# Community phylogenetic analysis with picante

Steven Kembel (skembel@uoregon.edu)

FESIN workshop - July 26, 2009

## Contents

# 1 Installing picante

The picante homepage is located at `http://picante.r-forge.r-project.org`. From within R, you can install the latest version of picante by typing "`install.packages(picante, dependencies=TRUE)`". Typing "`help(functionName)`" will display documentation for any function in the package.

# 2  Reading and writing data

Most analyses in `picante` will use a few different types of data (community data, phylogenetic data, and trait data). The following commands load the package and the example data object `phylocom` which contains the community data (`phylocom$sample`), phylogeny (`phylocom$phylo`), and trait data (`phylocom$traits`) included with the Phylocom software:

```
> library(picante)
> data(phylocom)
> names(phylocom)

[1] "phylo"  "sample" "traits"
```

Even though we have the Phylocom example data loaded already in the `phylocom` object, let's walk through the process of loading data into R for a community phylogenetic analysis.

## 2.1  Community data

`picante` uses the `vegan` community data format: a data.frame of presences or abundances with communities in the rows and species in the columns. The names of species should match names in the phylogeny.

```
> phylocom$sample
```

|         | sp1 | sp10 | sp11 | sp12 | sp13 | sp14 | sp15 | sp17 | sp18 | sp19 | sp2 | sp20 | sp21 |
|---------|-----|------|------|------|------|------|------|------|------|------|-----|------|------|
| clump1  | 1   | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 1   | 0    | 0    |
| clump2a | 1   | 2    | 2    | 2    | 0    | 0    | 0    | 0    | 0    | 0    | 1   | 0    | 0    |
| clump2b | 1   | 0    | 0    | 0    | 0    | 0    | 0    | 2    | 2    | 2    | 1   | 2    | 0    |
| clump4  | 1   | 1    | 0    | 0    | 0    | 0    | 0    | 2    | 2    | 0    | 1   | 0    | 0    |
| even    | 1   | 0    | 0    | 0    | 1    | 0    | 0    | 1    | 0    | 0    | 0   | 0    | 1    |
| random  | 0   | 0    | 0    | 1    | 0    | 4    | 2    | 3    | 0    | 0    | 1   | 0    | 0    |

|         | sp22 | sp24 | sp25 | sp26 | sp29 | sp3 | sp4 | sp5 | sp6 | sp7 | sp8 | sp9 |
|---------|------|------|------|------|------|-----|-----|-----|-----|-----|-----|-----|
| clump1  | 0    | 0    | 0    | 0    | 0    | 1   | 1   | 1   | 1   | 1   | 1   | 0   |
| clump2a | 0    | 0    | 0    | 0    | 0    | 1   | 1   | 0   | 0   | 0   | 0   | 2   |
| clump2b | 0    | 0    | 0    | 0    | 0    | 1   | 1   | 0   | 0   | 0   | 0   | 0   |
| clump4  | 0    | 0    | 2    | 2    | 0    | 0   | 0   | 0   | 0   | 0   | 0   | 1   |
| even    | 0    | 0    | 1    | 0    | 1    | 0   | 0   | 1   | 0   | 0   | 0   | 1   |
| random  | 1    | 2    | 0    | 0    | 0    | 0   | 0   | 2   | 0   | 0   | 0   | 0   |

The example data set is based on the file `sample` included with Phylocom. The Phylocom format for community data is a tab-delimited text file. Each row of the file contains the community ID, abundance, and species ID for a single taxon, separated by tabs. To load a file in this format, we'd execute the following commands, which set the working directory to the local directory containing the file of interest and load a file in Phylocom format. This will automatically convert the file from Phylocom format to the site by species matrix format that `picante` works with:

```
> setwd("/phylocom")
> samp <- readsample("sample")
> samp
```

