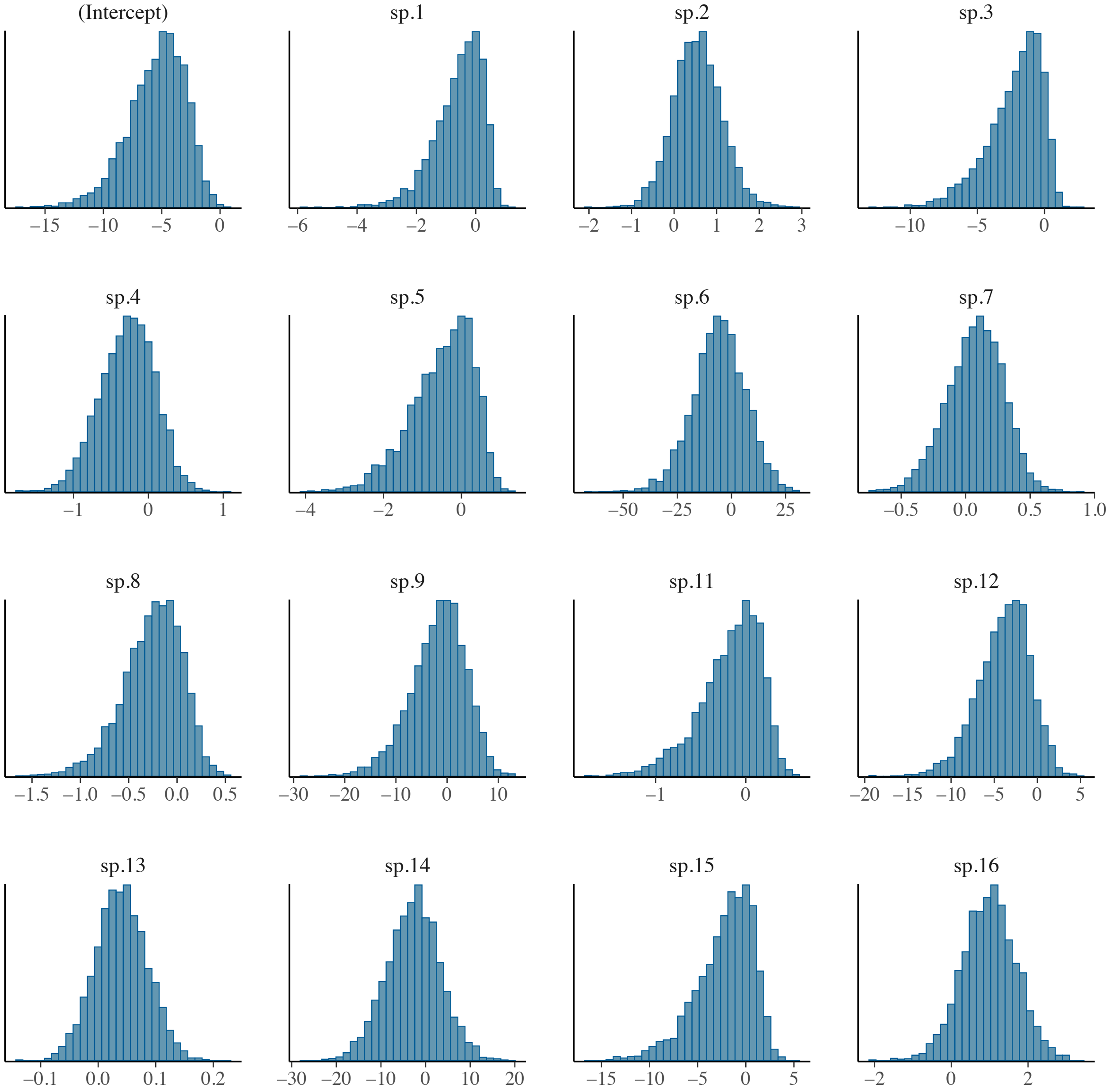


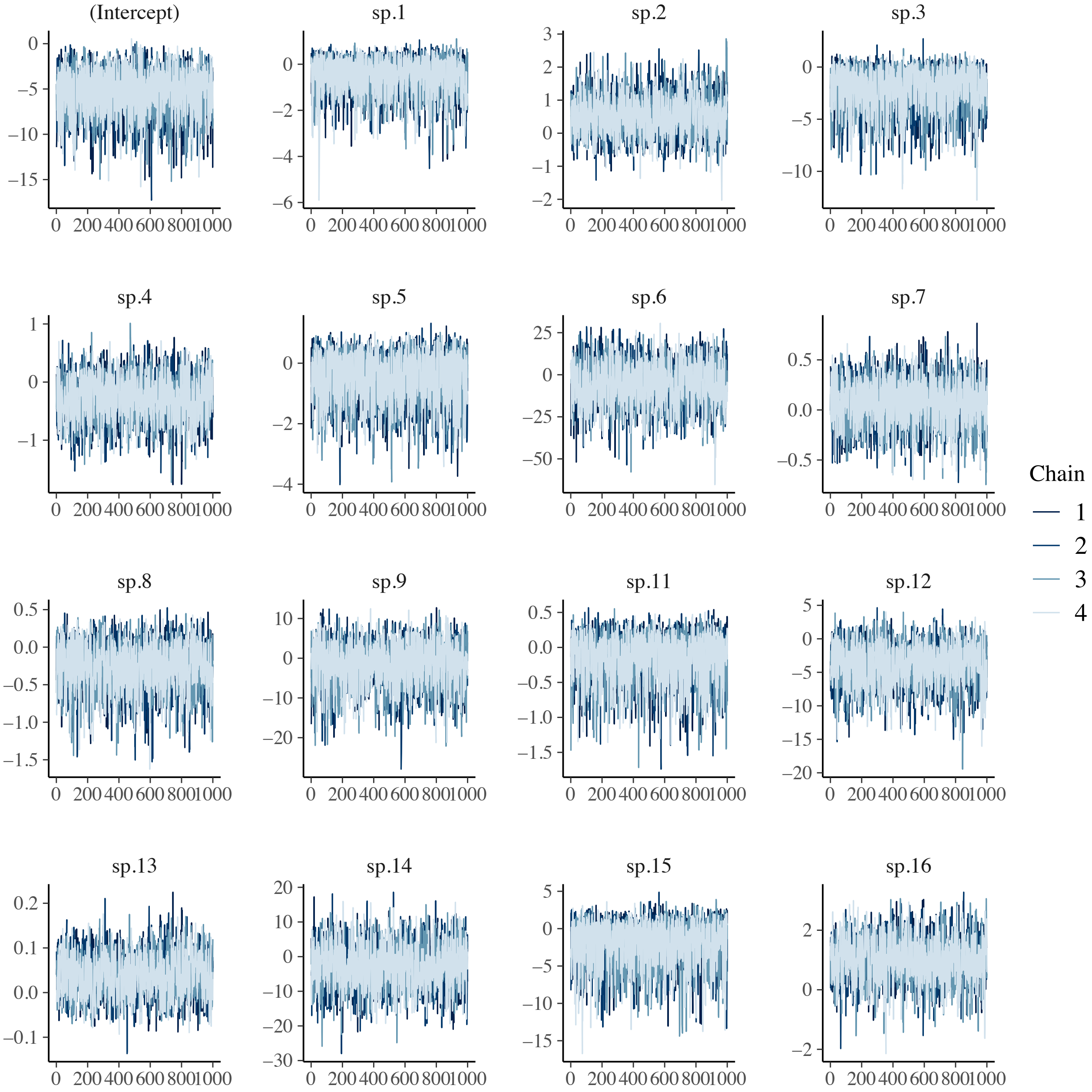
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```


(Intercept)
−15
−10
−5
0
sp.1
−6
−4
−2
0
sp.2
−6
−4
−2
0
2
4
sp.3
−10
−5
0
5
sp.4
0.0
0.5
1.0
sp.5
−4
−2
0
sp.6
−40
−20
0
20
40
sp.7
−0.5
0.0
0.5
1.0
sp.8
−2
−1
0
1
sp.10
−30
−20
−10
0
10
20
sp.11
−1.0
−0.5
0.0
0.5
sp.12
−10
0
10
sp.13
−0.4
−0.2
0.0
0.2
sp.14
−20
−10
0
10
20
sp.15
−20
−10
0
sp.16
−4
−2
0
2
4

