An introduction to *pez*

William D. Pearse (wdpearse@umn.edu)

September 17, 2014

Contents

1 Preamble

You can install *pez* by typing `install.packages("pez")`, and get a listing of the functions in the package by typing `library(help=pez)`. If you find any bugs, or have any feature requests for the package, please use the online tracker at. Indeed, please contribute to the package using at its GitHub site—help is always welcome! If you just can't wait to get the latest version, you can install the latest version directly from *GitHub* (`require(devtools);install_github('willpearse/pez')`).

While ***pez*** contains much novel code, it relies heavily on the *R* ecosystem. Much of the community phylogenetic metric functions are wrappers around existing code (detailed in the help files for each function); notably *caper* (**?**) and *picante* (**?**) but many others. Please cite the authors of these packages so that their hard-work is rewarded!

2 Data formats in *pez*

pez functions work with *comparative community ecology* objects, called `comparative.comm` objects. These are designed to help keep phylogenies, community data matrices, species trait data, and environmental data all in the same place in a format that makes it easy to work with them. They're much less scary than they sound!

Below we load *pez*, some example data that comes with it, and then make a `comparative.comm` object. You can examine the phylogeny (`tree`), community data (`comm`), and trait data (`data`) that went into making dataset for yourself, although

all the data types are explained in more detail below. Below we use the `?` dataset to show *pez*'s features.

```
library(pez)
data(laja)
data <- comparative.comm(invert.tree, river.sites,
    invert.traits)
```

pez is conservative; if you give it trait data for only half of the species in your community data, the `comparative.comm` object will only contain data on those species that have both trait data and community data. The same goes for the phylogeny, and for sites with environmental data. *pez* will warn you about the loss of species or traits when you print the object to screen, and while it's making the comparative.comm object (unless you set the argument `warm=FALSE`).

You can also subset your comparative.comm object to exclude certain species or sites, in much the same way you can a `data.frame`. Note that *pez* will not (by default) warn you if this operation drops out certain species or sites. For example:

```
site.subset <- data[1:5, ]
spp.subset <- data[, 1:3]
spp.subset.warnings <- data[, 1:3, warn = TRUE]

## Warning:  Mismatch between phylogeny and other data, dropping 57 tips
## Warning:  Mismatch between community matrix and other data, dropping
57 columns
## Warning:  Mismatch between traits and other data, dropping 57 columns
```

2.1 Phylogenies

pez uses the `phylo` format in the `ape` package to store phylogenies. You can load your own phylogenies using the ape functions `read.tree` and `read.nexus`.

2.2 Community data

pez uses the same community data format as the `vegan` package: a `matrix` with sites in the rows and species in the columns. The elements of the community matrix can be species abundances or presence/absence (1/0). Not all the species in your matrix have to be present in a site—there can be empty columns in your data. This

is particularly important when using the dispersion measures (see below). Your data should be named, with row names that correspond to sites, and column names that correspond to species.

2.3 Trait data

Trait data should be a `data.frame` with row names that correspond to the species in the phylogeny, and named columns for each separate trait.

