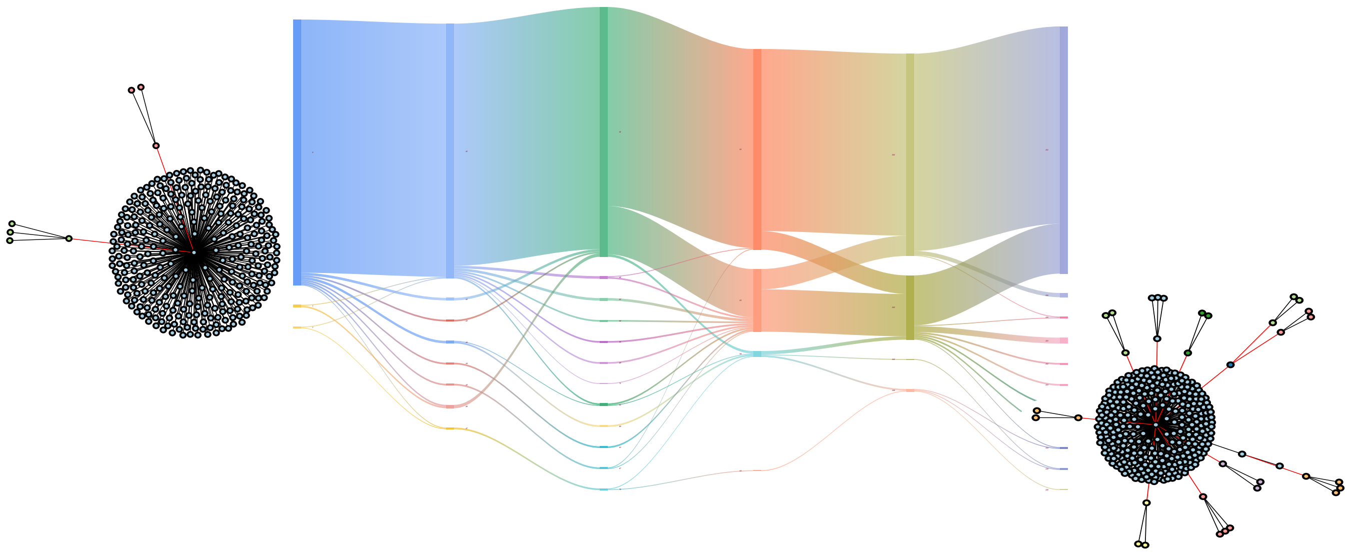
Opportunistic attachment assembles plant-pollinator networks: A walk through of the analysis

Lauren Ponisio

May 18, 2017



1 Overview

In our study we examin the temporal dynamics of plant-pollinator network assembly using variety of different methods including 1) network change point detection 2) node/species-level position variation, 3) species and interaction turnover 4) network-level metrics and 5) extinction simulations. We are committed to reproducible science and all analytical code will be maintained on github, along with this write up.

The entire analysis is executable from the main.sh file. All of the packages needed to run the analyses are listed in the packages.sh file, including the specific versions of the python dendropy library, which is very finicky. We have had no problems with backward compatibility for the R scripts (in terms of R versions from v. 2-3.4 or packages) but the change point analysis must be run in python2 (not easily convertible to python3, though this is potentially in progress).

Navigate to the analysis folder within the github repo (hedgerow_assembly) then the main.sh file can be selected and run. I would not recommend doing this unless you are prepared to have two cores of your machine running for days to weeks (depending on how many of iterations of the change point analysis specific in mainChangePoint.sh loop), but you could run of the analyses in the study by running this line in BASH.

```
bash main.sh
```

This will somewhat helpfully print the results of each analysis and re-create any accompanying figures.

We will walk through each the main script for each analysis individually.

2 Change Point Analysis

The change point analysis which is a mix of python and R. hedgerows.py (the meat of the change point analysis) runs in parallel on two cores (which can be modified in the hedgerows.py script by toggling the ncores object), but will likely take a many hours (a day on running on my 2.5 GHz Intel Core i7 mac pro) depending on your machine.

There is some stochasticity in the change point analysis in the hypothesis testing step where samples are drawn from the null distribution and the likelihood of a change point occurring is calculated from those samples. Because of this, we re-ran the change point detection analysis 1000 times (this was done on a cluster) and used the change points that were identified in 95% of runs in subsequent analyses.

mainChangePoint.sh compiles the python code, then creates the .pairs files needed for the change point analysis (dataPrep.R). Nodes names need to be consistent across years within a site, and for simplicity we made them consistent across all sites as well. This enabled us to provide a single .lut master list of the nodes to the analysis.