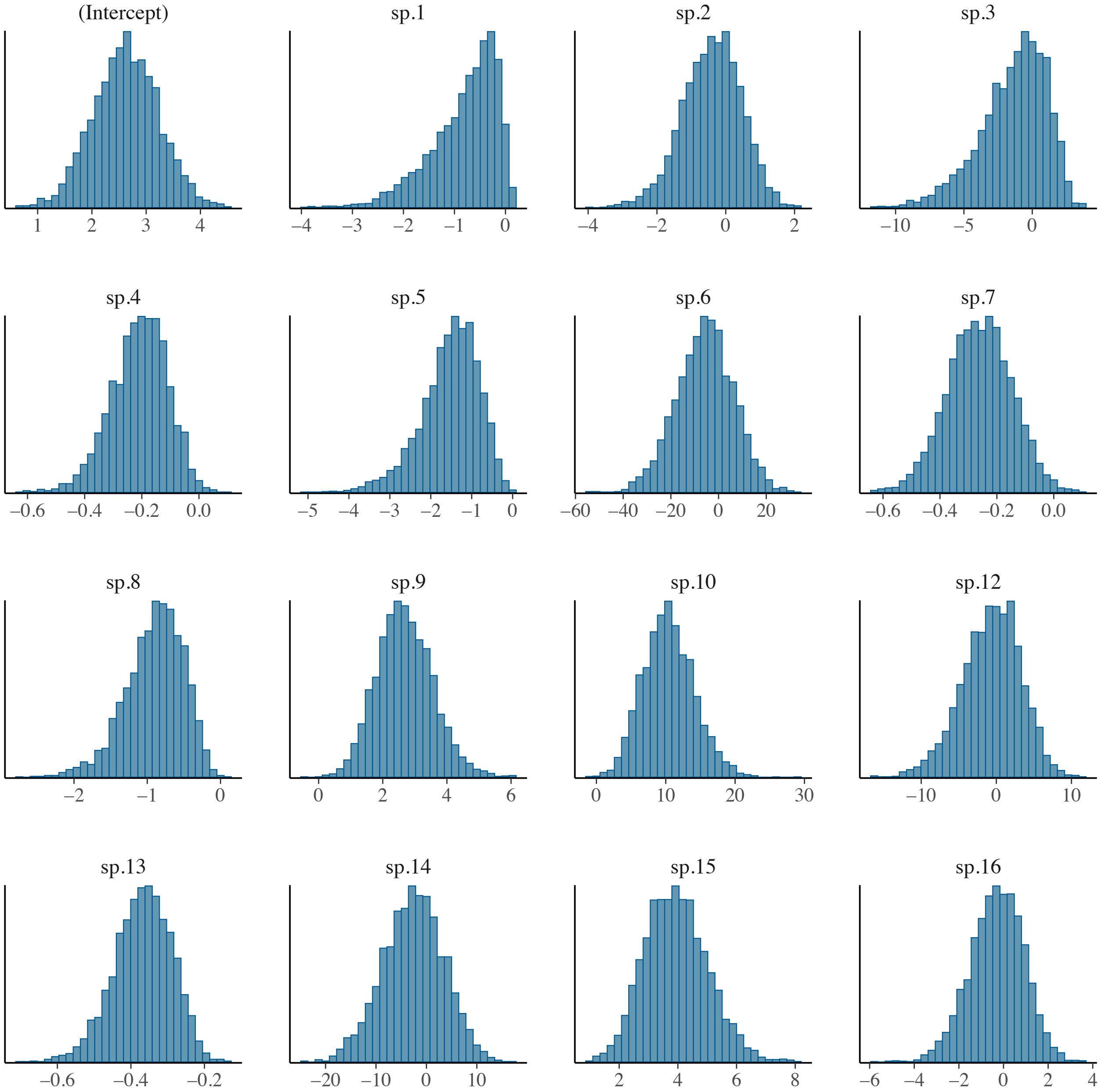


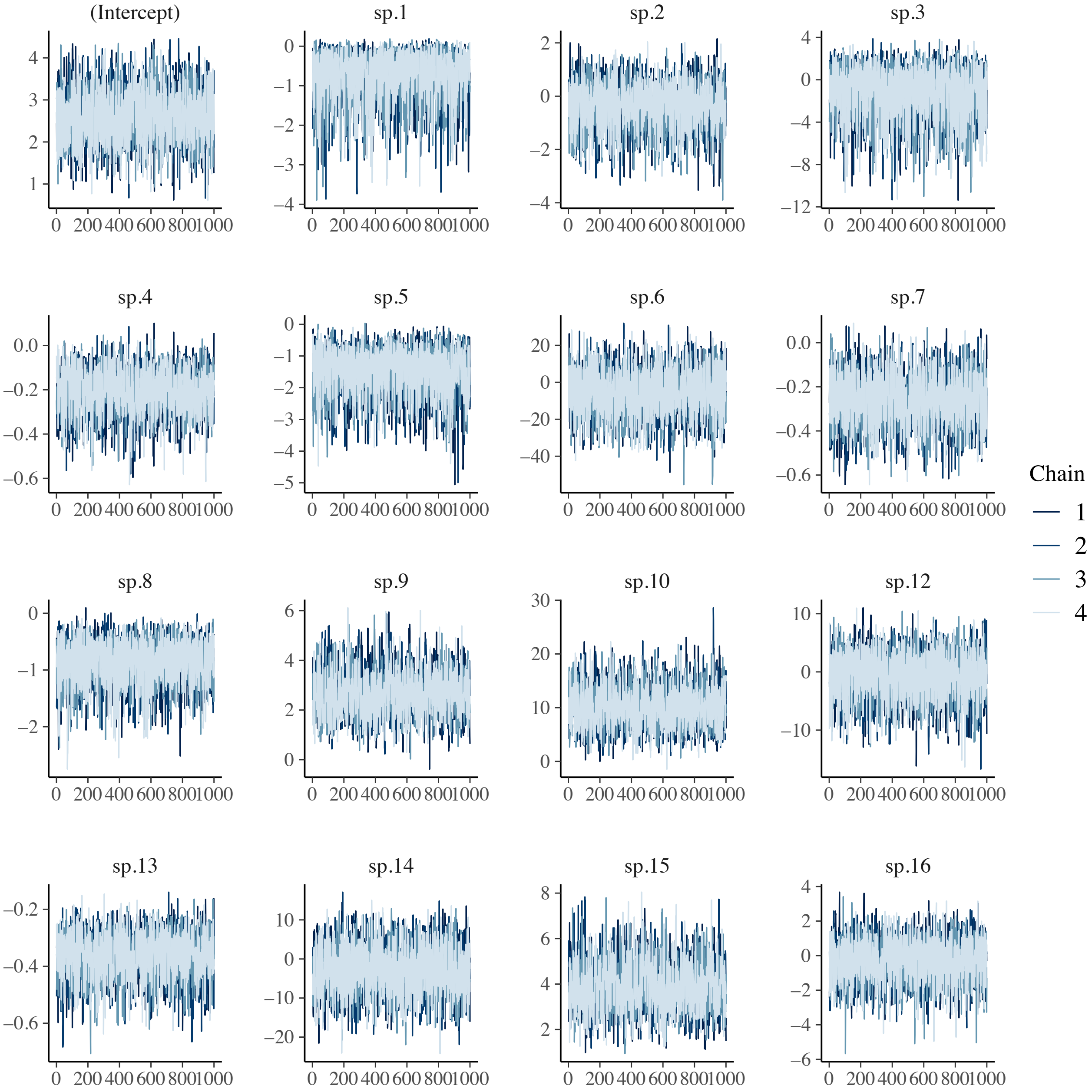
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```


(Intercept)
−15
−10
−5
0
sp.1
−6
−4
−2
0
sp.2
−2
−1
0
1
2
3
sp.3
−10
−5
0
sp.4
−1
0
1
sp.5
−4
−2
0
sp.6
−50
−25
0
25
sp.7
−0.5
0.0
0.5
1.0
sp.8
−1.5
−1.0
−0.5
0.0
0.5
sp.9
−30
−20
−10
0
10
sp.11
−1
0
sp.12
−20
−15
−10
−5
0
5
sp.13
−0.1
0.0
0.1
0.2
sp.14
−30
−20
−10
0
10
20
sp.15
−15
−10
−5
0
5
sp.16
−2
0
2