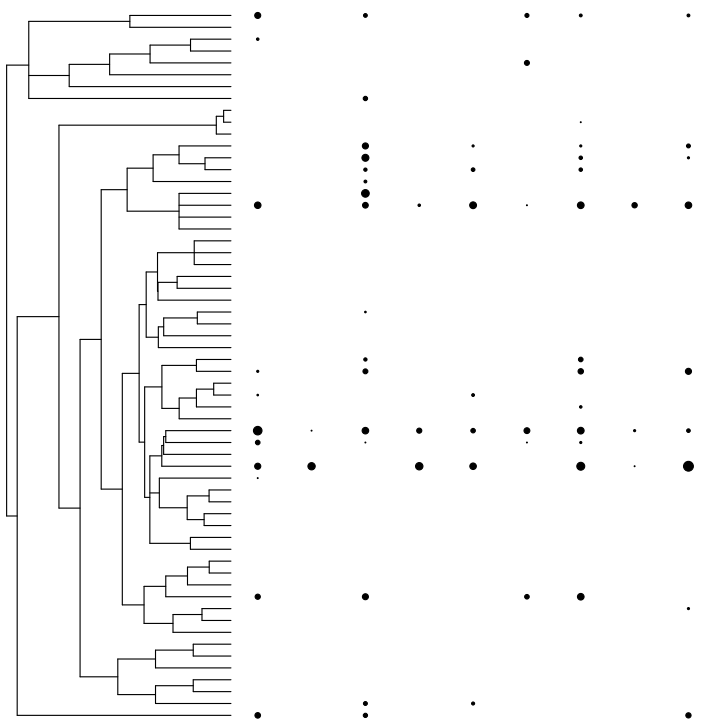
2.4 Environemntal data

Environmental data should be a `data.frame` with row names that correpsond to the sites in your community data, and separate (named) columns for each kind of environmental data.

3 Plotting and exploring data

pez comes with a few functions that are intended to make exploring your data slightly easier. For instance, you can plot out graphs of species abundances in communities (note the use of a function in the `dot.cex` argument).

```
plot(data, abundance = TRUE, dot.cex = function(x) ifelse(x >
    0, 2 * log10(x), 0))
```



4 Community phylogenetic metrics

pez splits community phylogenetic metrics into four functions according to the scheme outlined by `?`: `shape`, `evenness`, `dispersion`, and `dissimilarity`. Shape metrics measure the structure of an community phylogeny, while evenness metrics additionally incorporate species abundances. Dispersion metrics examine whether phylogenetic biodiversity in an assemblage differs from the expectation of random assembly from a given set of species. Finally, dissimilarity measures the pairwise difference in phylogenetic biodiversity between assemblages.

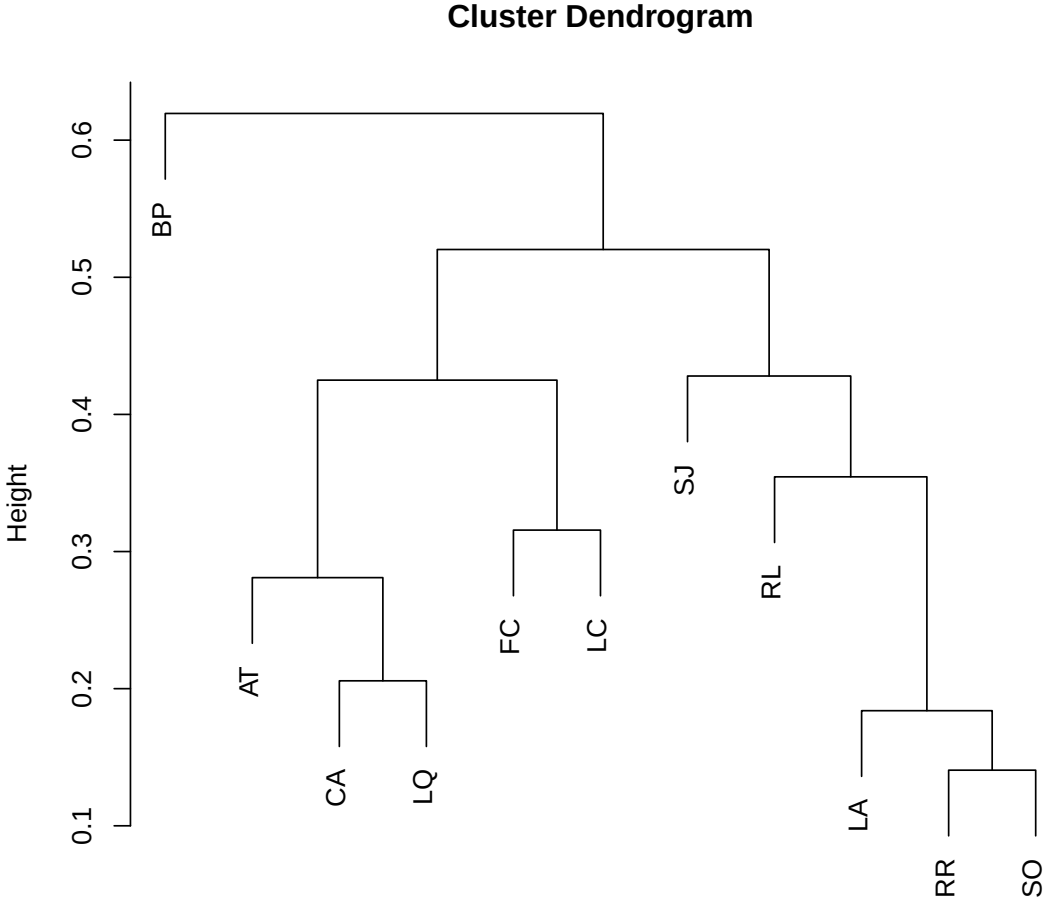
You can calculate all metrics within a class at the same time (which is what we recommend), or you can pick a particular one. Below we show how to calculate the metrics, and give examples of how to work with their output.

```
shape.output <- shape(data)
# Get all the coefficients
# coef(shape.output) ...or...
# shape.output$coef ...or...
```

```
shape.output$mpd

##  [1]  985.3 1161.6  924.8  986.4  976.4
##  [6] 1117.3  950.1  813.8  924.6  894.0
## [11]  978.9

dissimilarity.output <- dissimilarity(data,
    metric = "phylosor")
plot(hclust(dissimilarity.output$phylosor))
```