Next, the change point detection loop is run. This function begins by fitting the generalized hierarchical random graph model (GHRG) which outputs consensus trees that are used by the change point detection algorithm. There is only slight variability in the fitting of the GHRG between runs, so once the first loop is run, the consensus trees are used in subsequent runs.

The output of the change point analysis is saved as a list of the change points identified and likelihood scores (formatted by prepChangePointOutput.R).

After the loop, consensusChangePoints.R drops the change points that were not identified by 95% of the runs and then creates consensus trees for each chunk of years between change point in python (postChangePoint.py). The outputs of the change point analysis were very specialized to that analysis, so we had to convert them to .glm files for plotting and further exploration in R (convertConsensusTrees.py)

The plotting and statistical analyses are in R in different scripts. The plotting function will create the network, community and flow plots (networks.R). The binomial linear mixed models are run in the comparison.R script, along with addition statistical analysis requested by the reviewers testing whether change points tended to occur between years with a larger difference in samples.

```
mod.chpt <- glm(cbind(chpt.trial$chpts,
                      chpt.trial$trial - chpt.trial$chpts) ~
    chpt.trial$status, family="binomial")
print(summary(mod.chpt))
```

```
##
## Call:
## glm(formula = cbind(chpt.trial$chpts, chpt.trial$trial - chpt.trial$chpts) ~
##     chpt.trial$status, family = "binomial")
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -1.67612  -0.46233  -0.42803  -0.09731   1.67955
##
## Coefficients:
##                            Estimate Std. Error z value Pr(>|z|)
## (Intercept)                 -1.5041     0.4513  -3.333 0.000861 ***
## chpt.trial$statusmature     -0.8473     0.8668  -0.978 0.328315
## chpt.trial$statuscontrol    -2.6703     1.1041  -2.418 0.015585 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 21.029  on 20  degrees of freedom
## Residual deviance: 12.035  on 18  degrees of freedom
## AIC: 31.654
##
## Number of Fisher Scoring iterations: 6
```

```
print(exp(cbind(coef(mod.chpt), confint(mod.chpt))))
```

```
## Waiting for profiling to be done...
```

```
##                                           2.5 %    97.5 %
## (Intercept)              0.22222222 0.082931016 0.5027608
## chpt.trial$statusmature  0.42857143 0.058657189 2.0812211
## chpt.trial$statuscontrol 0.06923077 0.003574624 0.4311991
```

In the data, site status ”maturing” corresponds to the assembling hedgerows (i.e., they are maturing). ”Mature” corresponds to the non-assembling hedgerows > 10 years old, and ”control” corresponds to the non-assembling, weedy field margins.

In this analysis the assembling hedgerows are set as the intercept, so weedy field margins (”control”) have significantly less change points than assembling hedgerows (”maturing”). Mature and maturing are about the same.

3 Characteristics of species that contribute to change points

The degree of each species was calculated from a larger dataset including observations across the northern central valley (approx. 18000 interaction records). These scores were rarefied by treating plants like sites in a traditional rarefaction (Winfree *et al.*, 2014). This helped to disentangle degree with the number of observations of each species. The number of observations of a species/the total number of samples at each site was then calculated as an estimate of persistence. For each species, its weighted closeness in the network is also calculated. All of these calculations were done from the dataPrep/dataPrep.R file which needs access to some non-publically available data, though the data needed for these analysis are all in github.

cv.R calculates the cv of closeness for each species at each site, and appends the values for persistence and rarefied degree. For the pollinators:

```
## ************************************************************
## pollinators
## ************************************************************

pol.cv <- calcCvTrait(spec, ## specimen data
                   specs[specs$speciesType =="pollinator",], ## network metrics
                   trait1="occ.date", ## first trait of interest, persistence
                   trait2="r.degree", ## second trait of interest
                   ## rarefied degree
                   time.col="assem", ## name of the time column
                   abund.col="weighted.closeness", ## network metric
                   ## of interest
                   cv.function=cv, ## function to use for calculating cv
                   zero2na=TRUE, ## don't convert NAs to zeros in cv calc
                   standard.cv=TRUE, ## log the cv
                   na.rm=TRUE,
                   species.type="GenusSpecies")
```