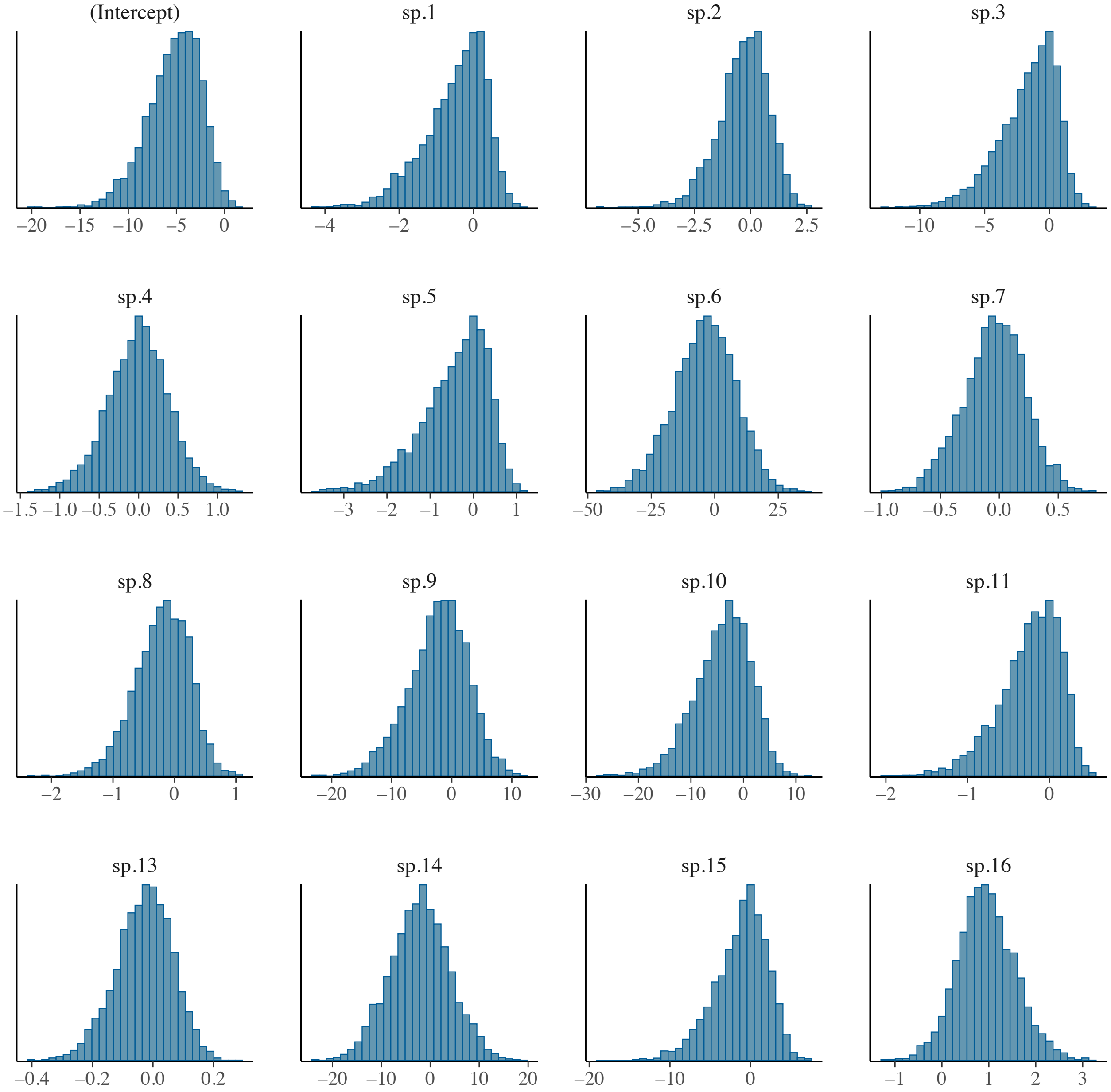


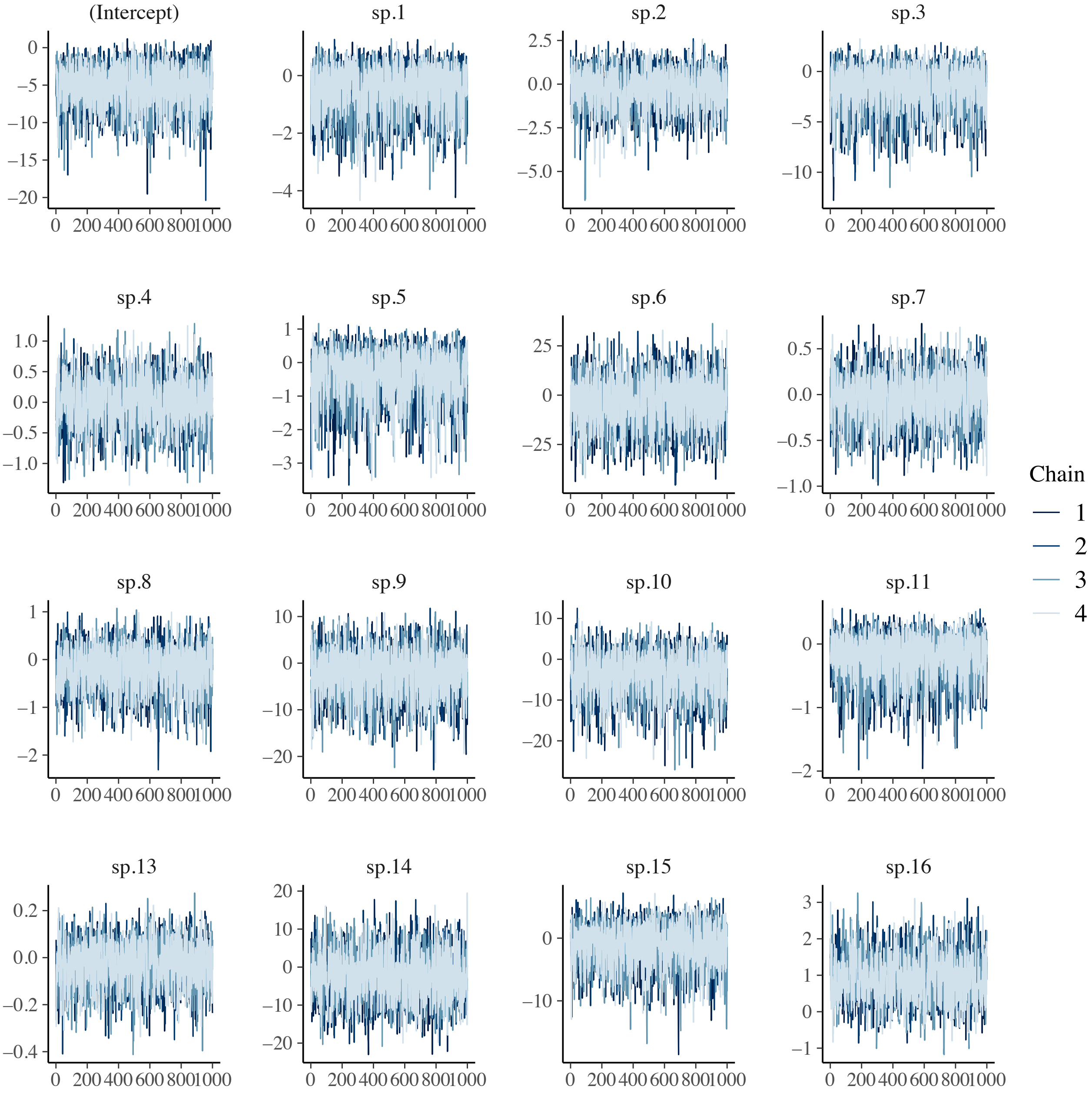
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```


(Intercept)
1
2
3
4
sp.1
−4
−3
−2
−1
0
sp.2
−4
−2
0
2
sp.3
−10
−5
0
sp.4
−0.6
−0.4
−0.2
0.0
sp.5
−5
−4
−3
−2
−1
0
sp.6
−60
−40
−20
0
20
sp.7
−0.6
−0.4
−0.2
0.0
sp.8
−2
−1
0
sp.9
0
2
4
6
sp.10
0
10
20
30
sp.12
−10
0
10
sp.13
−0.6
−0.4
−0.2
sp.14
−20
−10
0
10
sp.15
2
4
6
8
sp.16
−6
−4
−2
0
2
4

