

An introduction to *pez*

William D. Pearce (wdpearse@umn.edu)

October 28, 2014

Contents

1	Preamble	1
2	Data formats in <i>pez</i>	2
3	Community phylogenetic metrics	4
4	Eco-evolutionary regression	6
5	Functional phylogenetic distances and traitgrams	8
6	Simulation	9

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 [https://github.com/willpearse/pez/issues](#). 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* (Orme *et al.*, 2013) and *picante* (Kembel *et al.*, 2010) but many others as well. Please cite the authors of these packages in your publications so that their hard-work is rewarded!

2 Data formats in *pez*

pez functions work with `comparative.comm` objects (*comparative community ecology*). 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 Helmus & Ives (2012) dataset to show *pez*'s features.

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

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`. Dropping a site that contained the only instance of a species will remove it from dataset, and *pez* will not warn you about this unless you specify `[, , warn=TRUE]`. For example:

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

## Warning: Mismatch between community matrix and other data, dropping
0 columns

## Comparative community dataset of 60 taxa:
## Phylogeny:
## 54 internal nodesCommunity data:
## 10 sites, 60 taxa
```

```
## Trait data:
##      2  variables
## Environmental data:
##      10  sites,  6  variables
```

pez makes it easier to work with and manipulate datasets. The functions **species** and **sites** are safe ways of manipulating all the parts of your data at the same time. For example:

```
species(data)[1:2]

## [1] "Acari"          "Erpetogomphus"

species(data)[1:2] <- c("new", "names")
sites(data)[1:2] <- c("newer", "names")
data <- data[, colSums(data$comm) > 5]
```

The final example above showed you can work with the internal components of a **comparative.comm** to get things done quicker, in this case removing all species that were only recorded five times in the dataset. The help entry for **cc.manip** contains more examples, and **plot.comparative.comm** is a quick plotting tool. While we have provided a number of wrapper functions to make life easier, it's easy (and useful) to interact with the underlying community ecology (**\$comm**), phylogeny (**\$phy**), traits (**\$data**), and environmental (**\$env**) data. Adding extra trait data or environmental variables on the fly is no problem, but be aware that *pez* cannot check for missing data unless you call **comparative.comm** on something.

3 Community phylogenetic metrics

pez splits community phylogenetic metrics into four functions according to the scheme outlined by Pearse *et al.* (2014): **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. The intention is to make it easy to work with different community phylogenetic metrics, since each captures a different part of the structure of your data. Working with **shape**, **evenness**, and **dispersion** metrics is exactly the same, so below we only show **shape**.

```
shape.output <- shape(data)
dim(coef(shape.output))

## [1] 11 15

coef(shape.output)[1:3, 1:3]

##           psv      psr      mpd
## CA 0.6618 11.912 970.2
## FC 0.7202 7.202 1055.7
## LA 0.6660 7.993 976.4
```

Both **shape** and **evenness** metrics, by default, only calculate the **all-quick** metrics; specifying **metric='all'** will calculate slower metrics such as Pagel's λ . These can take a *very* long time to calculate for large datasets! You can also calculate these metrics using functional traits, a square-rooted phylogeny (following Letten & Cornwell, 2014), or any kind of distance matrix you can put together. The argument **traitgram** can be used to set a distance matrix that mixes explanatory power from phylogeny and traits, following Cadotte *et al.* (2013) (see below), and you can compare the output of different traitgram values. Not all metrics can meaningfully be calculated using external distance matrices, traitgrams, or square-rooted phylogenies, however.

```

sqrt <- shape(data, sqrt.phy = TRUE)

