Evolution on Ecological Networks

M.K. Lau

March 6, 2014

Contents

1 6 Mar 2014

Finished running simulations and getting network stats (cscore, modularity, nestedness)

Question: incorporate centrality?

There are lots of different kinds of centrality measures. Let's assume that you are thinking about degree centrality for the moment, then a local measure of centrality is the number of edges that touch the node. Thus, in a mutualistic network, this centrality would tell you the number of directly interacting mutualist partners. A more global measure (considering indirect interactions, or pathways) would give you an idea of the size of the reachable species or community. Does this help? There are a couple of really good centrality review papers cited in my Throughflow centrality paper if you are interested. We can also discuss further.

- Look at Stuart's centrality paper
- Global centrality gives the size of the reachable species or genotypes in the community
- In a bipartite network, local centrality tells you how many interactions a species/genotype has
- Local degree centrality measures how many edges touch a node.
- Which centrality measure?

Looking at results:

2 24 Feb 2014

Writing modularity analysis script.

Look at cg$_s$*imulationsrepo*.

Started working on een$_m$*od*.*R*, *moveittocg*$_s$*imulationsrepo*.

Also, write and move the assocaited .sh file to cg$_s$*imulations*.

3 06 Feb 2014

Re-organized so that simulations are contained in cg$_s$*imulationsrepository*.

4 29 Jan 2014

SES exhibits complex patterns with respect to both nestedness and selection.

SES analyses:

1. Is ses correlated with selection?

   As slection increases, SES first decreases strongly, then begins to increase again.

2. Is ses correlated with nestedness?

   Communities can exhibit high clustering (i.e. low ses values), but not exhibit nestedness. This is possibly due to selection first clustering species, but then sorting species out, which increases nestedness.

```
> sim.nest <- read.csv('../results/een_exp_sym/nest_sim.csv')
> ses <- read.csv('../results/een_ses.csv')[,-1]
> ses[is.na(ses)] <- 0
> ses <- as.vector(t(ses))
> selection <- (0.00007924*2.511886^(sim.nest[,1]-1))
> plot(ses~selection)
> plot(sim.nest[,2]~ses)
> pairs(cbind(ses,nest=sim.nest[,2],selection))
>
```

5 25 Jan 2014

Debugging error in rmTrees for type.

```
"Error in sample.int(length(x), size, replace, prob) :
  too few positive probabilities"
```

6 28 Jan 2014

Continuing to work on figures.

7 27 Jan 2014

Making figures

```
> library(ComGenR)
> sim.s <- dget('../results/een_exp_sym/comsim_sym.rdata')
> sim.a <- dget('../results/een_exp_asym/comsim_asym.rdata')
> tree.gpm <- read.csv('../data/tree_gpm_een_sim.csv')
> tree.pheno <- read.csv('../data/trees_een_sim.csv')
> insects <- read.csv('../data/insects_een_sim.csv')
> sim.nest <- read.csv('../results/een_exp_sym/nest_sim.csv')
> selection <- (0.00007924*2.511886^((sim.nest[,1])-1))
> ###Plot of phenotypes
> plot(density(tree.pheno[tree.gpm[,1]==unique(tree.gpm[,1])[1],1]),ylim=c(0,0.25),
+      main='',xlim=c(5,27),col=1,xlab='Tree Phenotypic Value')
> for (i in 2:length(unique(tree.gpm[,1]))){
+   lines(density(tree.pheno[tree.gpm[,1]==unique(tree.gpm[,1])[i],1]),ylim=c(0,1),
+         main='',col=1)
+ }
> ###
>
> ###Nestedness by selection intensity
> plot(sim.nest[1:79,2]~selection[1:79],xlab='Selection',ylab='Nestedness Temperature
> nest.mu <- tapply(sim.nest[1:79,2],factor(selection[1:79]),mean)
> lines(spline(as.numeric(names(nest.mu)),nest.mu))
> ###Ordination of community
> ord <- nmds.min(nmds(vegdist(sim.a[[9]][[1]])))
> ch.plot(ord,tree.gpm[,1])
> ###Evenness
> ##Using pielous eveness = H / log(specnumber(com))
```

```
> even <- function(x){diversity(x)/log(specnumber(x))}
> even.sym <- lapply(sim.s,function(x) lapply(x,even));even.sym <- unlist(lapply(ev
> even.asym <- lapply(sim.a,function(x) lapply(x,even));even.asym <- unlist(lapply(
> mean(even.sym-even.asym)
> t.test((even.sym-even.asym))
>
```