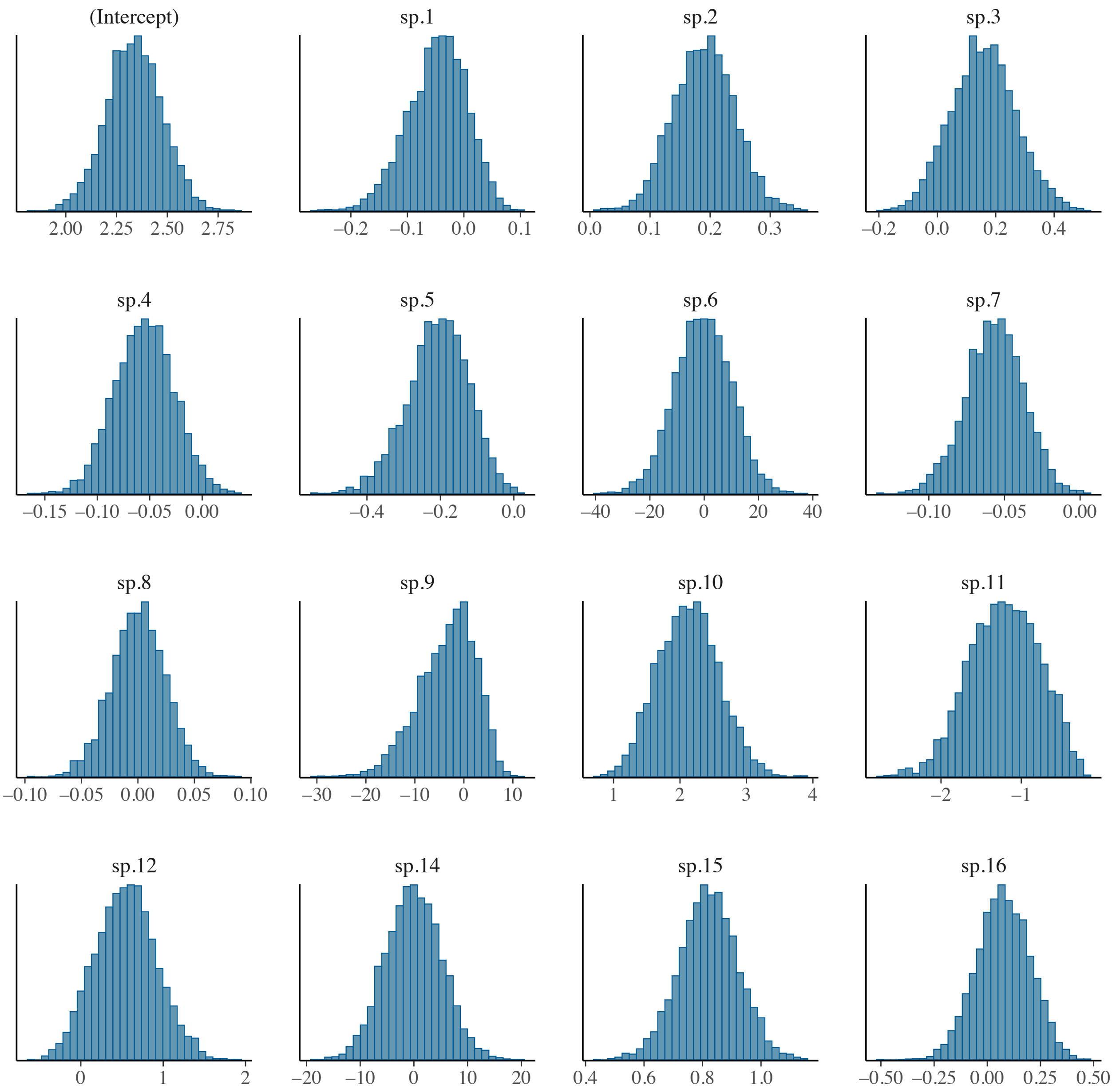


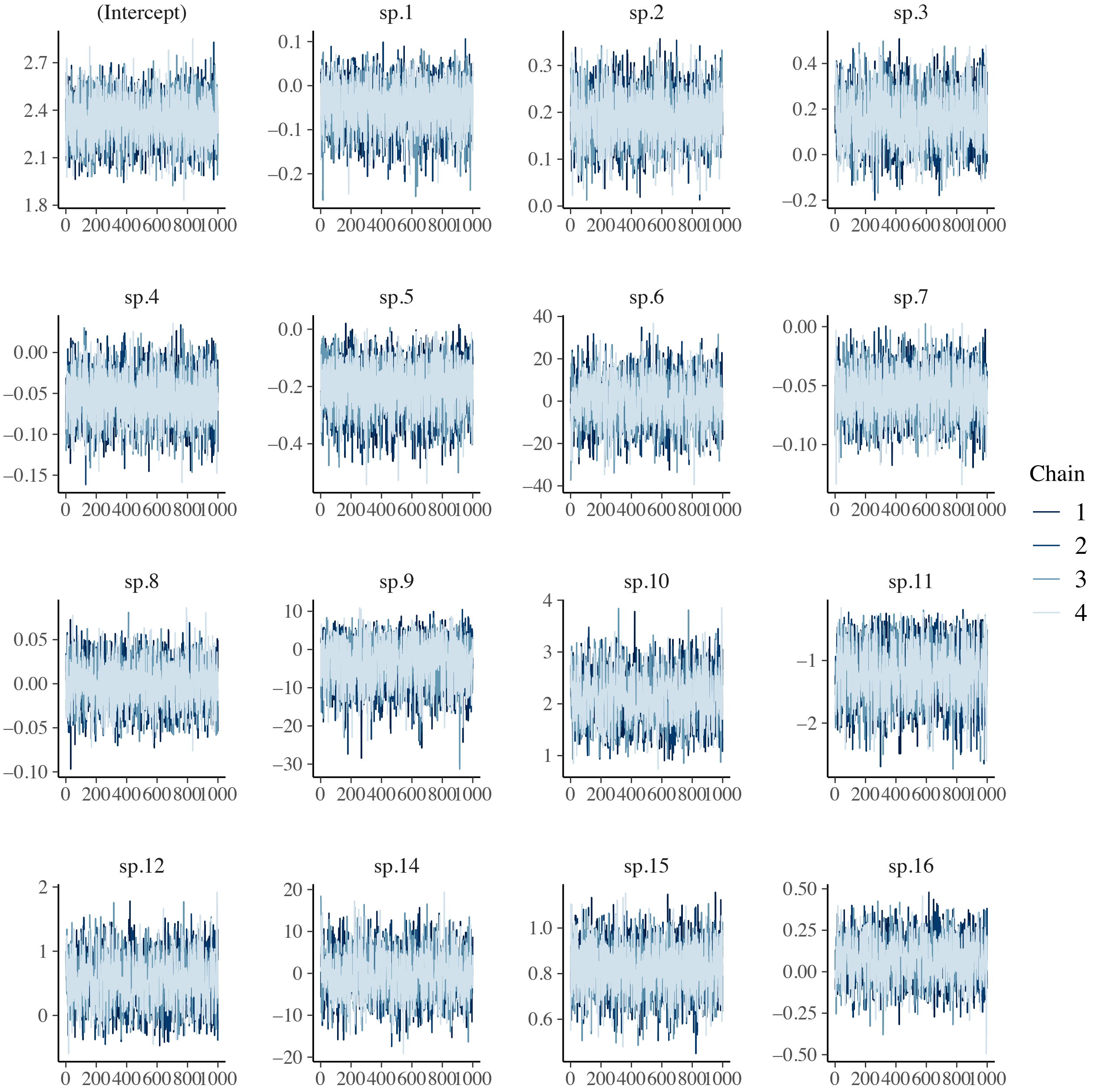
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```


(Intercept)
−20
−15
−10
−5
0
sp.1
−4
−2
0
sp.2
−5.0
−2.5
0.0
2.5
sp.3
−10
−5
0
sp.4
−1.5
−1.0
−0.5
0.0
0.5
1.0
sp.5
−3
−2
−1
0
1
sp.6
−50
−25
0
25
sp.7
−1.0
−0.5
0.0
0.5
sp.8
−2
−1
0
1
sp.9
−20
−10
0
10
sp.10
−30
−20
−10
0
10
sp.11
−2
−1
0
sp.13
−0.4
−0.2
0.0
0.2
sp.14
−20
−10
0
10
20
sp.15
−20
−10
0
sp.16
−1
0
1
2
3