We can then use a linear mixed model to regress the cv of weighted closeness by rarefied degree and persistence. Random effects of site and species are included. Because degrees of freedom are not well-dinfed for a mixed model Satterthwaite approximations implemented in the lmeTest package are used so that p-values can be calculated for the linear mixed models (Kuznetsova *et al.*, 2014).

```
pol.mod <- lmer(formula.cv, data=pol.cv$lm.data)
print(summary(pol.mod))

## Linear mixed model fit by REML t-tests use Satterthwaite approximations
##   to degrees of freedom [lmerMod]
## Formula: cv ~ occ.date + r.degree + (1 | Site) + (1 | GenusSpecies)
##    Data: pol.cv$lm.data
##
## REML criterion at convergence: 1043.5
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -5.0709 -0.3940  0.1942  0.6762  1.6799
##
## Random effects:
##  Groups       Name        Variance Std.Dev.
##  GenusSpecies (Intercept) 0.01056  0.1028
##  Site         (Intercept) 0.05128  0.2265
##  Residual                 0.61513  0.7843
## Number of obs: 427, groups:  GenusSpecies, 66; Site, 20
##
## Fixed effects:
##              Estimate Std. Error        df t value Pr(>|t|)
## (Intercept)  3.696191   0.094045 36.350000  39.302  < 2e-16 ***
## occ.date     0.741596   0.246251 15.420000   3.012  0.00855 **
## r.degree    -0.002708   0.001831 17.770000  -1.479  0.15669
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##          (Intr) occ.dt
## occ.date -0.095
## r.degree -0.451 -0.674
```

```
vif.mer(pol.mod)
```

```
## occ.date r.degree
## 1.834127 1.834127
```

```
## variance inflation factors < 2, so okay!!! (Zurr et al. 2010)
```

There is a significant position relationship between cv of closeness and persistence (”occ.date”) but not rarified degree (”r.degree”). The variance inflation factors are both before 2, which Zuur *et al.* (2010) suggests is acceptable.

We found no statistically significant relationships in the plant data.

```
## ************************************************************
## plants
## ************************************************************

plant.cv <- calcCvTrait(spec,
                     specs[specs$speciesType =="plant",],
                     trait1="occ.plant.date",
                     trait2="plant.r.degree",
                     time.col="assem",
                     abund.col="weighted.closeness",
                     cv.function=corCv,
                     zero2na=TRUE,
                     standard.cv=TRUE,
                     na.rm=TRUE,
                     species.type="PlantGenusSpecies")

plant.mod <- lmer(formula.plant.cv, data=plant.cv$lm.data)
print(summary(plant.mod))
```

```
## Linear mixed model fit by REML t-tests use Satterthwaite approximations
##   to degrees of freedom [lmerMod]
## Formula:
## cv ~ occ.plant.date + plant.r.degree + (1 | Site) + (1 | GenusSpecies)
##    Data: plant.cv$lm.data
##
## REML criterion at convergence: 99.7
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -3.6898 -0.3189  0.0615  0.4319  1.9013
##
## Random effects:
##  Groups       Name        Variance Std.Dev.
##  Site         (Intercept) 0.1071   0.3273
##  GenusSpecies (Intercept) 0.0000   0.0000
##  Residual                 0.2777   0.5270
## Number of obs: 50, groups:  Site, 18; GenusSpecies, 15
##
## Fixed effects:
##                  Estimate Std. Error        df t value Pr(>|t|)
## (Intercept)     4.865e+00  2.107e-01  4.614e+01  23.088   <2e-16 ***
## occ.plant.date -3.180e-01  6.459e-01  4.342e+01  -0.492    0.625
## plant.r.degree -6.915e-05  2.736e-03  4.299e+01  -0.025    0.980
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##             (Intr) occ.p.
## occ.plnt.dt -0.022
## plant.r.dgr -0.607 -0.684
```

```
vif.mer(plant.mod)
```

```
## occ.plant.date plant.r.degree
##       1.878196       1.878196
```

```
## variance inflation factors <1, so okay (Zurr et al. 2010)

## ***************************************************************
## the reviewers wanted a quantile regression, can only include on
## random effect at a time
```

You can play with using different network metrics to calculate the cv of. You can also change the function used for calculating the cv, or whatever function you would like to calculate between years.

```
 ## network metric options
 colnames(specs)[1:24]
```

```
##  [1] "degree"                        "normalised.degree"
##  [3] "species.strength"              "interaction.push.pull"
##  [5] "nestedrank"                    "PDI"
##  [7] "resource.range"                "species.specificity.index"
##  [9] "PSI"                           "node.specialisation.index.NSI"
## [11] "betweenness"                   "weighted.betweenness"
## [13] "closeness"                     "weighted.closeness"
## [15] "Fisher.alpha"                  "partner.diversity"
## [17] "effective.partners"            "proportional.generality"
## [19] "proportional.similarity"       "d"
## [21] "tot.int"                       "mean.k"
## [23] "sd.k"                          "k"
```