Appendix 1. Comparing nmds and ipd.

```
> h2c.nmds <- read.csv('../results/een_exp_asym/h2c_shuster.csv')
> h2c.ipd <- read.csv('../results/een_exp_asym/h2c_perm.csv')
> h2c.nmds <- as.numeric(apply(h2c.nmds,2,function(x) x[2]))
> h2c.ipd <- unlist(matrix(h2c.ipd,ncol=3,byrow=TRUE)[,2])
> #selection <- as.numeric(gl(9,10))
> h2c.dat <- cbind(Selection=selection,h2c.nmds,h2c.ipd)
> h2c.dat <- h2c.dat[1:79,]
>                                          #
> par(mfrow=c(1,3))
> plot(h2c.nmds~Selection,data=h2c.dat,ylab='H2C NMDS')
> abline(lm(h2c.nmds~Selection,data=data.frame(h2c.dat)))
> plot(h2c.ipd~Selection,data=h2c.dat,ylab='H2C IPD')
> abline(lm(h2c.ipd~Selection,data=data.frame(h2c.dat)))
> plot(h2c.nmds~h2c.ipd,data=h2c.dat,xlab='H2C IPD',ylab='H2C NMDS')
> abline(lm(h2c.nmds~h2c.ipd,data=data.frame(h2c.dat)))
>
```

8 26 Jan 2014

Looking at een experiments from yesterday

```
> ex.sym.rm <- data.frame(read.csv('../results/een_exp_sym/rm_rnd.csv'),read.csv('../
> names(ex.sym.rm) <- c('sel','rnd','deg','typ')
>                                          #
> ex.asym.rm <- data.frame(read.csv('../results/een_exp_asym/rm_rnd.csv'),read.csv('.
> names(ex.asym.rm) <- c('sel','rnd','deg','typ')
> ###Remove selection level 9
> ex.sym.rm <- ex.sym.rm[-81:-90,]
> ex.asym.rm <- ex.asym.rm[-81:-90,]
>                                          #
> par(mfrow=c(1,2))
> pairs(data.frame(ex.sym.rm,ex.asym.rm))
>                                          #
> exp <- c(rep('sym',nrow(ex.sym.rm)),rep('asym',nrow(ex.asym.rm)))
> een.exp <- rbind(ex.sym.rm,ex.asym.rm)
> sel <- as.numeric(sapply(as.character(een.exp[,1]),function(x) strsplit(x,split='
> sel <- (0.00007924*2.511886^((as.numeric(sel))-1))
> een.exp <- data.frame(exp=exp,selection=sel,een.exp[,-1])
> colnames(een.exp)
>                                          #
> summary(lm(rnd~exp*selection,data=een.exp))
> summary(lm(deg~exp*selection,data=een.exp))
> summary(lm(typ~exp*selection,data=een.exp))
>                                          #plot each experiment together by the rem
```

```
> par(mfrow=c(1,3))
> plot(rnd~selection,data=een.exp,col=as.numeric(een.exp$exp))
> plot(deg~selection,data=een.exp,col=as.numeric(een.exp$exp))
> plot(typ~selection,data=een.exp,col=as.numeric(een.exp$exp))
>                                          #plot each removal type to compare
> par(mfrow=c(1,3))
> plot(rnd~deg,data=een.exp,col=as.numeric(een.exp$exp))
> plot(rnd~typ,data=een.exp,col=as.numeric(een.exp$exp))
> plot(deg~typ,data=een.exp,col=as.numeric(een.exp$exp))
>                                          #density plots
> ex.split <- split(een.exp,een.exp$exp)
> par(mfrow=c(1,3))
> plot(density(ex.split[[1]]$rnd),main='',col=1,ylim=c(0,20));lines(density(ex.split
> plot(density(ex.split[[1]]$deg),main='',col=1,ylim=c(0,20));lines(density(ex.split
> plot(density(ex.split[[1]]$typ),main='',col=1,ylim=c(0,20));lines(density(ex.split
>                                          #selection
> par(mfrow=c(1,2))
> plot(rnd~jitter(selection,factor=10),data=ex.split[[1]],col='red',ylab='Percent Noc
> points(deg~jitter(selection,factor=10),data=ex.split[[1]],col='green')
> points(typ~jitter(selection,factor=10),data=ex.split[[1]],col='blue')
> plot(rnd~jitter(selection,factor=10),data=ex.split[[2]],col='red',ylim=c(0.85,1),y
> points(deg~jitter(selection,factor=10),data=ex.split[[2]],col='green')
> points(typ~jitter(selection,factor=10),data=ex.split[[2]],col='blue')
>
```