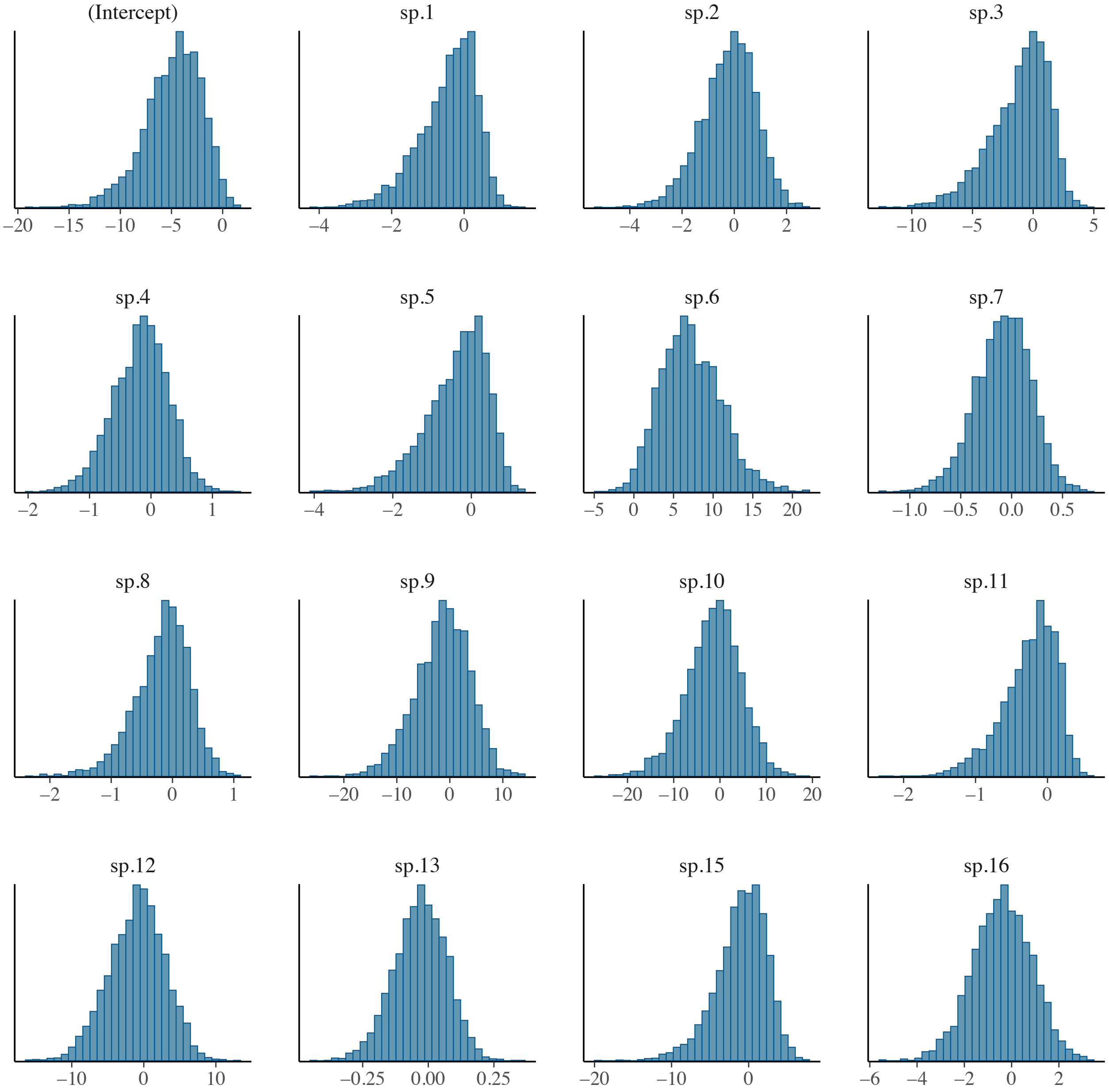


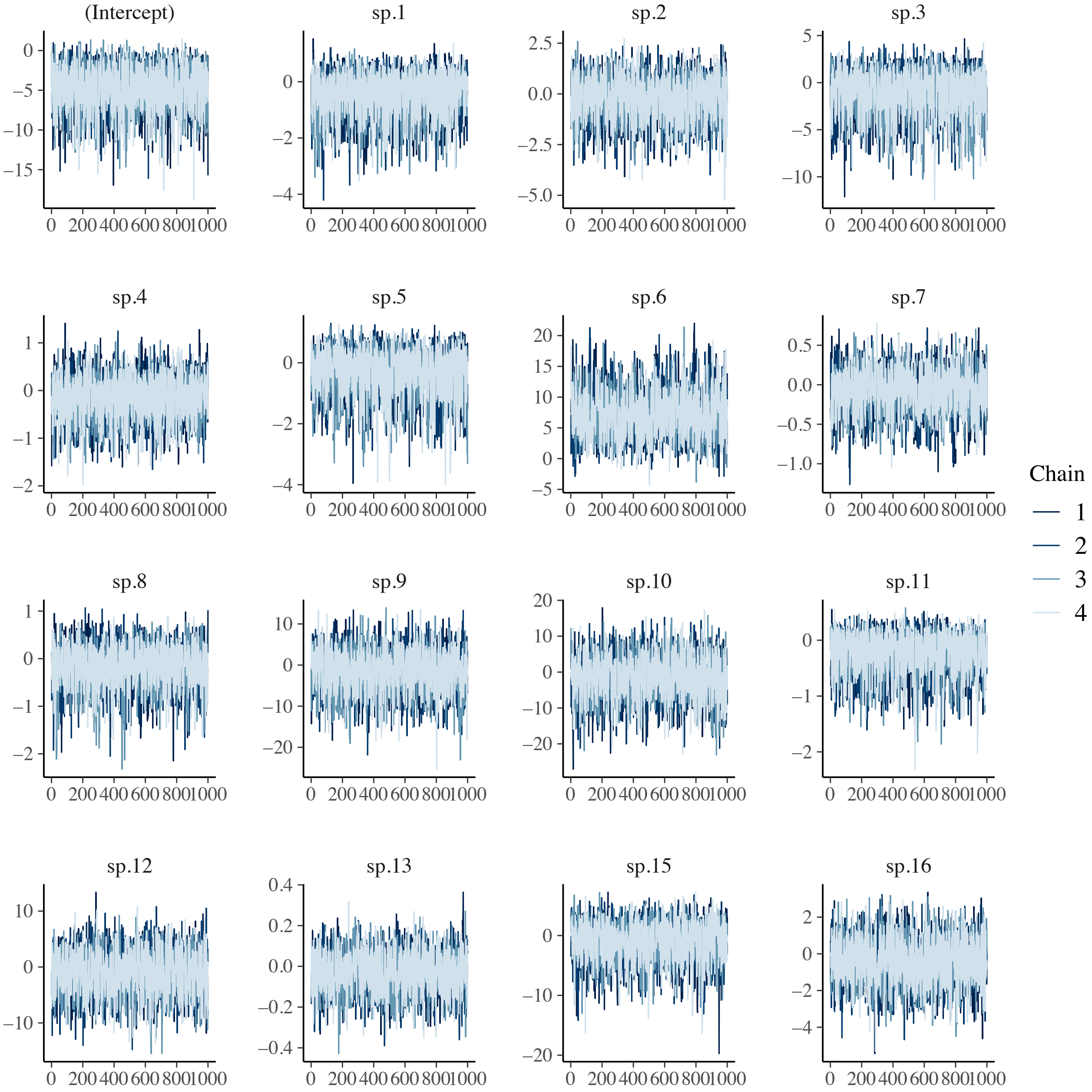
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```


(Intercept)
2.00 2.25 2.50 2.75
sp.1
−0.2 −0.1 0.0 0.1
sp.2
0.0 0.1 0.2 0.3
sp.3
−0.2 0.0 0.2 0.4
sp.4
−0.15 −0.10 −0.05 0.00
sp.5
−0.4 −0.2 0.0
sp.6
−40 −20 0 20 40
sp.7
−0.10 −0.05 0.00
sp.8
−0.10 −0.05 0.00 0.05 0.10
sp.9
−30 −20 −10 0 10
sp.10
1 2 3 4
sp.11
−2 −1
sp.12
0 1 2
sp.14
−20 −10 0 10 20
sp.15
0.4 0.6 0.8 1.0
sp.16
−0.50 −0.25 0.00 0.25 0.50