The reviewers suggested we also do quantile regressions, which can only include on random effect at a time, but yielded similar results.

```
## ***************************************************************
## the reviewers wanted a quantile regression, can only include on
## random effect at a time

## occ.date is significant
pol.lm.cv.quant <- lqmm(fixed=cv ~ occ.date + r.degree, random=~1,
                        group= GenusSpecies,
                        data=pol.cv$lm.data,
                        control=list(LP_max_iter=10^4))
pol.sum.boot.quant <- summary.boot.lqmm(boot(pol.lm.cv.quant,
                                             R=100))

## Quantile 0.5
##                   Value Std. Error lower bound upper bound Pr(>|t|)
## (Intercept)  3.7160065  0.0854433   3.5464685      3.8855  < 2e-16 ***
## occ.date     0.7040012  0.2454156   0.2170433      1.1910  0.00504 **
## r.degree    -0.0010938  0.0032266  -0.0074960      0.0053  0.73533
## scale        0.2869339  0.0185467   0.2501333      0.3237  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

4 Species and interaction turnover

4.1 Pollinators, plants and interactions

To calculate species and interaction turnover over, we borrow methods from the β-diversity world (Ponisio *et al.*, 2016; Chase *et al.*, 2011; Anderson *et al.*, 2006, 2011; Anderson, 2006). The goal is to calculate the pairwise dissimilarity between years within each site of the dataset using a dissimilarity estimator that incorporates species abundances, while also accounting for unobserved species (Chao *et al.*, 2005), and then following (Chase *et al.*, 2011), correct our estimates of β-diversity using null models.

The nulls.R script prepares null communities in variety of ways to calculate an expected distribution by generating randomized communities and calculating the dissimilarity of these communities. To do this, we defined the species pool within each site as the species and number of individuals present across all samples from that site. We then generated 999 random communities by constraining either 1) the total number of individuals caught at each site (individual nulls) or 2) the species richness at each site (alpha div nulls). We also had a binary option that constrained but marginal sums (occurrence nulls).

Next, in the function calcBetaStatus.R executed by beta-div.R, for the observed and nulls communities, we calculated the pair-wise dissimilarity between sites. We then used the null community dissimilarities to calculate the expected β-diversity when communities are randomly assembled but constrained so that they have either the same 1) number of individuals or 2) species richness as the observed communities and with species drawn from a meta-community with the same species abundance distributions. In order to do this, we followed Chase *et al.* (2011). Specifically, we calculated the fraction of randomly assembled communities with dissimilarity values less than (and half of those equal to) that of the observed community. We used this fraction as a “corrected dissimilarity score” for our observed data. Corrected dissimilarity values near one indicate that our observed communities exhibit more species turnover between sites than expected under a random assembly process while values near 0.5 indicate that our observed communities exhibit levels of turnover more in line with the null expectation. We calculated the corrected dissimilarities for each type of randomized community. We then generated principle coordinate axes (PCoA) based on the corrected pair-wise dissimilarities (Oksanen *et al.*, 2013; Anderson *et al.*, 2006, 2011; Anderson, 2006). We calculated dispersion for each site type by finding the centroid in PCoA space for that site type and then calculating the distances from sites of that type to that centroid using the betadisper function in the vegan packages (Oksanen *et al.*, 2013). The centroid is the point that minimizes the sum of these distances.

```
dis <- mapply(function(a, b, c, d)
    calcBetaStatus(comm= a, ## observed communities
                   status= b, ## vector of site types
                   dis.method, ## dissimilarity metric
                   nulls=c, ## null communities
                   occ=binary, ## binary or abundance weighted?
                   years=d, ## calculate beta div within?
                   sub=type,
                   zscore=FALSE), ## use Chase method not zscores
    a=comm$comm,
    b=comm$status,
    c= nulls,
    d= comm$comm,
    SIMPLIFY=FALSE)
```

Changing zscore to TRUE calculates dissimilarity zscores instead of Chase *et al.* (2011)'s pvalue-like method.

Toggling alpha and beta in the beta-div.R script changes between the null model options. When binary is TRUE, the binary null models are used and the dissimilarity method is ”jaccard” (all set up in the initialize_beta.R). When alpha is TRUE, species richness but not individual abundances are constrained, and the reverse when alpha is FALSE. The chao dissimilarity measure is used in both abundance-weighted cases. There is little difference between any of the options, except mature sites have significantly less turnover in pollinators with the alpha null options, which is likely most appropriate.