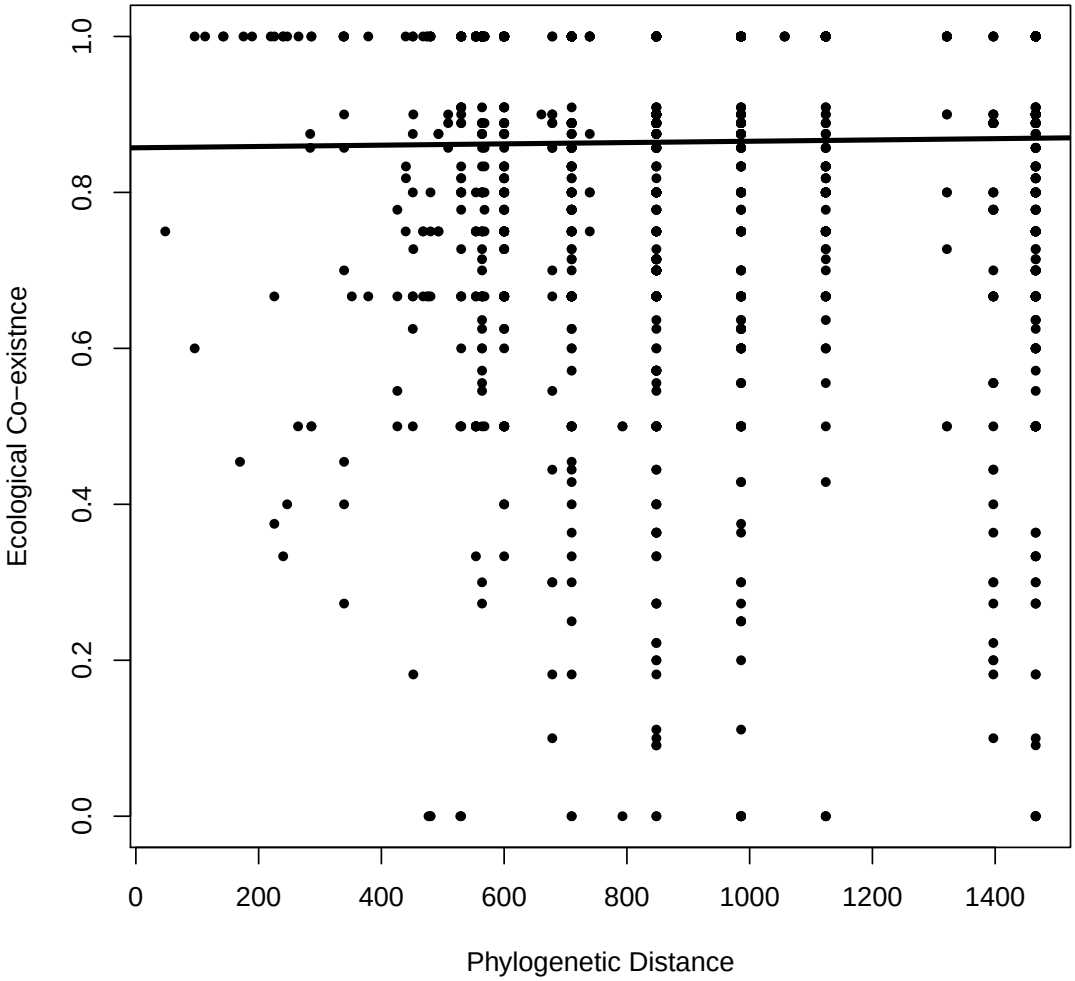
Note that *phylosor* (**?**) is reported as a dissimilarity in *pez*: it's not the fraction of shared branch lengths, but 1- the fraction of shared branch length. Remember: the function is called `dissimilarity`!