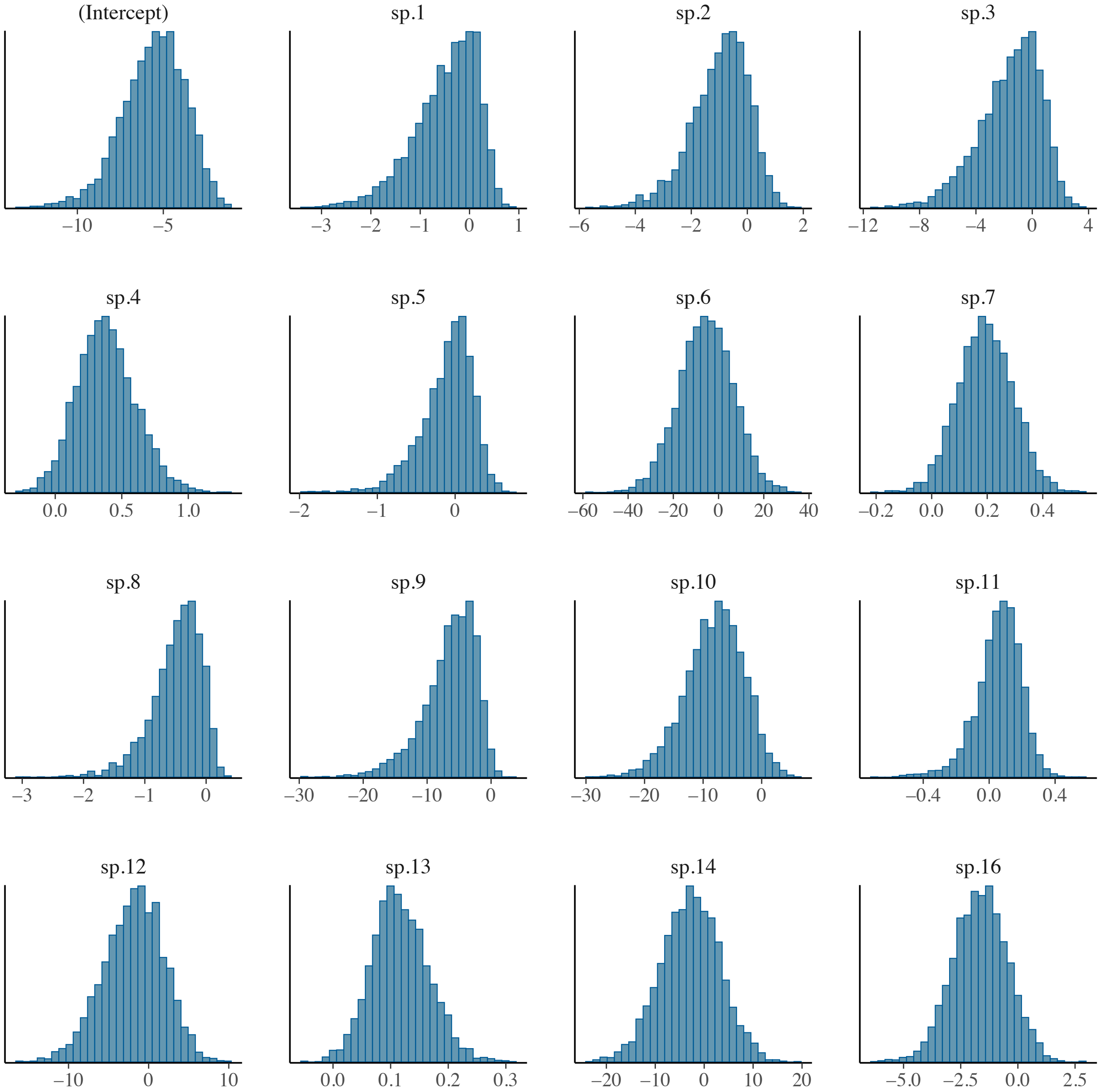


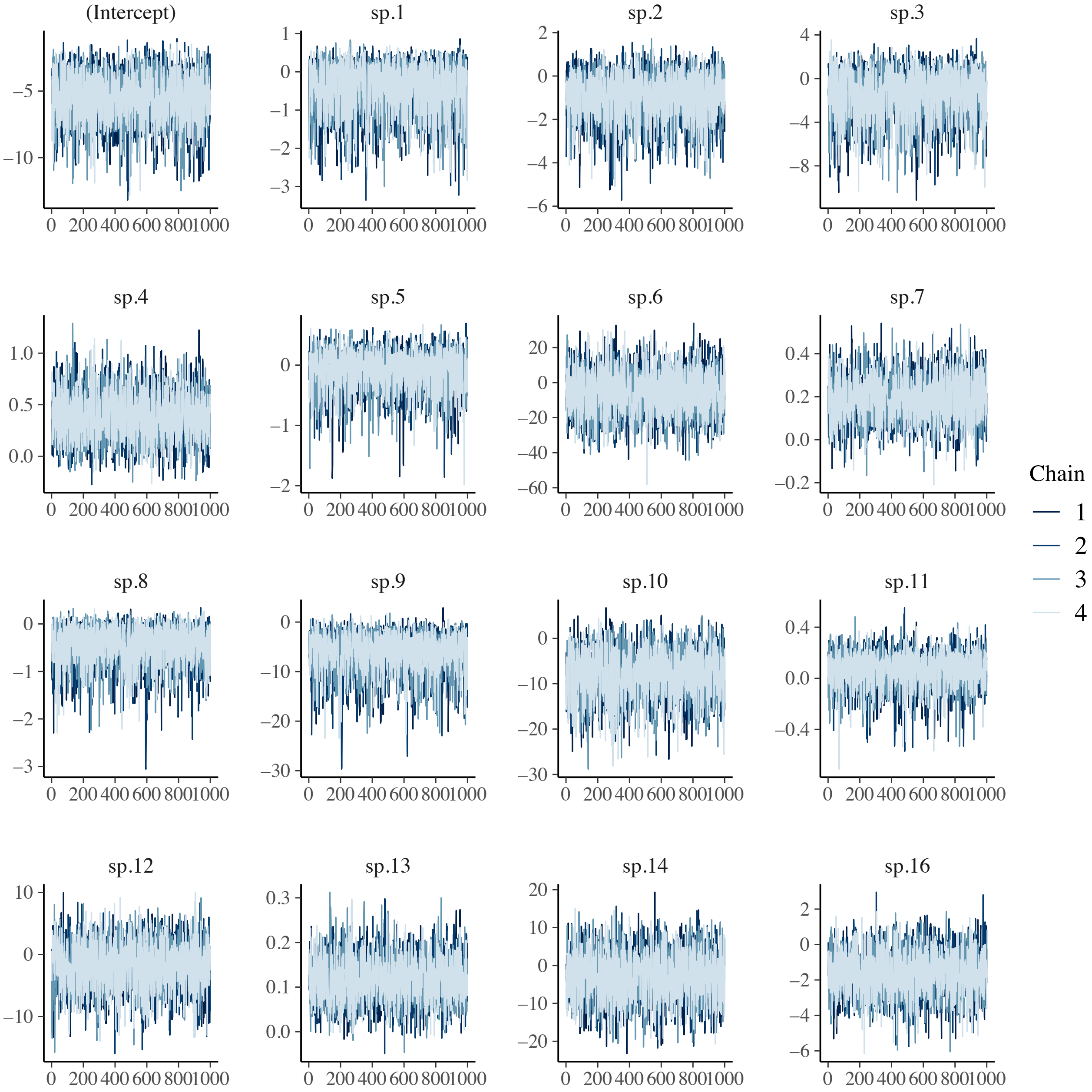
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```


(Intercept)
−20
−15
−10
−5
0
sp.1
−4
−2
0
sp.2
−4
−2
0
2
sp.3
−10
−5
0
5
sp.4
−2
−1
0
1
sp.5
−4
−2
0
sp.6
−5
0
5
10
15
20
sp.7
−1.0
−0.5
0.0
0.5
sp.8
−2
−1
0
1
sp.9
−20
−10
0
10
sp.10
−20
−10
0
10
20
sp.11
−2
−1
0
sp.12
−10
0
10
sp.13
−0.25
0.00
0.25
sp.15
−20
−10
0
sp.16
−6
−4
−2
0
2