## Warning: Phylogeny is not ultrametric; see function details

traits <- shape(data, traitgram = 1) #traits alone
traits <- shape(data, traitgram = c(0, 0.5)) #phylogeny and both
traits <- shape(data, ext.dist = as.dist(cophenetic(data$phy)))

```

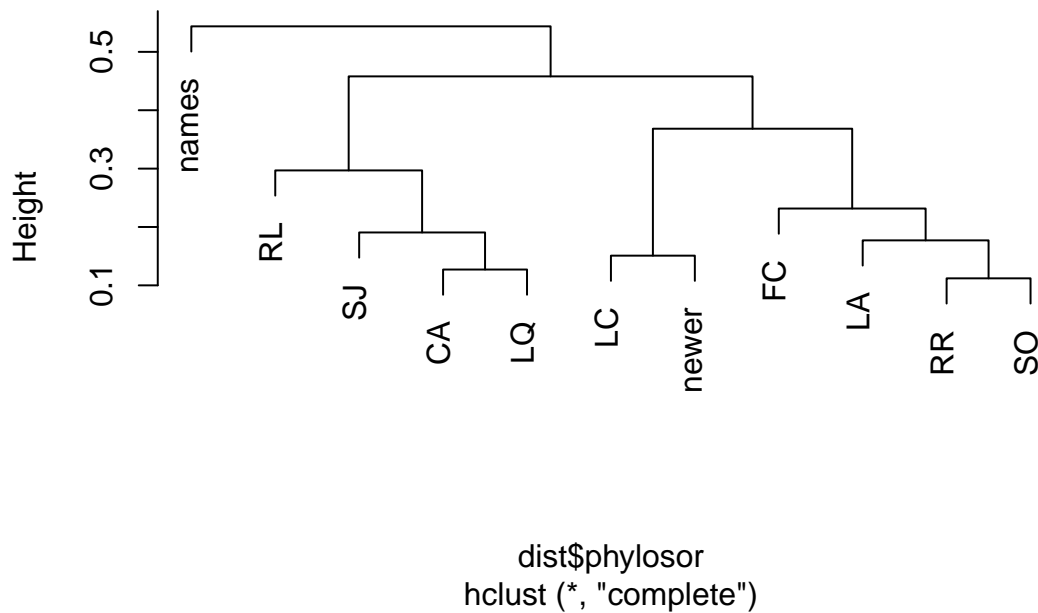
`dissimilarity` works slightly differently, because it returns a list of distance matrices that describe your community data. *phylosor* (Bryant *et al.*, 2008) is reported as a dissimilarity in *pez*: it's not the fraction of shared branch lengths, but 1- the fraction of shared branch length. This is not how it is in other packages, but remember: the function is called `dissimilarity`!

```

dist <- dissimilarity(data, "phylosor")
plot(hclust(dist$phylosor))

```

Cluster Dendrogram



4 Eco-evolutionary regression

Calculating metric values is useful, but often we want to make statistical models. *pez* features a set of regression techniques, based on Cavender-Bares *et al.* (2006) and ?, which are described in the helpfiles for `eco.xxx.regression` and `fingerprint.regression`.

The functions described in `eco.xxx.regression` focus on relating the co-occurrence of species to species' phylogenetic (`phy`), trait (`trait`), and environmental tolerances (`env`). The environmental tolerances are based on Pianka's distance and derived from your `$env` data, while the trait distances can be based on any distance metric you can define. These are useful to explore your data, but also because the trait results are used in the *fingerprint regression* described below.

```
phy <- eco.phy.regression(data, permute = 10)
trait <- eco.trait.regression(data, permute = 10,
  method = "quantile", tau = c(0.25, 0.5,
    0.7))
trait <- eco.trait.regression(data, altogether = FALSE)
```

In the last line above, we calculated separate regressions for each trait in our dataset (returning a `eco.xxx.regression.list` object). While this isn't particularly thrilling in this dataset where we only have two traits, such a regression forms the basis of the `fingerprint.regression`. In this, we will regress the association between species co-occurrence and trait similarity for each trait against the phylogenetic conservatism of each trait. Which is a mouthful, but the papers describing it (Cavender-Bares *et al.*, 2006; ?) go into more detail. *pez* does things slightly differently to these original papers, in that it uses measures of phylogenetic 'signal' instead of Mantel tests (`phy.signal`), and provides more distance matrix and regression model options for the link between co-occurrence and trait similarity.

Figure 1 may make things clearer. On the horizontal axis we move from where there is a positive correlation between co-occurrence and each trait's similarity (left) to a negative correlation (right). On the vertical axis, traits are arranged according to whether they show trait conservation (top) or lability/lack of phylogenetic inertia (bottom). Remember: each of your traits makes up one data-point in this space, but in figure 1 we have made a cartoon of a single trait in different quadrants of the graph to make things clearer. If communities are not just assembled *but have also evolved* under limiting similarity, traits should tend to lie in the blue circle. Above the blue circle, traits have evolved under niche conservatism and habitat filtering is taking

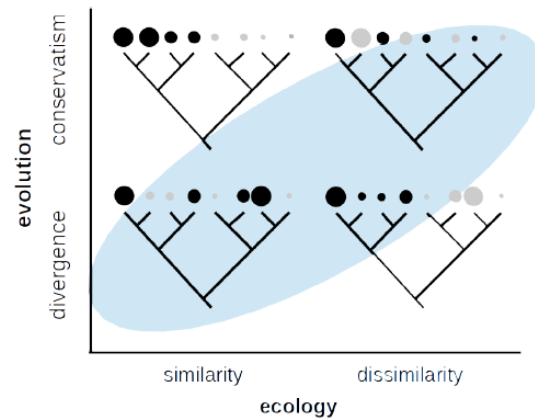


Figure 1: Overview of a fingerprint regression. A hypothetical trait is shown in each quadrant; the size of the circles represents the numerical value of the trait, and colour of the circles represents hypothetical communities.

place across those traits. By creating a regression such as this with your data, you are directly relating the present-day ecology of species to their evolutionary history.

