

ECOG-03315.

Zurell, D., Pollock, L. J. and Thuiller, W. (2018), Do joint species distribution models reliably detect interspecific interactions from co-occurrence data in homogenous environments? *Ecography*. Accepted author manuscript. doi: 10.1111/ecog.03315

Supplementary material

- Additional figures A1-A5
- R code for simulating species co-occurrence following (Araújo and Rozenfeld 2014)

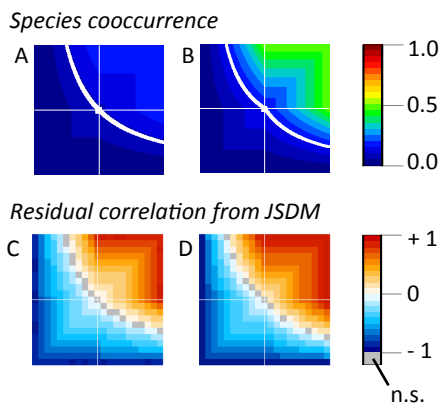


Figure A1. Expected co-occurrence (A-B) and residual correlation (estimated by JSJM; C-D) across the biotic interaction space of two species A and B with unequal prevalence. (A,C) Species A has prevalence of 0.1 and species B has prevalence of 0.5. (B,D) Species A has prevalence of 0.2 and species B has prevalence of 0.4. For further information, please see Figure 1 in main text.

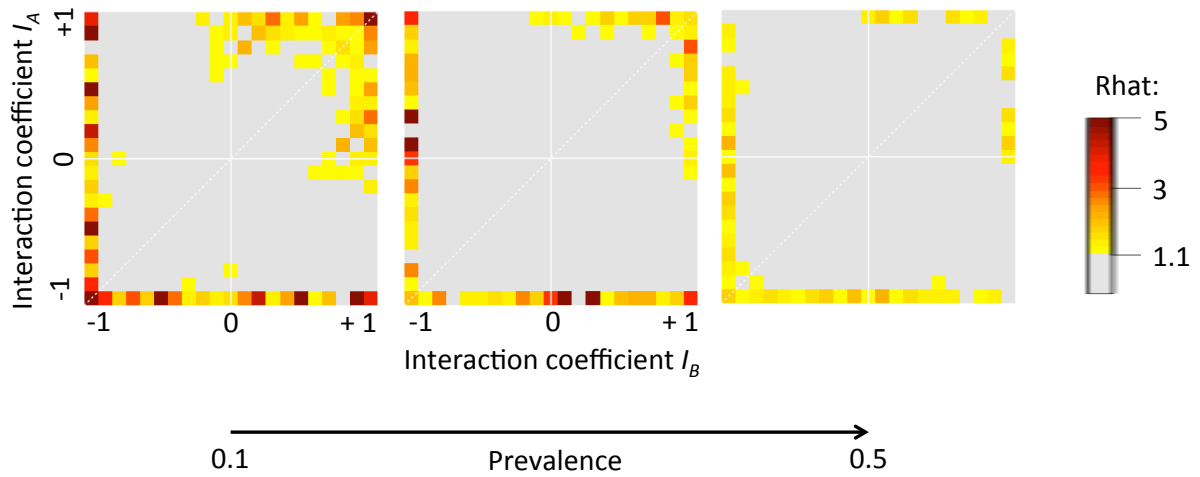


Figure A2. Rhat values obtained for default runs of 10000 iterations and 3 chains. Convergence is achieved when chains are well mixed with rhat values below 1.1 (shaded in grey). Species A and B had equal prevalence of 0.1, 0.3 and 0.5, respectively.