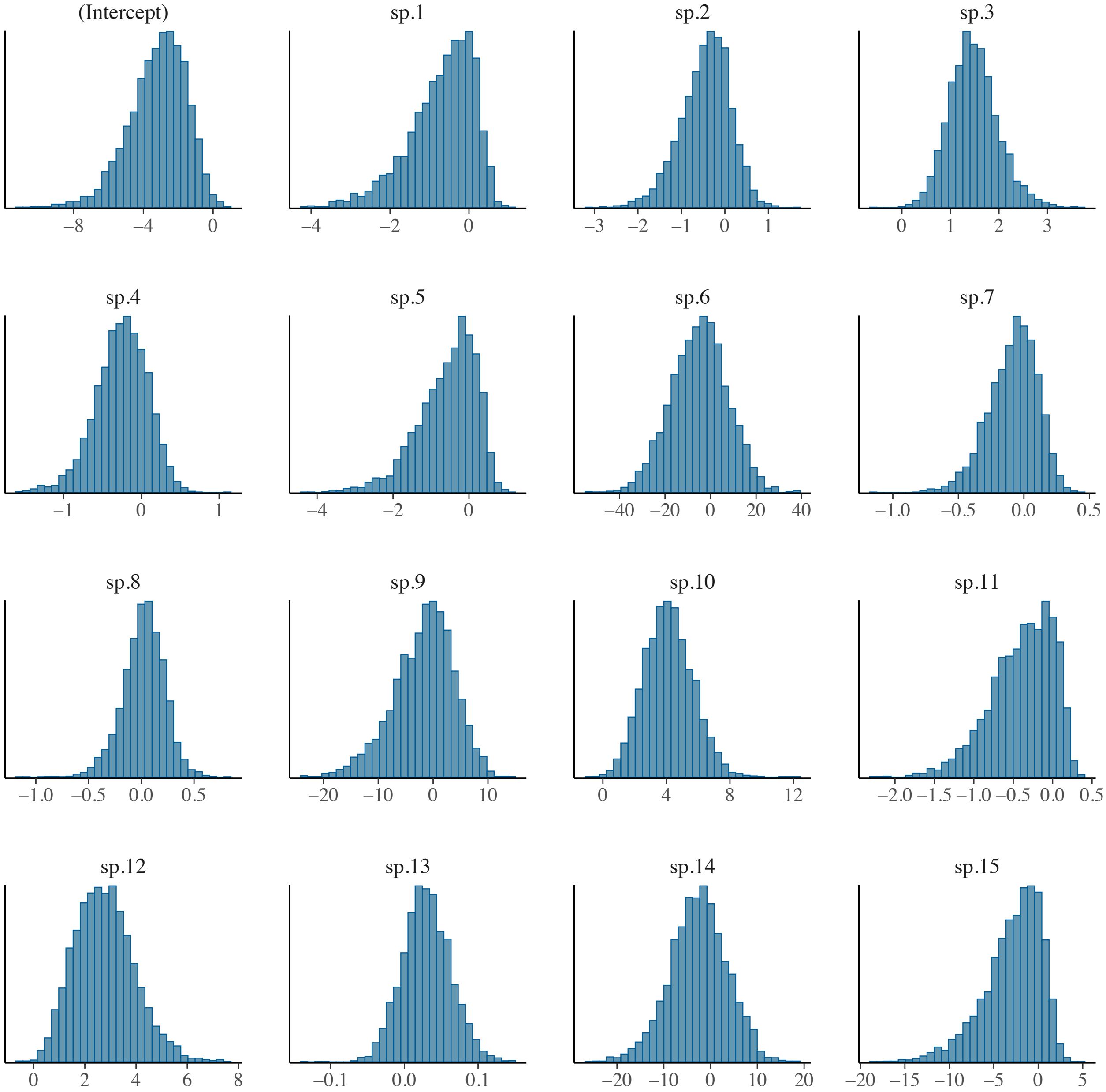


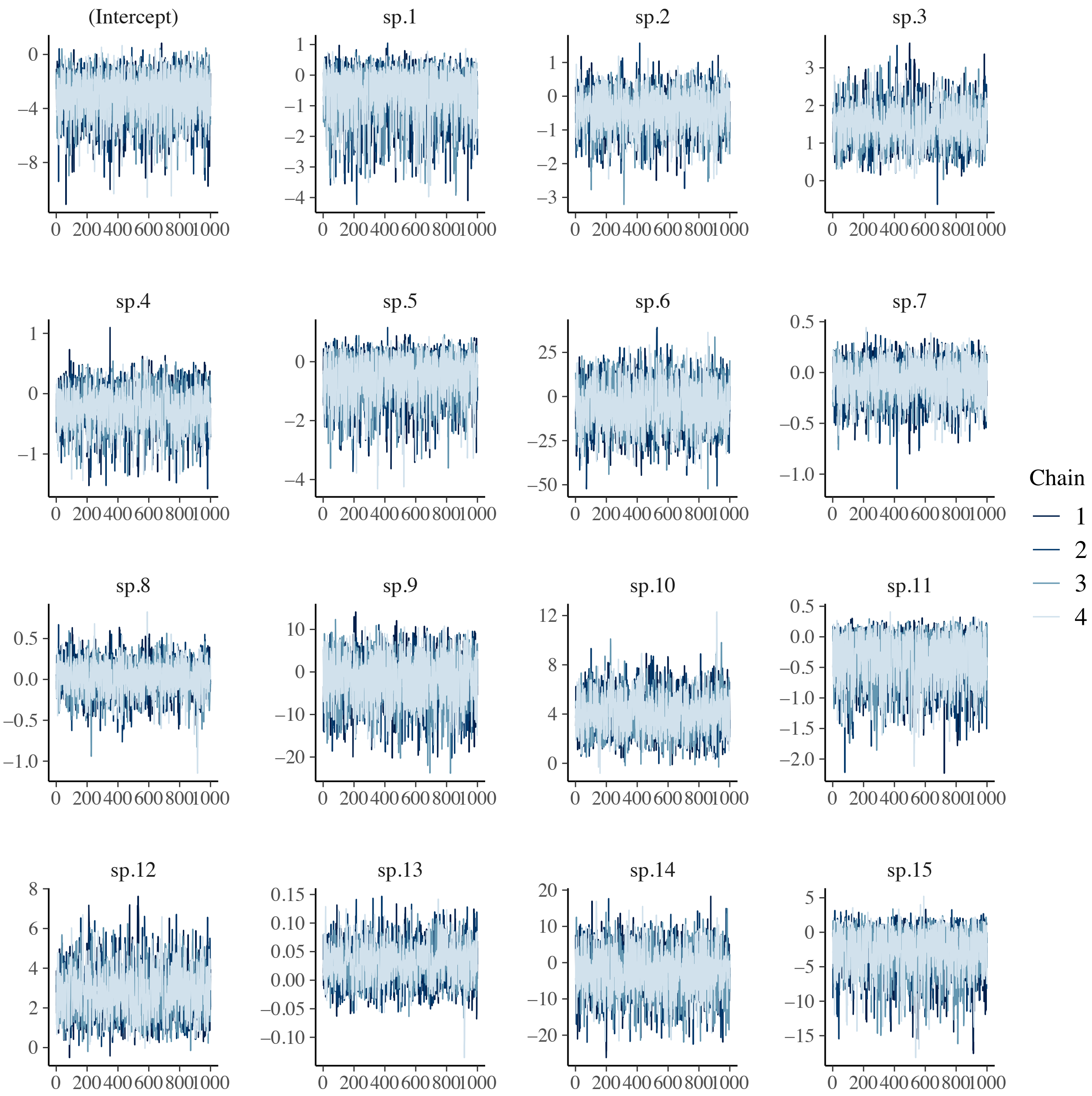
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```


(Intercept)
−10
−5
sp.1
−3
−2
−1
0
1
sp.2
−6
−4
−2
0
2
sp.3
−12
−8
−4
0
4
sp.4
0.0
0.5
1.0
sp.5
−2
−1
0
sp.6
−60
−40
−20
0
20
40
sp.7
−0.2
0.0
0.2
0.4
sp.8
−3
−2
−1
0
sp.9
−30
−20
−10
0
sp.10
−30
−20
−10
0
sp.11
−0.4
0.0
0.4
sp.12
−10
0
10
sp.13
0.0
0.1
0.2
0.3
sp.14
−20
−10
0
10
20
sp.16
−5.0
−2.5
0.0
2.5



```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```


(Intercept)
−8
−4
0
sp.1
−4
−2
0
sp.2
−3
−2
−1
0
1
sp.3
0
1
2
3
sp.4
−1
0
1
sp.5
−4
−2
0
sp.6
−40
−20
0
20
40
sp.7
−1.0
−0.5
0.0
0.5
sp.8
−1.0
−0.5
0.0
0.5
sp.9
−20
−10
0
10
sp.10
0
4
8
12
sp.11
−2.0
−1.5
−1.0
−0.5
0.0
0.5
sp.12
0
2
4
6
8
sp.13
−0.1
0.0
0.1
sp.14
−20
−10
0
10
20
sp.15
−20
−15
−10
−5
0
5


(Intercept)
sp.1
sp.2
sp.3
sp.4
sp.5
sp.6
sp.7
sp.8
sp.9
sp.10
sp.11
sp.12
sp.13
sp.14
sp.15
Chain
1
2
3
4
0
200
400
600
800
1000

Dynamic model

We address how prevailing network motifs contribute to species persistence using a system of ordinary differential equations derived form a multispecies Lotka-Volterra model (Case 2000, Stouffer *et al.* 2010, Grilli *et al.* 2017b, Saavedra *et al.* 2017).

We consider the four three-species network modules in isolation, which are resebmled by the matrices in equations 4, 5, 6, 7. We generate 100 networks for each module. All simulations are run to 50 model time-steps. We measure community persistence as the fraction of species that survive at each time-step across the modules.