5 Eco-evolutionary regression (*pez*)

pez is intended to replace and improve upon earlier Visual Basic/C programs called *EcoPhyl*. You can regress the relative coexistence of species in your dataset against

those species phylogenetic (`eco.phy.regression`) and trait (`eco.trait.regression`) dissimilarity, as well as shared habitat preferences based on environmental tolerances (`eco.env.regression`).

```
model <- eco.trait.regression(data, method = "lm")
more.complex.model <- eco.trait.regression(data,
    method = "mantel", altogether = FALSE,
    permute = 10)
plot(eco.phy.regression(data, method = "quantile"),
    pch = 20)
```



However, the real power in this approach comes from combining information about the phylogenetic signal of species traits with the output from an `eco.trait.regression` of those traits. This amounts to plotting something about the evolution of those traits against their ecology. This sort of approach was first proposed by **?** (figure 4), and you can use the `fingerprint.regression` function to carry it out. Originally, **?** proposed the use of Mantel regressions (`eco.phy.regression`), but in *pez* you can use measures of phylogenetic signal (`phy.signal`). There's no need to do all these steps separately, though:

```
model <- fingerprint.regression(data, eco.permute = 10)
```

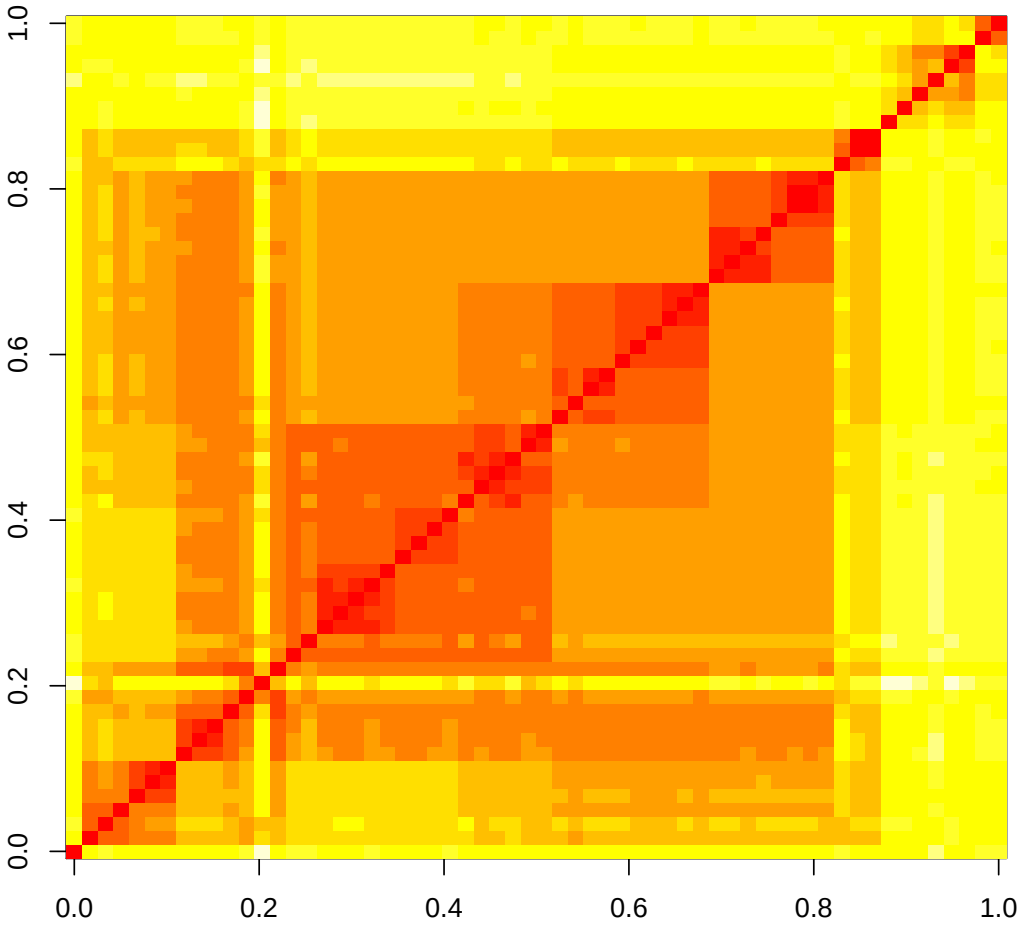
6 Traitgrams

If FD_{ij} and PD_{ij} are functional and phylogenetic distances between species i and j, the functional-phylogenetic distance between i and j is,

$$((1-a)FD_{ij}^{p} + aPD_{ij}^{p})^{1/p} \tag{1}$$