|         | sp1 | sp10 | sp11 | sp12 | sp13 | sp14 | sp15 | sp17 | sp18 | sp19 | sp2 | sp20 | sp21 |
|---------|-----|------|------|------|------|------|------|------|------|------|-----|------|------|
| clump1  | 1   | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 1   | 0    | 0    |
| clump2a | 1   | 2    | 2    | 2    | 0    | 0    | 0    | 0    | 0    | 0    | 1   | 0    | 0    |
| clump2b | 1   | 0    | 0    | 0    | 0    | 0    | 0    | 2    | 2    | 2    | 1   | 2    | 0    |
| clump4  | 1   | 1    | 0    | 0    | 0    | 0    | 0    | 2    | 2    | 0    | 1   | 0    | 0    |
| even    | 1   | 0    | 0    | 0    | 1    | 0    | 0    | 1    | 0    | 0    | 0   | 0    | 1    |
| random  | 0   | 0    | 0    | 1    | 0    | 4    | 2    | 3    | 0    | 0    | 1   | 0    | 0    |

|         | sp22 | sp24 | sp25 | sp26 | sp29 | sp3 | sp4 | sp5 | sp6 | sp7 | sp8 | sp9 |
|---------|------|------|------|------|------|-----|-----|-----|-----|-----|-----|-----|
| clump1  | 0    | 0    | 0    | 0    | 0    | 1   | 1   | 1   | 1   | 1   | 1   | 0   |
| clump2a | 0    | 0    | 0    | 0    | 0    | 1   | 1   | 0   | 0   | 0   | 0   | 2   |
| clump2b | 0    | 0    | 0    | 0    | 0    | 1   | 1   | 0   | 0   | 0   | 0   | 0   |
| clump4  | 0    | 0    | 2    | 2    | 0    | 0   | 0   | 0   | 0   | 0   | 0   | 1   |
| even    | 0    | 0    | 1    | 0    | 1    | 0   | 0   | 1   | 0   | 0   | 0   | 1   |
| random  | 1    | 2    | 0    | 0    | 0    | 0   | 0   | 2   | 0   | 0   | 0   | 0   |

Similarly, the process of converting community data into the format used by Phylocom can be automated. The following commands will write the site by species matrix as a file in Phylocom format and as a comma-separated matrix readable by spreadsheet programs:

```
> writesample(samp, "sample.picanteoutput.txt")
> write.csv(samp, "sample.matrix.picanteoutput.csv")
```

If the data were stored in a file in site by species format already, we could load the data in that format. For example, we can load the data back in from the comma-separated file we just wrote. Note that we need to specify that the site names are contained in the first column of the data:

```
> samp <- read.csv("sample.matrix.csv", row.names = 1)
```

## 2.2 Phylogenetic data

`picante` uses the `ape` package's `phylo` format for phylogenetic trees. Phylogenies in the Newick format can be loaded from a local file or a text string into a R phylogeny object using the `read.tree` command. More information on the Newick format for phylogenies can be found in the help for the `read.tree` function.