```
 binary <- FALSE
 alpha <- TRUE
```

Changing ”type” to ”pols”, ”plants”, or ”Ints” calculates turnover of pollinators, plants and interactions respectively. All of these options are executed in the main.sh script.

The dispersion values were then used in linear mixed-effect models to investigate the effect of different site types on ”β-diversity”. Site is included as a random effect. For pollinators:

```
mod <- lmer(dist ~ status +  (1|site),
            data=dats)
print(summary(mod))
```

```
## Linear mixed model fit by REML t-tests use Satterthwaite approximations
##   to degrees of freedom [lmerMod]
## Formula: dist ~ status + (1 | site)
##    Data: dats
##
## REML criterion at convergence: -173.6
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -2.5340 -0.7712  0.0072  0.6896  2.3481
##
## Random effects:
##  Groups   Name        Variance  Std.Dev.
##  site     (Intercept) 5.471e-20 2.339e-10
##  Residual             2.268e-02 1.506e-01
## Number of obs: 198, groups:  site, 38
##
## Fixed effects:
##                   Estimate Std. Error         df t value Pr(>|t|)
## (Intercept)       0.388962   0.012819 195.000000  30.342   <2e-16 ***
## statusmature     -0.050733   0.025639 195.000000  -1.979   0.0493 *
## statusmaturing   -0.008949   0.042240 195.000000  -0.212   0.8324
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
## Correlation of Fixed Effects:
##             (Intr) statsmtr
## statusmatur -0.500
## statusmtrng -0.303  0.152
```

Follow a reviewer's suggestion, we also included a linear model that took temporal auto-correlation into account, with the same results but not a large gain in fit given the additional parameters (tested using AIC)

```
mod.spatial <- lme(dist ~ status,
                   random = ~ 1 + cYear | site,
                   correlation=corAR1(form=~cYear),
                   data=dats,
                   control=list(maxIter=10^5, niterEM=10^5))

summary(mod.spatial)
```

```
## Linear mixed-effects model fit by REML
##  Data: dats
##         AIC       BIC   logLik
##   -160.7266 -134.5426 88.36329
##
## Random effects:
##  Formula: ~1 + cYear | site
##  Structure: General positive-definite, Log-Cholesky parametrization
##             StdDev     Corr
## (Intercept) 0.05875859 (Intr)
## cYear       0.01568352 -1
## Residual    0.14390959
##
## Correlation Structure: ARMA(1,0)
##  Formula: ~cYear | site
##  Parameter estimate(s):
##         Phi1
## -0.007385481
## Fixed effects: dist ~ status
##                      Value  Std.Error  DF   t-value p-value
## (Intercept)      0.3881307 0.01264276 160 30.699827  0.0000
## statusmature    -0.0478766 0.02679690  35 -1.786649  0.0827
## statusmaturing   0.0025782 0.04266169  35  0.060434  0.9522
##  Correlation:
##                (Intr) statsmtr
## statusmature   -0.472
## statusmaturing -0.296  0.140
##
## Standardized Within-Group Residuals:
##         Min          Q1         Med          Q3         Max
## -2.76201315 -0.71766223  0.03396789  0.67383510  2.20624889
##
## Number of Observations: 198
## Number of Groups: 38
```

```
AIC(mod) -AIC(mod.spatial)
```

```
## [1] -2.82759
## spatial auto-correlation has a larger AIC
```

4.2 Weighted interaction turnover

To calculate weighted interaction turnover we borrowed concepts from calculating phylogenetic β-diversity. We began by creating a ”phylogenetic” tree of interactions based on their position in the network using the getLinkCommunities function in the linkcomm package (Kalinka & Tomancak, 2011). For bipartite networks, the set of neighbors is used to count nodes for the edge similarity metric because node i and node j cannot share an edge in a bipartite network. The partition density for bipartite networks is calculated as:

$$(2/M) * \sum_{c} m_c * (m_c + 1 - n_c)/(2 * n_{c_0} * n_{c_1} - 2 * (n_c - 1)) \tag{1}$$

where M is the total number of edges, m_c is the number of edges in subset c, n_c is the number of nodes in subset c, n_{c_0} is the number of nodes in partition 0, and n_{c_1} is the number of nodes in partition 1 (from the getLinkCommunities helpfile).

With the resulting dendrogram and interaction dissimilarity matrices, we use the comdist function in the picante package (Kembel *et al.*, 2010) to calculate expected ”phylogenetic” distance separating two interactions drawn randomly from different communities. This gives us a measure of interaction turnover weighted by how interaction similarity. This is all wrapped up in the function CalcCommDis

```
phylo.int <- calcCommDis(spec, "Int", lc, abund.w=TRUE)
```

abund.w = TRUE weighted dissimilarity scores by species abundances.

We use a linear mixed model to regress weighted interaction turnover. When interactions were weighted by their similarity, both assembling and mature hedgerows had higher rates of turnover than field margins.