This distance matrix can be computed using the `funct.phylo.dist` function. For example, using the `data` object, $a = 0.5$, and $p = 2$, we have,

```
fpd.data <- funct.phylo.dist(data, phyloWeight = 0.5, p = 2)
image(as.matrix(fpd.data))
```



We can use these distance matrices in community randomization tests,

```
library(picante)
ses.mpd(data$comm, funct.phylo.dist(data, phyloWeight = 0.5, p = 2))

##    ntaxa mpd.obs mpd.rand.mean
## AT    29  0.5361        0.5372
## BP     8  0.6336        0.5341
## CA    24  0.4986        0.5346
```

```
## FC    18  0.5591        0.5367
## LA    12  0.5154        0.5362
## LC    21  0.5920        0.5368
## LQ    28  0.5067        0.5353
## RL    18  0.4491        0.5356
## RR    16  0.5268        0.5353
## SJ    20  0.4836        0.5339
## SO    15  0.5242        0.5339
##    mpd.rand.sd mpd.obs.rank mpd.obs.z
## AT     0.02693          472  -0.04415
## BP     0.06995          917   1.42244
## CA     0.03252          149  -1.10516
## FC     0.04069          701   0.54963
## LA     0.05258          330  -0.39646
## LC     0.03520          948   1.56800
## LQ     0.02830          168  -1.01179
## RL     0.03933           27  -2.19944
## RR     0.04502          413  -0.18983
## SJ     0.03726           98  -1.35074
## SO     0.04454          407  -0.21747
##    mpd.obs.p runs
## AT     0.472  999
## BP     0.917  999
## CA     0.149  999
## FC     0.701  999
## LA     0.330  999
## LC     0.948  999
## LQ     0.168  999
## RL     0.027  999
## RR     0.413  999
## SJ     0.098  999
## SO     0.407  999
```