Networks with assymetric species distributions are more susceptible to remvovals overall, but this susceptibility is driven by the species abundances and not the effect of genotype. When species abundances are even, then the effect of selection on a genetically based phenotype has a strong impact on the robustness of the community.

9 25 Jan 2014

Debugged een$_e xp_s ymandeen_e xp_a symaswellasrmTrees.Theexperimentscriptsbothneededtheremovale$

rmTrees type was selecting trees at random even though it was selecting a phenotype. Each step chose a random type and then a random tree, which made the algorithm equaivalent to selecting a tree at random.

10 24 Jan 2014

Final EEN analyses continued. Currently running on romer. Two scripts: een$_s ym.Randeen_a sym.R.Bot$ $15pm$.

11 23 Jan 2014

Working on the final EEN analyses:

1. Simulate networks with varying levels of genotypic effects and a percent genetic variance equivalent to published community heritability estimates
2. In an appendix, develop the use of permanova R2, inlcuding:
   (a) Calculation using Anderson's method

(b) Comparison to NMDS method in Shuster 2006

3. Measure nestedness and examine correlation between genetic variance and nestedness

4. Conduct Removal experiments (random vs targeted removal):

   (a) Random removal

   (b) Genotype removal based on phenotpic similarity

   (c) Centralized species removal

Coding in source file, een.R.

12 22 Jan 2014

Re-organized the flow for community simulation with cgSim.

Added two new functions to support cgSim: simTrees and simSpp.

Compared output (gamma and H2C) with Shuster 2006 output. Everything checks out with the exception of NMDS rotation error and slight numerical differences in H2C.