An example of a Newick tree would be something like "((A,B),C);". We can turn this text tree into a `phylo` object using the `read.tree` command:

```
> simpletree <- read.tree(text = "((A,B),C);")
> simpletree

Phylogenetic tree with 3 tips and 2 internal nodes.

Tip labels:
[1] "A" "B" "C"

Rooted; no branch lengths.

> plot(simpletree)
```
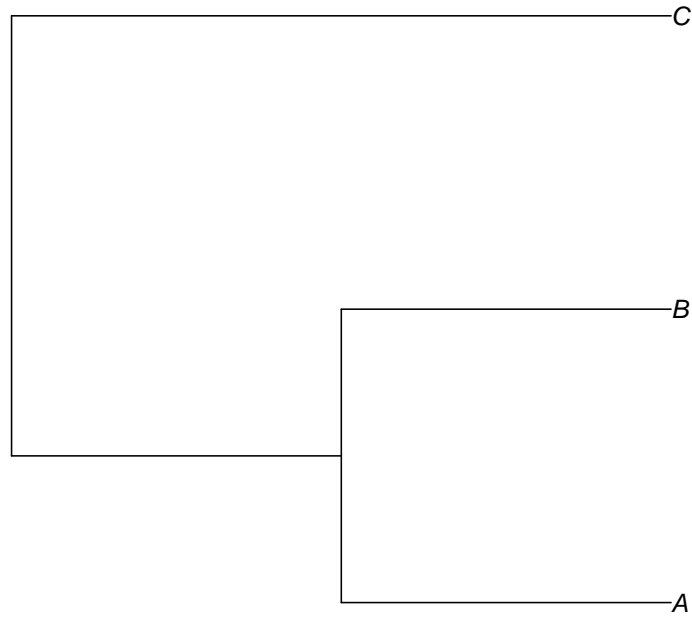
We can load the Phylocom example phylogeny from a file and look at it in Newick format. Notice that the phylogeny does not have any branch lengths. Many comparative analyses require phylogenetic branch lengths, so we'll then set all branch lengths equal to 1 using the `compute.brlen` function and look at it again:

```
> phy <- read.tree("phylo")
> phy

Phylogenetic tree with 32 tips and 31 internal nodes.

Tip labels:
        sp1, sp2, sp3, sp4, sp5, sp6, ...
        Node labels:
        A, B, C, D, E, F,...
```
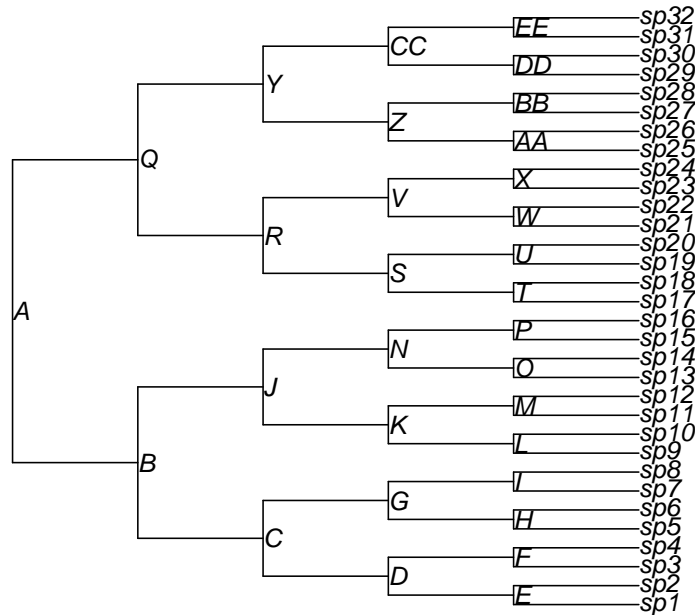
Rooted; no branch lengths.

```
> write.tree(phy)
```

[1] "(((((sp1,sp2)E,(sp3,sp4)F)D,((sp5,sp6)H,(sp7,sp8)I)G)C,(((sp9,sp10)L,(sp11,sp12)

```
> phy <- compute.brlen(phy, 1)
> phy
```

Phylogenetic tree with 32 tips and 31 internal nodes.

Tip labels:
        sp1, sp2, sp3, sp4, sp5, sp6, ...
        Node labels:
        A, B, C, D, E, F,...

Rooted; includes branch lengths.

```
> write.tree(phy)
```

[1] "((((((sp1:1,sp2:1)E:1,(sp3:1,sp4:1)F:1)D:1,((sp5:1,sp6:1)H:1,(sp7:1,sp8:1)I:1)G:1

```
> plot(phy, show.node.label = TRUE)
```

## 2.3 Trait data

Data on ecological or phenotypic attributes of species can be stored in a data.frame or vector object. As with the community data, the trait data need to have species names that match the names in the phylogeny. We can read the trait data from a tab-delimited file using the `read.table` command. The Phylocom format for trait data includes an extra line of information that we will remove, and we will get species names from the first column of the file.

```
> traits <- read.table("traits", skip = 1, header = TRUE,
+     row.names = 1)
> traits

     traitA traitB traitC traitD
sp1       1      1      1      0
```