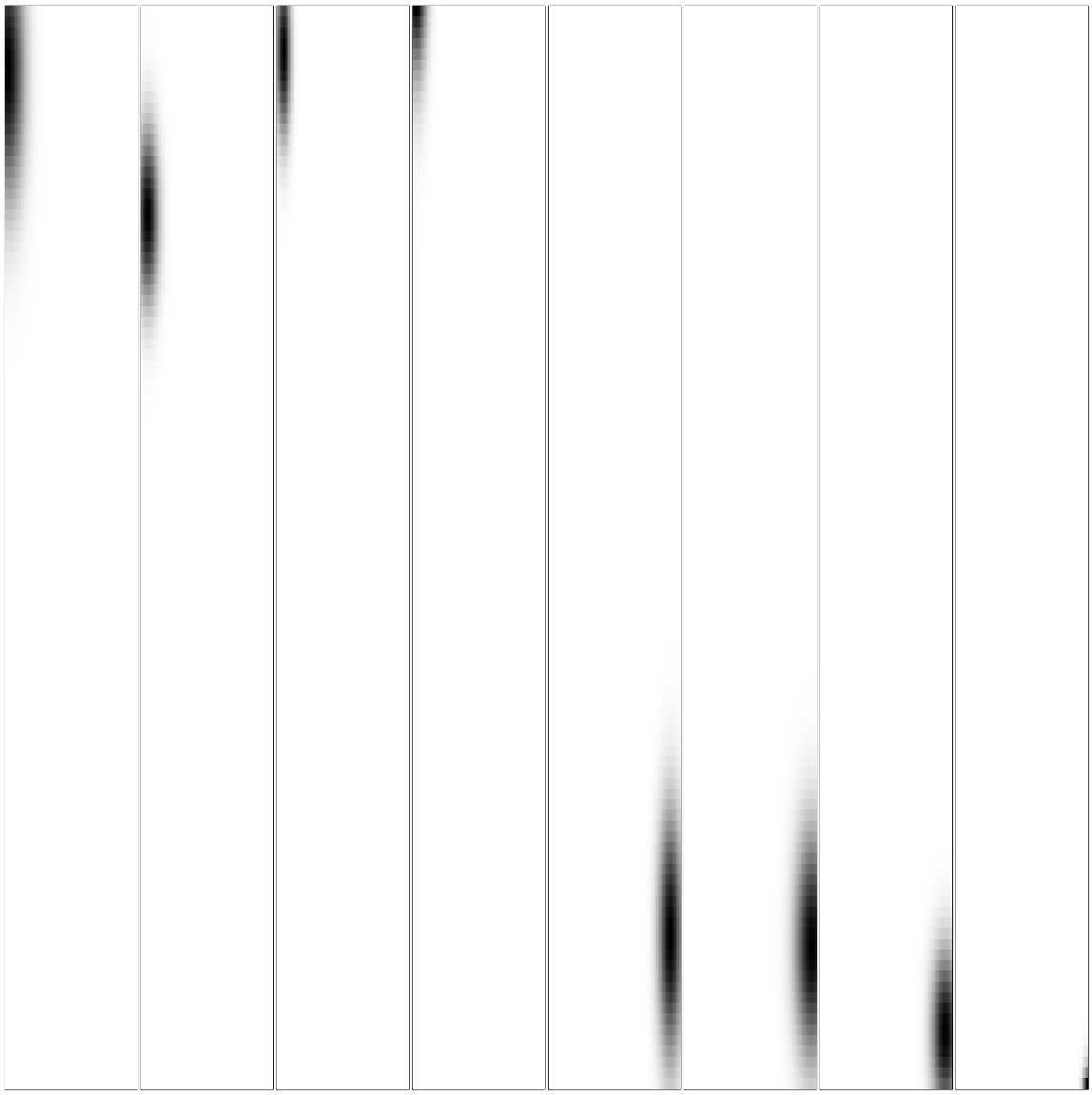
7 Simulation

A good simulation is one that does exactly what you want it to do, and *pez* provides a number of