```
model <- fingerprint.regression(data, eco.permute = 10)
## Error: unused argument (traits = TRUE)
```

Once you've performed a *fingerprint regression* (the name is new to *pez*), you can examine the coefficients of each of the `$eco` and `$evo` slots in your model, which correspond to the two axes in figure 1.

5 Functional phylogenetic distances and traitgrams

Taxa differ both functionally and phylogenetically, a fact that is clearly illustrated using traitgrams (Ackerly, 2009; Evans *et al.*, 2009). A traitgram of a `comparative.comm` object can be made using the `cc.traitgram` (a wrapper for the `traitgram` function in `picante`). Here is a traitgram for a particular assemblage of species against the `length` trait,

```
assemblage <- c("Nerophilus", "Hydroptila", "Psorophora",  
               "Simuliidae", "Psychodidae", "Ceratopogon",  
               "Nectopsyche", "Pedomoecus", "Ceratopsyche")  
dataAssemblage <- data[, species(data) %in% assemblage]  
cc.traitgram(dataAssemblage, "length")  
  
## Error: need finite 'ylim' values
```


The traitgram plots the phylogenetic time on the y-axis and **length** on the x-axis. Note that some taxa are very distantly related but nevertheless have converged to very similar trait values (e.g. Psorophora and Simuliidae), whereas others are closely related but functionally very dissimilar (e.g. Nerophilus and Nectopsyche).

Cadotte *et al.* (2013) argued that distances in this ‘traitgram space’ might provide a better indication of ecological differences between taxa, than if either function or phylogenetic data were used in isolation. 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 (Cadotte *et al.*, 2013),

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

where a is the phylogenetic weighting parameter and p is the exponent for the p -norm combination of phylogenetic and functional distances (e.g. $p = 2$ gives a Euclidean combination). This distance matrix can be computed using the `funct.phylo.dist` function. For example, for $a = 0.5$, and $p = 2$, we have,

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

One use of these distance matrices is in community randomization tests (Cadotte *et al.*, 2013), an example of which can be computed using the following code,

```
ses.mfpd.data <- picante::ses.mpd(data$comm, fpd.data)
head(ses.mfpd.data)[,c("ntaxa", "mpd.obs", "mpd.obs.p")]
```

```
##      ntaxa mpd.obs mpd.obs.p
## CA      18  0.5553    0.190
## FC      10  0.5802    0.467
## LA      12  0.5525    0.272
## LC      11  0.6463    0.896
## LQ      20  0.5535    0.129
## names    7  0.6684    0.869
```

6 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 Helmus & Ives (2012), 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.

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` and `sim.phy` functions to find out more.

Finally, you can also simulate sets of communities under phylogenetic and/or trait repulsion, using `sim.trait.asm`. These communities are excellent for use as null models to compare with your own data, and as such they take a `comparative.comm` object as an argument to generate communities that match your own data.

References

- Ackerly, D. (2009) Conservatism and diversification of plant functional traits: evolutionary rates versus phylogenetic signal. *Proc. Natl Acad. Sci.* **106**, 19699–19706.
- 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>.
- Cadotte, M., Albert, C. & Walker, S.C. (2013) The ecology of differences: assessing community assembly with trait and evolutionary distances. *Ecology Letters* **16**, 1234–1244.
- 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.
- Cavender-Bares, J., A. Keen & B. Miles (2006) Phylogenetic structure of Floridian plant communities depends on taxonomic and spatial scale. *The American Naturalist* **87**, S109–S122.
- Evans, M.E.K., Smith, S.A., Flynn, R.S. & Donoghue, M.J. (200) Climate, niche evolution, and diversification of the “Bird-cage” evening primroses (Oenothera, sections Anogra and Kleinia). *Am. Nat.* **173**, 225–240.

- 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.
- Letten, A.D. & Cornwell, W.K. (2014) Trees, branches and (square) roots: why evolutionary relatedness is not linearly related to functional distance. *Methods in Ecology and Evolution*.
- 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.