```
phylo.int.mod <- lmer(PhyloInt ~ SiteStatus +
                          (1|Site),
                      data=phylo.int$phylo.int)
print(summary(phylo.int.mod))
```

```
## Linear mixed model fit by REML t-tests use Satterthwaite approximations
##   to degrees of freedom [lmerMod]
## Formula: PhyloInt ~ SiteStatus + (1 | Site)
##    Data: phylo.int$phylo.int
##
## REML criterion at convergence: -1087.4
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -4.4271 -0.3718  0.1392  0.5481  2.9588
##
## Random effects:
##  Groups   Name        Variance Std.Dev.
##  Site     (Intercept) 0.003341 0.05781
##  Residual             0.004470 0.06686
## Number of obs: 459, groups:  Site, 38
##
## Fixed effects:
##                    Estimate Std. Error       df t value Pr(>|t|)
## (Intercept)        0.94117    0.02352 30.88000  40.014  < 2e-16 ***
## SiteStatuscontrol -0.11545    0.02738 31.12000  -4.216 0.000198 ***
## SiteStatusmature  -0.03339    0.03065 33.20000  -1.089 0.283853
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##             (Intr) StSttsc
## StSttscntrl -0.859
## SiteSttsmtr -0.767  0.659
```

5 Temporal changes in interaction patterns

5.1 Network structure

The baci.R script (before-after-control-impact, baci, the orginal design of the study, vs. the word for kisses in Italian) calculates various network metrics for the assembling hedgerows, standardizes them using null communities, then include then as response variables in linear models. The number of null communities can be changed by setting N at the top of the script.

CalcNetworkMetrics takes an argument ”index” which is passed to networkLevel within the package bipartite (Dormann *et al.*, 2008) if other metrics are desired. Unlike the turnover analysis, the networks include non-integers because the mean was taken over sampling rounds within a year. We therefore use the null model proposed by (Galeano *et al.*, 2009) to randomize the communities while fixing the total number of interactions, species and interaction frequency distributions (though any function that takes a single interaction matrix as an argument can be substituted using the null.fun argument of the calcNullStat function.

```
mets <- lapply(nets, calcNetworkMetrics, N)
cor.dats <- prepDat(mets,  spec)
```

Then regress the nestedness, modularity, specialization and connectance, plant/pollinator species richness by time (”ypr” in the data, years post restoration).

```
## ************************************************************
## linear mixed models
ys <- c("zNODF", "zmod.met.D", "zH2", "connectance")

formulas <-lapply(ys, function(x) {
    as.formula(paste(x, "~",
                     paste("scale(ypr)",
                           "(1|Site)",
                           "(1|Year)",
                           sep="+")))
})

formulas.spac <-lapply(ys, function(x) {
    as.formula(paste(x, "~",
                     paste("scale(ypr)",
                           sep="+")))
})

mods <- lapply(formulas, function(x){
    lmer(x,
         data=baci)
})