simulation functions that may be useful to you as (1) tools, or (2) starting

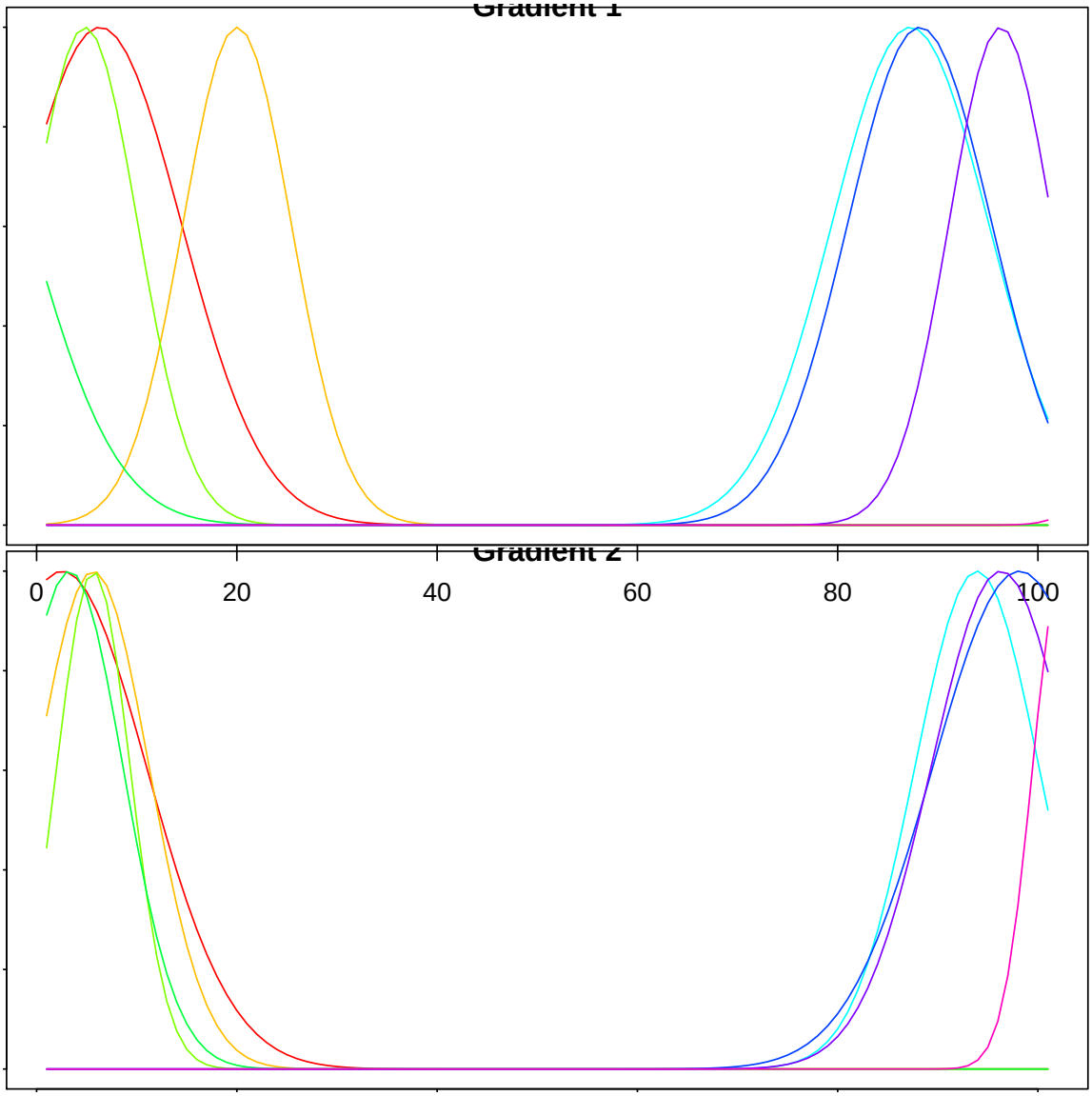
points for your own simulations.