7

```
sp2      1      1      2      0
sp3      2      1      3      0
sp4      2      1      4      0
sp5      2      2      1      0
sp6      2      2      2      0
sp7      2      2      3      0
sp8      2      2      4      0
sp9      1      3      1      1
sp10     1      3      2      1
sp11     2      3      3      1
sp12     2      3      4      1
sp13     2      4      1      1
sp14     2      4      2      1
sp15     2      4      3      1
sp16     2      4      4      1
sp17     1      1      1      1
sp18     1      1      2      1
sp19     2      1      3      1
sp20     2      1      4      1
sp21     2      2      1      1
sp22     2      2      2      1
sp23     2      2      3      1
sp24     2      2      4      1
sp25     1      3      1      1
sp26     1      3      2      1
sp27     2      3      3      1
sp28     2      3      4      1
sp29     2      4      1      1
sp30     2      4      2      1
sp31     2      4      3      1
sp32     2      4      4      1
```

If we have a data.frame of trait information in R and want to save it to a file that Phylocom could read, the `writetraits` function will take a data.frame and generate the file format that Phylocom expects, adding a first line that includes an indication of the data type of each trait. Or we could simply write the traits to a comma-separated file readable by your favorite spreadsheet program.

```
> writetraits(traits, "traits.picanteoutput.txt")
> write.csv(traits, "traits.picantecsvoutput.csv")
```

# 3   Visualizing trees and data

One of the main advantages of using R is that a suite of graphical and statistical tools
are included. Now that we've loaded our data sets, we can use some of those tools
to visualize them. Remember that we have three objects containing the community
(samp), phylogeny (phy), and trait (traits) data sets.

```
> samp <- phylocom$sample
> phy <- phylocom$phylo
> traits <- phylocom$traits
```

As a prelude to the analyses we're about to run, let's look at how species in some
of the communities in the example data set we just loaded are distributed across the
tips of the phylogeny. To do this, we first need to prune the phylogeny to include
only the species that actually occurred in some community.

```
> prunedphy <- prune.sample(samp, phy)
> prunedphy

Phylogenetic tree with 25 tips and 24 internal nodes.

Tip labels:
        sp1, sp2, sp3, sp4, sp5, sp6, ...
        Node labels:
        A, B, C, D, E, F,...

Rooted; includes branch lengths.
```

We also need to make sure the species are arranged in the some order in the
community data and the phylogeny. This is an important step - several functions in
picante assume that the community or trait data and phylogeny data have species
arranged in the same order, so it's good to always make sure we've done so before
running any analysis. The following command sorts the columns of samp to be in
the same order as the tip labels of the phylogeny:

```
> samp <- samp[, prunedphy$tip.label]
```

Now let's see how taxa from the six communities in the Phylocom example data set are arranged on the tree. The following commands set up the layout of the plot to have 2 rows and 3 columns, and then plot a black dot for the species present in each of the six communities:

```
> par(mfrow = c(2, 3))
> for (i in row.names(samp)) {
+       plot(prunedphy, show.tip.label = FALSE, main = i)
+       tiplabels(tip = which(samp[i, ] > 0), pch = 19, cex = 2)
+ }
```