13 14 Jan 2014

Does genetic variation contribute to nestedness in ecological networks?

```
> library(ComGenR)
>   ##Types of trait distribution
>   ##uniform/random
```

```
> tree.trait <- runif(100,10,20)
> arth.trait <- rnorm(100,15,1)
> S <- tree.trait - arth.trait
> hist(S)
> plot(tree.trait,arth.trait)
>   ##increasing
>
>   ##decreasing
>   ##convex (bowl down)
> tree.trait <- rnorm(100,15,1)
> arth.trait <- rnorm(100,15,1)
> S <- tree.trait - arth.trait
> par(mfrow=c(1,2))
> hist(S)
> plot(tree.trait,arth.trait)
>   ##concave (bowl up)
>
> ###Run a set of simulations across a gradient of increasing genotype variation an
>
> ##Increase the range for a uniform distribution
> c.r <- 15.5 #center of the range
> ns.r <- 15 #number of steps over the range
> ng.r <- 10 #number of genotypes over the range
> nr.g <- 5 #number of reps for each genotype
> s.r <- 0.5 #step size over range
> steps <- seq(0,ns.r*s.r,by=s.r)
```

```
> tp.l <- list()
> for (i in 1:ns.r){
+   tp.l[[i]] <- gpmTrees(seq((c.r-steps[[i]]),(c.r+steps[[i]]),length=ng.r),nr.g)
+ }
> cp.l <- gpmCom(n=20)
> com.sim <- lapply(tp.l,function(x) cgSim(tree.pheno=x,insect=cp.l,YY=1,GG=8,reps=
> com.sim. <- list()
> for (i in 1:length(com.sim)){
+   com.sim.[[i]] <- round(com.sim[[i]][[1]][[1]][[8]],0)
+ }
> com.nest <- unlist(lapply(com.sim.,function(x) nestedtemp(x)$statistic))
> V.g <- unlist(lapply(tp.l,function(x) var(x[,2])))
>                                          #Correlation between H2Cnms~Vg and H2Cper
> cs.nms1d <- list()
> cs.stress <- list()
> for (i in 1:length(com.sim.)){
+   cs.nms1d[[i]] <- nmds(vegdist(com.sim.[[i]]),1,1)
+   cs.stress[[i]] <- min(cs.nms1d[[i]]$stress)
+   cs.nms1d[[i]] <- nmds.min(cs.nms1d[[i]])[,1]
+ }
> h2c.nms <- lapply(cs.nms1d,getH2C,g=as.character(tp.l[[1]][,1]))
> h2c.nms <- do.call(rbind,h2c.nms)[,2]
> h2c.adonis <- pblapply(com.sim.,function(x,g) adonis(x~g),g=as.character(tp.l[[1]
> h2c.perm <- unlist(lapply(h2c.adonis,function(x) x$aov.tab[1,5]))
> summary(lm(h2c.nms~V.g))
> summary(lm(h2c.perm~V.g))
```

```
> plot(h2c.nms~V.g,pch=19,xlab='Vg',ylab='H2C',col='grey',ylim=c(0,0.85),font.lab=2)
> points(h2c.perm~V.g,pch=19,col='black')
> abline(lm(h2c.nms~V.g),col='grey',lty=2)
> abline(lm(h2c.perm~V.g),col='black')
> legend('topleft',legend=c('NMDS','PerMANOVA'),col=c('grey','black'),pch=19)
> h2c.ancova <- data.frame(vg=c(V.g,V.g),
+                          type=c(rep('nms',length(h2c.nms)),rep('perm',length(h2c.pe
+                          h2c=c(h2c.nms,h2c.perm))
> summary(lm(h2c~vg+type,data=h2c.ancova))
>                                          #
> par(mfrow=c(1,2))
> plot(density(tp.l[[1]][,2]),main='',xlab='Tree Phenotype',xlim=c(5,27),ylim=c(0,1),
> for (i in 2:length(tp.l)){lines(density(tp.l[[i]][,2]),col=heat.colors(length(tp.
> plot(com.nest~I(V.g/max(V.g)),pch=1,col='grey',cex=1.5)
> points(com.nest~I(V.g/max(V.g)),pch=19,col=heat.colors(length(tp.l))[1:length(tp.
> abline(lm(com.nest~I(V.g/max(V.g))))
>                                          #
> par(mfrow=c(1,2))
> plotweb(com.sim.[[1]][order(apply(com.sim.[[1]],1,function(x) sum(sign(x))),decre
+                       order(apply(com.sim.[[1]],2,function(x) sum(sign(x))),decre
> plotweb(com.sim.[[15]],method='normal')
> cgPlotweb <- function(x,g){
+   x <- x[order(apply(x,1,function(x) sum(sign(x))),decreasing=TRUE),
+            order(apply(x,2,function(x) sum(sign(x))),decreasing=TRUE)]
+   plotweb(apply(x,2,function(x,g) tapply(x,g,mean),g=g),method='normal')
+ }
```

```
> cgImage <- function(x,xlab='',ylab='',col=c(0,1)){
+   x <- x[order(apply(x,1,function(x) sum(sign(x))),decreasing=TRUE),
+          order(apply(x,2,function(x) sum(sign(x))),decreasing=TRUE)]
+   x[x!=0] <- 1
+   image(t(x),col=col,xaxt='n',yaxt='n')
+   mtext(side=c(1,2),c(xlab,ylab),font=2)
+ }
> par(mfrow=c(1,2))
> cgPlotweb(com.sim.[[1]],tp.l[[1]][,1])
> cgPlotweb(com.sim.[[15]],tp.l[[15]][,1])
> cgImage(com.sim.[[1]],xlab='Tree Genotype',ylab='Insect Species',col=c(0,'grey'))
> cgImage(com.sim.[[15]],xlab='Tree Genotype',ylab='Insect Species',col=c(0,'grey'))
>
```