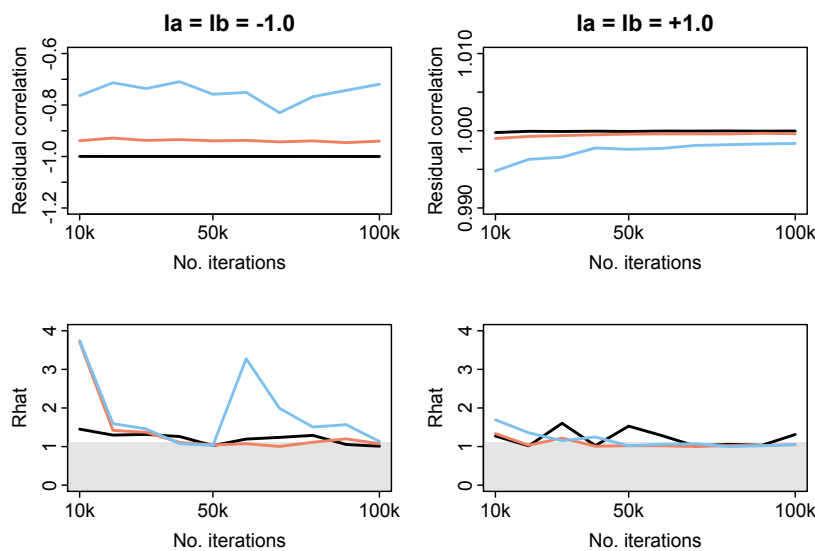


Figure A3. Residual correlation (top) and rhat values (bottom) for increasing number of iterations. Shown are the extreme cases of negative-negative interactions (left column) and positive-positive interactions (right column) with maximum interaction strength. These maximum interactions strengths proved to result in more problematic rhat values for the default of 10000 iterations (cf. Fig. S2). The grey area in the lower panels shows the rhat level of max. 1.1 indicating convergence. The colours represent different prevalence levels, with both species A and B having equal prevalence (blue: 0.1; red: 0.3; black: 0.5). Residual correlations remain comparably stable irrespective of number of iterations whereas rhat values vary and do not show a strict monotonic decrease with increasing iterations.

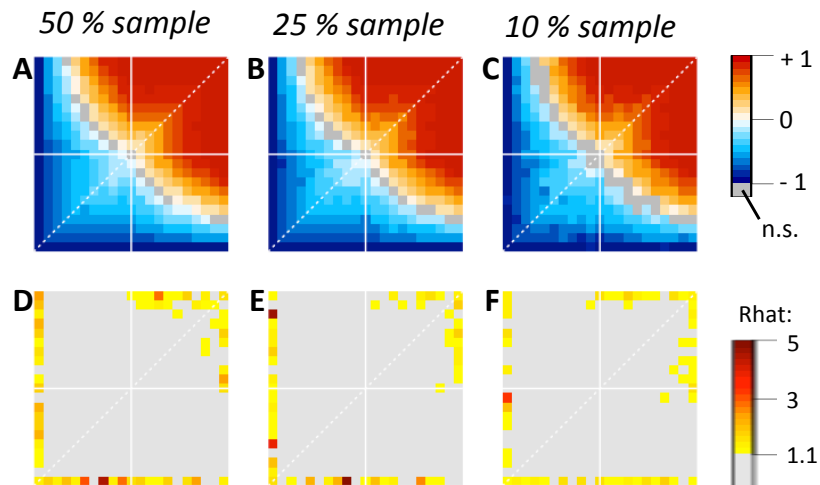


Figure A4. Effect of sample size on residual correlations estimated by JSdMs. Both species A and B have prevalence of 0.3. (A-C) show the residual correlations estimated by JSdMs for decreasing sample sizes (A: $n=5000$, meaning 50% of landscape is sampled, B: $n=2500$, meaning 25% of the landscape, C: $n=1000$, meaning 10% of the landscape). (D-F) show the corresponding R_{hat} values. For simplicity, we only present one realisation for all combinations of interaction coefficients I_A and I_B (please refer to Figure 1 in main text for more information on axes).

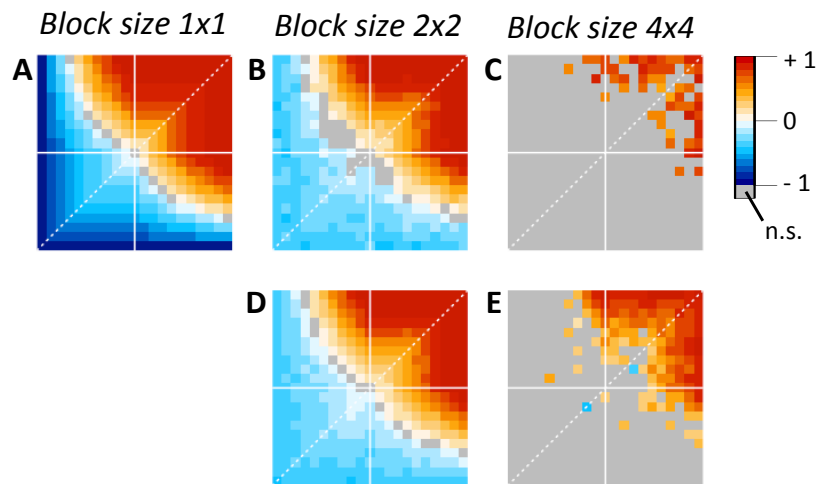


Figure A5. Effect of sample size on scale dependence of residual correlations estimated by JSdMs. Both species A and B have prevalence of 0.3. (A-C) show the residual correlations estimated by JSdMs if the original landscape of 100×100 cells is aggregate consecutively (A: original landscape with 100×100 cells, B: 50×50 cells, C: 25×25 cells). (D-E) show the residual correlations estimated by JSdMs for aggregated data with a constant landscape size of 100×100 cells. For simplicity, we only present one realisation for all combinations of interaction coefficients I_A and I_B (please refer to Figure 1 in main text for more information on axes).