names(mods) <- ys
## results
lapply(mods, summary)
```

```
## $zNODF
## Linear mixed model fit by REML t-tests use Satterthwaite approximations
##   to degrees of freedom [lmerMod]
## Formula: zNODF ~ scale(ypr) + (1 | Site) + (1 | Year)
##    Data: baci
##
## REML criterion at convergence: 186.2
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -1.7060 -0.5585 -0.1869  0.4507  2.2506
##
## Random effects:
##  Groups   Name        Variance Std.Dev.
##  Year     (Intercept) 1.477    1.215
##  Site     (Intercept) 6.427    2.535
##  Residual             8.241    2.871
## Number of obs: 36, groups:  Year, 8; Site, 5
##
## Fixed effects:
##             Estimate Std. Error     df t value Pr(>|t|)
## (Intercept)   6.1644     1.3091 4.7280   4.709   0.0061 **
## scale(ypr)    1.2424     0.6717 6.2160   1.850   0.1121
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##            (Intr)
## scale(ypr) 0.008
##
## $zmod.met.D
## Linear mixed model fit by REML t-tests use Satterthwaite approximations
##   to degrees of freedom [lmerMod]
## Formula: zmod.met.D ~ scale(ypr) + (1 | Site) + (1 | Year)
##    Data: baci
##
## REML criterion at convergence: 116.8
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -1.8149 -0.7167 -0.2496  0.7042  2.8513
##
## Random effects:
##  Groups   Name        Variance Std.Dev.
##  Year     (Intercept) 0.000    0.000
##  Site     (Intercept) 0.000    0.000
##  Residual             1.471    1.213
## Number of obs: 36, groups:  Year, 8; Site, 5
##
## Fixed effects:
##             Estimate Std. Error      df t value Pr(>|t|)
## (Intercept)  -0.4738     0.2022 34.0000  -2.344   0.0251 *
## scale(ypr)    0.2363     0.2050 34.0000   1.152   0.2572
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##            (Intr)
## scale(ypr) 0.000
##
## $zH2
## Linear mixed model fit by REML t-tests use Satterthwaite approximations
##   to degrees of freedom [lmerMod]
## Formula: zH2 ~ scale(ypr) + (1 | Site) + (1 | Year)
##    Data: baci
##
## REML criterion at convergence: 189.7
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -2.0587 -0.4511 -0.1303  0.5563  2.0243
##
## Random effects:
##  Groups   Name        Variance Std.Dev.
##  Year     (Intercept) 3.509    1.873
##  Site     (Intercept) 3.461    1.860
##  Residual             9.107    3.018
## Number of obs: 36, groups:  Year, 8; Site, 5
##
## Fixed effects:
##             Estimate Std. Error      df t value Pr(>|t|)
## (Intercept)  -5.9515     1.1889  6.2850  -5.006  0.00213 **
## scale(ypr)   -1.4434     0.8571  6.8770  -1.684  0.13680
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##            (Intr)
## scale(ypr) 0.025
##
## $connectance
## Linear mixed model fit by REML t-tests use Satterthwaite approximations
##   to degrees of freedom [lmerMod]
## Formula: connectance ~ scale(ypr) + (1 | Site) + (1 | Year)
##    Data: baci
##
## REML criterion at convergence: -99.5
##
## Scaled residuals:
```

```
##      Min       1Q   Median       3Q      Max
## -1.62761 -0.54457 -0.04675  0.38328  2.08088
##
## Random effects:
##  Groups   Name        Variance  Std.Dev.
##  Year     (Intercept) 0.0000000 0.00000
##  Site     (Intercept) 0.0009775 0.03126
##  Residual             0.0021487 0.04635
## Number of obs: 36, groups:  Year, 8; Site, 5
##
## Fixed effects:
##               Estimate Std. Error        df t value Pr(>|t|)
## (Intercept)   0.210108   0.015979  3.937000   13.15 0.000213 ***
## scale(ypr)   -0.022972   0.008004 30.733000   -2.87 0.007359 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##            (Intr)
## scale(ypr) -0.003
```

As per the suggestion of reviewers, we also tested the same hypotheses with linear mixed models that included temporal auto-correlation, and splines. Like the turnover models, the temporal auto-correlation models showed similar results, but the fit was not substantially improved by adding the additional parameters. The spline chose to fit a line, indicating that the linear models were sufficient.

And the splines:

```
## ************************************************************
## splines

## for gaussian models
mods.splines <- lapply(ys, calcSpine)

## $info
## [1] 0
##
## $smoothingParameters
##    mu     v
## 10000 10000
##
## $info
## [1] 0
##
## $smoothingParameters
##    mu     v
## 11000 10000
##
## $info
## [1] 0
##
## $smoothingParameters
##    mu     v
## 10000 10000
##
```

```
## $info
## [1] 0
##
## $smoothingParameters
##    mu     v
## 21437  1750

names(mods.splines) <- ys

## for poisson models
mods.splines.log <- lapply(poi.ys, calcSpine, calc.log=TRUE)

## $info
## [1] 0
##
## $smoothingParameters
##    mu     v
## 11000 10000
##
## $info
## [1] 0
##
## $smoothingParameters
##    mu     v
## 11000 10000
##
## $info
## [1] 0
##
## $smoothingParameters
##    mu     v
## 11000 10000

names(mods.splines.log) <- poi.ys
```

The smoothing parameters indicate lines were fit.

5.2 Network robustness

The resilience.R script calculates resilience to species extinction using the method of (Memmott *et al.*, 2004). Species can be dropped by abundance (extinction.method = ”abund”) or degree (extinction.method =”degree”). From the helpfile: ”The procedure of this function is simple. For example imagine the web to represent a pollination web, in which pollinators die one by one. Set all entries of a column to zero, see how may rows are now also all-zero (i.e., species that are now not pollinated any more), and count these as secondary extinctions of the primary exterminated pollinator.” The area under the curve is then calculated.

This area is then regressed against time (”ypr”) Random effects of year and site are included.