```r
# intransitive competition
mat1 <- matrix(c(0,0,-1,-1,0,0,0,-1,0),3,3)
# module 1
mat2 <- t(matrix(c(0,0,0,1,0,-1,1,-1,0),3,3))
# module 2
mat3 <- matrix(c(0,0,0,-1,0,1,-1,1,0),3,3)
# module 3
mat4 <- matrix(c(0,-1,-1,0,0,1,0,1,0),3,3)
matmodules<-list(mat1,mat2,mat3,mat4)
# some pars
nrun=100
tst=50
time <- seq(0,tst,by=1)

# store data here
mechmod1.list<-rep(list(rep(list(NA),nrun)),4)

## run
for(i in 1:4){
  m0<-matmodules[[i]]
  npos<-sum(m0[which(m0==1)])
  nneg<-abs(sum(m0[which(m0==-1)]))
  for(j in 1:nrun){
    print(j)
    m0<-matmodules[[i]]
    # sample randomly r and k for each species
    kpar<-rtruncnorm(3, 1, 10, 5, 1)
    rpar<-rtruncnorm(3, 0.1, 1, 0.5, 0.1)
    bA<- rep(1,3)
    names(bA) = NULL
    # assign alpha randomly from observed alphas
    aijpos<-sample(aij_o_cut[which(aij_o_cut>0)], npos)
    aijneg<-sample(aij_o_cut[which(aij_o_cut<0)], nneg)
    # assign alpha pos e neg
    m0[which(m0==1)]<-aijpos
```

```
    m0[which(m0==-1)]<-aijneg
    diag(m0)<- -3
    # rescaling alpha to A_ij
    for(kk in 1:3){
        m0[kk,] = m0[kk,]*rpar[kk]/kpar[kk]
    }
    ###
    parms=c(rpar=rpar, m0=m0)
    N = rep(0,3)
    state <- c(N = bA)
    mod1<-function(time, N, parameters){
        dN <- N*(rpar+as.vector(t(m0 %*% N)))
        list(c(dN))
    }
    # ode model
    mechmod<-ode.1D(y=state,
                    times=time,
                    func = mod1,
                    parms = parameters,
                    nspec = 1)
    # export results
    mechmod1.list[[i]][[j]]<-
      apply(mechmod[-1,-1],1, function(x) length(which(x>1)))/3
}}

# save data here
mechmod1.df<-data.frame(time=rep(1:tst,nrun*4),
                        nrun=gl(nrun,tst,nrun*(tst)*4),
                        module=gl(4,nrun*tst, 4*nrun*tst),
                        persist=NA)

for(i in 1:4){
    mechmod1.df$persist[which(as.numeric(mechmod1.df$module)==i)]<-
      unlist(mechmod1.list[[i]])
}
```

We use generalized additive models (`bam` function in `mgcv`; Wood *et al.* 2015) to analyse the effects of three-species associations on community persistence. Time was included as a smooth function. Results indicate significant differences in community persistence among the the four network modules. In particular, the fraction of surviving species is the highest in module 1 and module 3, while it is lowest in intransitive competition.

```
# with AR1
simdat <- start_event(mechmod1.df, column="time", event=c("module", "nrun"))
# autocorrelation
```

```r
mod.mech1 <- bam(persist ~ module + s(time, by=module),
                 AR.start=simdat$start.event, data=mechmod1.df)

anova.gam(mod.mech1)
```

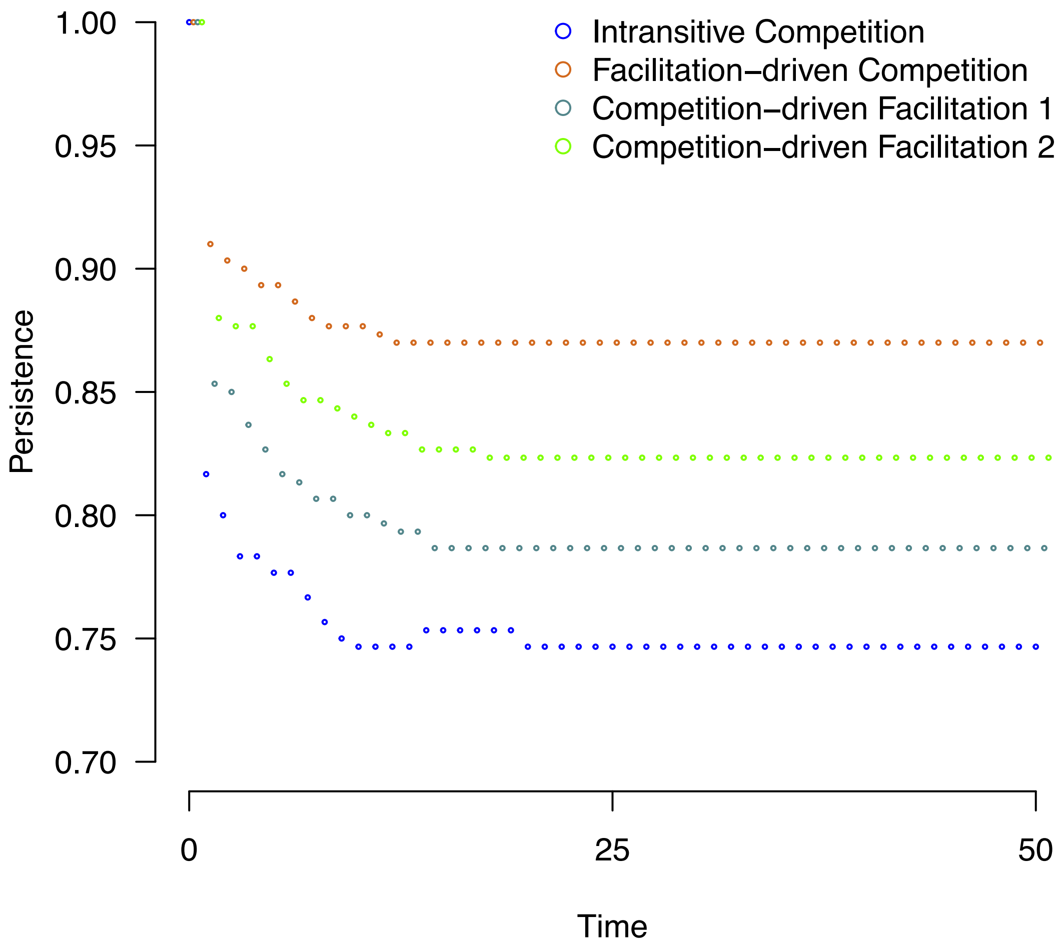
```
##
## Family: gaussian
## Link function: identity
##
## Formula:
## persist ~ module + s(time, by = module)
##
## Parametric Terms:
##        df     F p-value
## module  3 375.3  <2e-16
##
## Approximate significance of smooth terms:
##                    edf Ref.df     F  p-value
## s(time):module1 3.187  3.960 5.906 9.09e-05
## s(time):module2 2.264  2.822 2.851   0.0296
## s(time):module3 3.289  4.084 7.768 2.52e-06
## s(time):module4 2.971  3.695 7.204 1.81e-05
```

```r
summary(mod.mech1)
```

```
##
## Family: gaussian
## Link function: identity
##
## Formula:
## persist ~ module + s(time, by = module)
##
## Parametric coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 0.753471   0.002657  283.60   <2e-16 ***
## module2     0.120643   0.003757   32.11   <2e-16 ***
## module3     0.040761   0.003757   10.85   <2e-16 ***
## module4     0.077338   0.003757   20.58   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Approximate significance of smooth terms:
##                    edf Ref.df     F  p-value
## s(time):module1 3.187  3.960 5.906 9.09e-05 ***
## s(time):module2 2.264  2.822 2.851   0.0296 *
```

```
## s(time):module3 3.289  4.084 7.768 2.52e-06 ***
## s(time):module4 2.971  3.695 7.204 1.81e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =  0.0572   Deviance explained = 5.79%
## fREML =  -5022  Scale est. = 0.03528   n = 20000
```

```r
# plot model 1
modpch<-c(15:18)
modcol<-c("blue","chocolate","cadetblue4","chartreuse")

plot(0,0,type="n",xlim=c(0,50), ylim=c(0.70,1),
     xaxt="n",xlab="Time",ylab="Persistence",yaxt="n", axes=F)
axis(1,at=c(0,25,50), las=1)
axis(2,at=c(0.7,0.75,0.8, 0.85, 0.9, 0.95, 1), las=1)
legend("topright",bty="n", pch=1, col=modcol,
legend=c("Intransitive Competition",
"Facilitation-driven Competition",
"Competition-driven Facilitation 1",
"Competition-driven Facilitation 2"))
points(0,1,pch=1, cex=0.3, col=modcol[1])
points(0+1/4,1,pch=1, cex=0.3, col=modcol[2])
points(0+2/4,1,pch=1, cex=0.3, col=modcol[3])
points(0+3/4,1,pch=1, cex=0.3, col=modcol[4])
for(i in 1:4){for(j in 1:tst){
  mu<-mean(mechmod1.df$persist[which(as.numeric(mechmod1.df$module)==i&
                                     as.numeric(mechmod1.df$time)==j)])
  points(j+(i-1)/4, mu, pch=1, cex=0.3, col=modcol[i])
}}
```


1.00
0.95
0.90
0.85
0.80
0.75
0.70
Persistence
0
25
50
Time
Intransitive Competition
Facilitation−driven Competition
Competition−driven Facilitation 1
Competition−driven Facilitation 2