R CODE FOR CO-OCCURRENCE SIMULATION

```
# code for simulating co-occurrence of two interacting species following Araujo &
Rozenfeld (2014)
```

```
sim.Dist = function(L=100,rhoA=0.3,rhoB=0.3,Ia=0,Ib=0,output='all',agg=NULL) {
```

```
# currently not considered: spatial autocorrelation, exclusive ranges
# output = 'all' include plotting of landscape
# output = 'cooccAB' only outputs the co-occurrence index
# output = 'cooccAB.null' outputs co-occurrence index and a T/F value indicating
correspondence of rhoAB with null expectation
```

```
land = matrix(1,nrow=L,ncol=L)
```

```
#-----
-----
```

```
n.A.cur = 0          # current number of ind A in landscape
n.A.target = floor(rhoA * L^2)      # expected number of ind A in landscape given
prevalence rhoA
```

```
n.A.ToTarget = n.A.target - n.A.cur
```

```
# randomly distribute species A
land[sample(seq_len(prod(dim(land)))[land==1],n.A.ToTarget)] = 2
```

```
#-----
-----
```

```
n.B.cur = 0
n.B.target = floor(rhoB * L^2)
```

```
#-----
-----
```

```
# calculate amount of co-occurrence cells according to eqs. 3-5
```

```
rhoAB.null = rhoA * rhoB
```

```
if (Ia > 0) {
```

```
  if (Ib > 0) {
    # (+/+)
    rhoAB = rhoA * rhoB + max(Ia,Ib) * (min(rhoA,rhoB) - rhoA*rhoB )
```

```
  } else {
    # (+/-)
    rhoAB = (rhoA * rhoB + Ia * (min(rhoA,rhoB) - rhoA * rhoB) ) * (1+Ib)
```

```
  }
```

```
} else {
```

```

if (Ib > 0) {
  # (-/+ )
  rhoAB = (rhoA * rhoB + Ib * (min(rhoA,rhoB) - rhoA * rhoB) ) * (1+Ia)

} else {
  # (-/-)
  rhoAB = rhoA * rhoB * (1 + min(Ia,Ib))
}
}

n.AB.target = rhoAB * L^2

#-----
# randomly sample co-occurrence cells and cells only occupied by species B

land[sample(seq_len(prod(dim(land))))[land==2],n.AB.target]] = 4

n.B.cur = sum(land==4)
n.B.ToTarget = n.B.target - n.B.cur

# randomly distribute species B
land[sample(seq_len(prod(dim(land))))[land==1],n.B.ToTarget]] = 3
cooccAB = sum(land==4)/(sum(land==3)+sum(land==2)+sum(land==4))

#-----
# spatial dependence?

if(!is.null(agg)) {

  require(raster)

  agg.land = function(agg.y) {
    land.r = raster(land,xmn=0,xmx=L,ymn=0,ymx=L)

    aggregate(land.r,agg.y,function(x,...){ifelse(4%in%x,4,ifelse(2%in%x&3%in%x,4,max(x)
    )))}
  }

  agg.land.true = function(agg.y) {
    land.r = raster(land,xmn=0,xmx=L,ymn=0,ymx=L)
    aggregate(land.r,agg.y,function(x,...){ifelse(4%in%x,4,max(x))})
  }

  agg.l = lapply(agg,function(agg.y){agg.land(agg.y)})
  names(agg.l)=paste0('agg',agg)

  agg.l.true = lapply(agg,function(agg.y){agg.land.true(agg.y)})
  names(agg.l.true)=paste0('agg',agg)

```

```

}

#-----
-----
# output

if (output=='all') {
  # require(fields)
  #
  print(image.plot(land,col=c('white','grey60','grey30','red'),breaks=c(.5,1.5,2.5,3.5,4.5
),axes=F,axis.args=list(at=1:4,labels=c('Empty','A','B','A & B'))))
  if (is.null(agg)) return(list(rhoAB=rhoAB,rhoAB.null=rhoAB.null,coocc=
cooccAB,land=land)) else
  return(list(rhoAB=rhoAB,rhoAB.null=rhoAB.null,coocc= cooccAB,land=land,
land.agg=lapply(agg.l,function(x){data.frame(coordinates(x),cells=values(x))}),land.agg.
true=lapply(agg.l.true,function(x){data.frame(coordinates(x),cells=values(x))}))) } else
if (output=='cooccAB') return(cooccAB) else
if (output=='cooccAB.null') return(list(coocc=cooccAB,null= rhoAB/rhoAB.null))
}

```

```

examp = sim.Dist(rhoA=0.3,rhoB=0.3,Ia=.5,Ib=.5,output='all',agg=c(2,4))

```