```
## no effect of ypr on robustness
mod.ypr <- lmer(Robustness ~ ypr
                + (1|Site) + (1|Year),
                data=res[!is.na(res$ypr),])
summary(mod.ypr)

## Linear mixed model fit by REML t-tests use Satterthwaite approximations
##   to degrees of freedom [lmerMod]
## Formula: Robustness ~ ypr + (1 | Site) + (1 | Year)
##    Data: res[!is.na(res$ypr), ]
##
## REML criterion at convergence: -166.9
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -1.99009 -0.62116 -0.07289  0.56358  2.37929
##
## Random effects:
##  Groups   Name        Variance  Std.Dev.
##  Site     (Intercept) 2.395e-03 0.048943
##  Year     (Intercept) 5.091e-05 0.007135
##  Residual             4.842e-03 0.069583
## Number of obs: 80, groups:  Site, 14; Year, 8
##
## Fixed effects:
##              Estimate Std. Error        df t value Pr(>|t|)
## (Intercept)  0.333299   0.016943 10.950000  19.671 6.85e-10 ***
## ypr          0.000965   0.004133 32.430000   0.233    0.817
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##     (Intr)
## ypr -0.374
```

The next estimate of resilience we use is based on how quickly information spreads through a network as estimated by algebraic connectivity.

References

Anderson, M.J. (2006). Distance-based tests for homogeneity of multivariate dispersions. *Biometrics*, 62, 245–253.

Anderson, M.J., Crist, T.O., Chase, J.M., Vellend, M., Inouye, B.D., Freestone, A.L., Sanders, N.J., Cornell, H.V., Comita, L.S., Davies, K.F. *et al.* (2011). Navigating the multiple meanings of β diversity: a roadmap for the practicing ecologist. *Ecol. Lett.*, 14, 19–28.

Anderson, M.J., Ellingsen, K.E. & McArdle, B.H. (2006). Multivariate dispersion as a measure of beta diversity. *Ecology Letters*, 9, 683–693.

Chao, A., Chazdon, R.L., Colwell, R.K. & Shen, T.J. (2005). A new statistical approach for assessing similarity of species composition with incidence and abundance data. *Ecol. Lett.*, 8, 148–159.

Chase, J.M., Kraft, N.J., Smith, K.G., Vellend, M. & Inouye, B.D. (2011). Using null models to disentangle variation in community dissimilarity from variation in α-diversity. *Ecosphere*, 2, art24.

Dormann, C., Gruber, B. & Fründ, J. (2008). Introducing the bipartite package: analysing ecological networks. *R News*, 8, 8.

Galeano, J., Pastor, J.M. & Iriondo, J.M. (2009). Weighted-interaction nestedness estimator (wine): a new estimator to calculate over frequency matrices. *Environ. Model. Softw.*, 24, 1342–1346.

Kalinka, A.T. & Tomancak, P. (2011). linkcomm: an r package for the generation, visualization, and analysis of link communities in networks of arbitrary size and type. *Bioinformatics*, 27.

Kembel, S., Cowan, P., Helmus, M., Cornwell, W., Morlon, H., Ackerly, D., Blomberg, S. & Webb, C. (2010). Picante: R tools for integrating phylogenies and ecology. *Bioinformatics*, 26, 1463–1464.

Kuznetsova, A., Bruun Brockhoff, P. & Haubo Bojesen Christensen, R. (2014). *lmerTest: Tests for random and fixed effects for linear mixed effect models (lmer objects of lme4 package).* R package version 2.0-11.

Memmott, J., Waser, N.M. & Price, M.V. (2004). Tolerance of pollination networks to species extinctions. *Proc. R. Soc. Lond. B*, 271, 2605–2611.

Oksanen, J., Guillaume Blanchet, F., Kindt, R., Legendre, P., Minchin, P., O'Hara, R., Simpson, G., Solymos, P., Stevens, M. & Wagner, H. (2013). *vegan: Community Ecology Package*.

Ponisio, L.C., M'Gonigle, L.K. & Kremen, C. (2016). On-farm habitat restoration counters biotic homogenization in intensively managed agriculture. *Global Change Biol.*, 22, 704–715.

Winfree, R., Williams, N.M., Dushoff, J. & Kremen, C. (2014). Species abundance, not diet breadth, drives the persistence of the most linked pollinators as plant-pollinator networks disassemble. *The American Naturalist*, 183, 600–611.

Zuur, A.F., Ieno, E.N. & Elphick, C.S. (2010). A protocol for data exploration to avoid common statistical problems. *Methods in Ecology and Evolution*, 1, 3–14.