`scape` allows you to repeat the analysis of **?**, simulating the assembly of species across a landscape given phylogenetically structured assembly. The parameters are complex, but they can generate some useful expected distributions, and give you a feel for regional assembly. You'd do well to read the paper that generated these models, but here's an example of their use:

```
library(ape)
library(plotrix)
tree <- stree(8, type = "balanced")
kk <- scape(tree, scape.size = 100, g.center = 100,
    g.range = 1, g.repulse = 1, wd.all = 150,
    signal.center = TRUE, signal.range = FALSE,
    same.range = FALSE, repulse = FALSE,
    center.scale = 1, range.scale = 1, repulse.scale = 1,
    site.stoch.scale = 0, sd.center = 3,
    sd.range = 1, rho = NULL, th = 20)
par(mfrow = c(1, Ntip(tree)), mar = c(0.1,
    0.1, 0.1, 0.1))
for (j in 1:Ntip(tree)) color2D.matplot(1 -
    kk$sppXs[, , j]/max(kk$sppXs[, , j]),
    xlab = "", ylab = "", main = "", border = NA,
    do.hex = FALSE, axes = FALSE)
```



```
par(mfrow = c(2, 1))
matplot((kk$X1), type = "l", xlab = "gradient",
    ylab = "probability", main = "Gradient 1",
    col = rainbow(dim(kk$X1)[2]), lty = 1)
matplot((kk$X2), type = "l", xlab = "gradient",
    ylab = "probability", main = "Gradient 2",
    col = rainbow(dim(kk$X2)[2]), lty = 1)
```



Alternatively, you can model the evolution of species and, at the same time, their assembly through a community. The only problem here is that the models are much simpler, but hopefully they are tunable to your liking! Explore the `sim.meta.comm` and `sim.meta.phy.comm` functions to find out more.

References

Bryant, J.A., Lamanna, C., Morlon, H., Kerkhoff, A.J., Enquist, B.J. & Green, J.L. (2008) Microbes on mountainsides: contrasting elevational patterns of bacterial and plant diversity. *Proceedings of the National Academy of Sciences* **105**, 11505–11511, URL http://www.pnas.org/content/105/suppl.1/11505.abstract.

Cavender-Bares, J., Ackerly, D.D., Baum, D.a. & Bazzaz, F.a. (2004) Phylogenetic overdispersion in Floridian oak communities. *The American Naturalist* **163**, 823–43.

Helmus, M.R. & Ives, A.R. (2012) Phylogenetic diversity-area curves. *Ecology* **93**, S31–S43.

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

Orme, D., Freckleton, R., Thomas, G., Petzoldt, T., Fritz, S., Isaac, N. & Pearse, W.D. (2013) *caper: comparative analyses of phylogenetics and evolution in R*. URL http://CRAN.R-project.org/package=caper, r package version 0.5.2.

Pearse, W.D., Cavender-Bares, J., Puvis, A. & Helmus, M.R. (2014) Metrics and models of community phylogenetics. *Modern Phylogenetic Comparative Methods and their Application in Evolutionary Biology—Concepts and Practice* (ed. L.Z. Garamszegi), Springer-Verlag, Berlin, Heidelberg.