14 9 Jan 2014

Developing the manuscript, package and vignette for lab meeting.

Package

1. Debug current set of functions
2. Populate help files
3. Check –as-cran
4. Build as v1.0
5. Post to github

6. Post to cran

Vignette

1. Write body text
2. Set figures
3. Proof

Manuscript

1. Two articles 1) Methods in Ecology and Evolution = Vignette and 2) Ecography = Modeling and Data
2. Summarize results
3. Write outline

15 7 Jan 2014

```
> ###Moved to src/een_sim.R
>
```

16 20 Dec 2013

Add data from Gina, Art, Rikke and Cam, Adrian?

The idea being to simulate but then observe patterns in real data.

17 18 Dec 2013

Sent outline (see Docs) to Steve. Check back with him in a couple of days.

18 17 Dec 2013

1. Conducting network modeling

19 16 Dec 2013

Results

- Lonsdorf model produces nested biparite genotype-species networks at high levels of genetic effects
- Lonsdorf model also produces co-occurrence network structure at high levels of genetic effects

Look at Jelle's models of stability.

Check Lonsdorf's model output with your R cod output.

Analyze the simulation output from two perspectives: 1) bipartaite network structure of tree species and 2) co-occurrence patterns in dependent communities.

From the bipartite perspective, if we assume nestedness, how will networks evolve.

From the co-occurrence perspective, what network patterns arise given different amounts of influence of genetics versus environment. The interaction context comes in the form of co-occurrences influencing encounter rates, which are the rate limiting step for interactions.

Also from both of these perspectives, keep in mind the possibility of actually looking at interactions among species.

Notes on Lonsdorf Code

- Line 30 "art$_r$" $doesn't match any other variables in the code, most likely a typo and shoul g be art_g$

20 11 Dec 2013

Strategy for Chap3

- - seeNets package on CRAN
  - seeNets pub MEE
  * Make sure to email Araujo et al.
  - add simulator, which will show the development of networks along a genetic gradient
  * This can be based on the Shuster 2006 paper so that you don't scoop the Lonsdorf manuscript

21 6 Dec 2013

What does the co.net tell us? Significant co-occurrences in units of standard deviations by default.

What does the dep.net tell us? Significant conditional probability of co-occurrence of one species given another. Conditional probability is the co-occurrence probability divided by the occurrence probability of one species.

Apply network modeling to Lonsdorf output:

```
> source('../src/lonsdorf.R')
> source('../../lichen_coo/src/seenetR.R')
> library(sna)
> x <- lapply(out[[1]][[4]],round,digits=0)
> for (i in 1:length(x)){
+   x[[i]][x[[i]]<10] <- 0
```

```
+ }
> con <- lapply(x,co.net)
> par(mfrow=c(1,2))
> coord <- gplot(abs(con[[1]]))
> gplot(abs(con[[8]]),coord=coord)
>
>
```

22 26 Nov 2013

Need to review species interaction network literature, focusing on simulations:

- Baiser's metabolic network
- Allesina's networks
- Sala's networks

23 Before 26 Nov 2013

Outline

- Main Question: how does the structure of ecological networks influence the evolution of species in the community?
- Simulate the evolution of traits determining interactions (and possibly linked traits of varying association)
- Review:

  - Salas manuscript
    - Different from Sala's in that it will explore more network structure, including modularity
  - Lonsdorf Manuscript
    - Different from Lonsdorf's in that the emphasis is exploring how network structure influences evolution
  - Shuster et al. (2006)
  - Nuismer and Doebeli (2004)
  - Nowak 2006 Chapter 8 (population genetics on graphs)

- Model:
  - Individual Based
  - Genetics -> binary vector
  - Phenotype -> vector of continuous trait values ($\sim N(\mu,\sigma)$)
  - Interaction Network -> adjacency matrix + trait matching
  - Community Dynamics -> demographics (birth - death) and stability
  - Ecosystem Dynamics -> productivity and energy flow
  - Evolution -> selection (fitness + reproduction) and recombination + mutation

- Pseudocode:

1. Generate interaction network
    - adjacency matrix

– make sure to control for covarying structural properties, such as connectence

2. Generate genome (i.e. create a binary vector)
3. Map genotype to phenotype (i.e. for each allele draw a value (0 <= value < infinity) from a normal distribution)
   – Repeat for n individuals in k species
4.

- Explore:
  - Stability - resistance and resilience
  - Additive versus non-additive (epistatic) genetic effects
  - Random, foundation, nested/modular
  - *Spatial (e.g. relative abundance effects)*
  - Others...?

24 Check out simecol

`http://cran.r-project.org/web/packages/simecol/vignettes/a-simecol-introduction.pdf`

```
> library(simecol)
> ?simecol
> ?indbasedModel
>
```

25 The Model

25.1 Generate Interaction Network

```
>                                          #generate a square, symmetric adjacency m
> rnet <- function(n){
+ x <- array(sample(c(0,1),(n^2),replace=TRUE),dim=c(n,n))
+ x[lower.tri(x)] <- t(x)[lower.tri(x)]
+ return(x)
+ }
>                                          #
>
```

25.2 Generate Genome

```
>                                          #generate a genome (binary) of length n
> rgenome <- function(n){
+   return(sample(c(0,1),n,replace=TRUE))
+ }
>                                          #generate a population of genomes of leng
> genomePop <- function(n){
+ g <- sapply(rep(n,10),function(x) rgenome(x))
+ g <- split(g, rep(1:ncol(g), each = nrow(g)))
+ return(g)
+ }
>                                          #generate a community of genomes
>                                          #generate a community
```

```
>                                          #nk is a list of population sizes nk <- s
> genomeCom <- function(nk){
+   return(lapply(nk,function(x) genomePop(x)))
+ }
>
>
```

25.3 Map Genotype to Phenotype

```
>                                          #map a genotype to a phenotype
> gpMap <- function(g,mu,sigma){
+   x <- cbind(g,mu,sigma)
+   y <- apply(x,1,function(x) if (x[1] == 0){0}else{rnorm(1,x[2],x[3])})
+   return(y)
+ }
>
```

25.4 Generate a population for each species

```
>                                          #generate a population
>                                          #g is a list of genotypes
>                                          #mu is a vector of population trait means
>                                          #sigma is a vector of population trait st
> genPop <- function(g,mu,sigma){
+   return(lapply(g,function(x,m,s) gpMap(x,m,s),m=mu,s=sigma))
+ }
>
```

```
>
```

26 Test Run

```
>                                          #generate a population of genotypes for e
>                                          #
> nk <- as.matrix(rep(15,25)) #create a kX1 matrix of genome lengths
> nk <- split(nk, rep(1:nrow(nk), each = ncol(nk))) #create a genome length list fo
> meta.genome <- genomeCom(rep(15,15)) #create a population of genomes for each spe
>                                          #1. initialize a population of genotypes
>                                          #2. map genotypes to phenotypes using tra
>                                          #3. calculate trait matching using adjace
>                                          #4. calculate fitness using trait matchin
>                                          #5. select a proportion for reproduction
>                                          #6. recombine and mutate
>                                          #7. select a proportion for death
> gn <- 15 #genome length
> mu <- runif(n,0,100)
> sigma <- rnorm(n,15,3)
> gpMap(rgenome(15),mu,sigma)
>
>
```