clump1            clump2a            clump2b

clump4              even              random
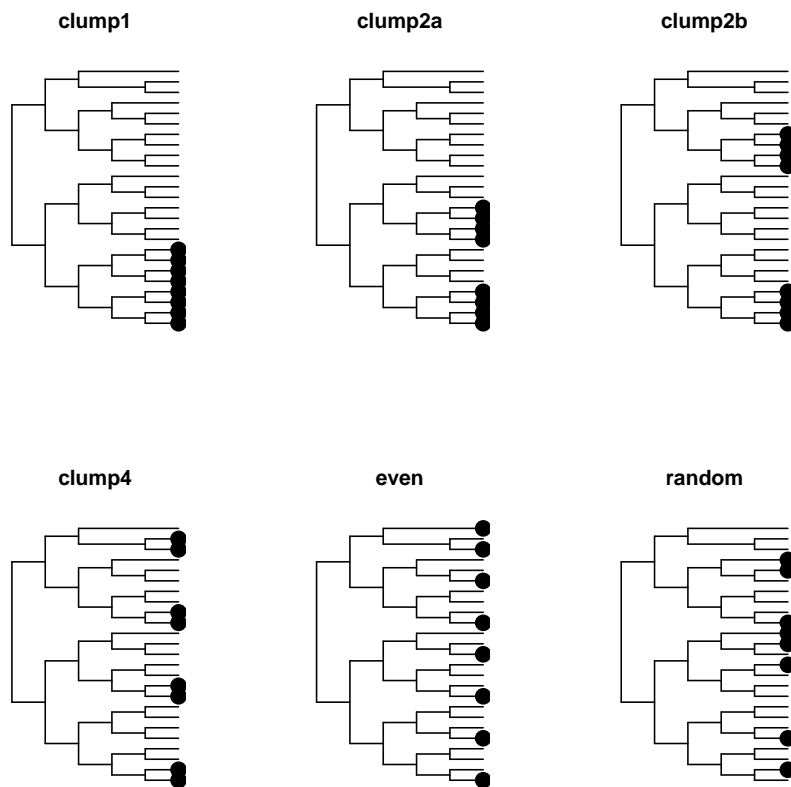
Similarly, let's visualize the trait values on the trees by plotting a different color for each trait value. The arguments to the tiplabels function give each trait value a unique color and adjust the size of the trait symbols.

10

```
> par(mfrow = c(2, 2))
> for (i in names(traits)) {
+     plot(phy, show.tip.label = FALSE, main = i)
+     tiplabels(pch = 22, col = traits[, i] + 1, bg = traits[,
+         i] + 1, cex = 1.5)
+ }
```
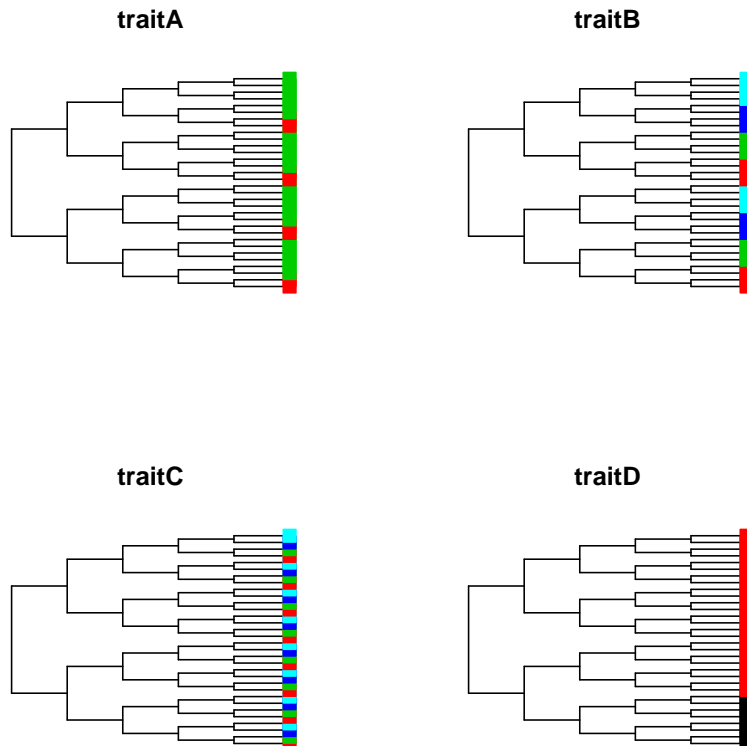


Looking at these plots, which communities would you expect to be phylogenetically clustered or phylogenetically even? Which of the four traits do you think has the greatest amount of phylogenetic signal?

# 4 Community phylogenetic structure

## 4.1 Phylogenetic diversity

One of the earliest measures of phylogenetic relatedness in ecological communities was the phylogenetic diversity (PD) index proposed by Faith (1992). Faith's PD is defined as the total branch length spanned by the tree including all species in a local community. The `pd` function returns two values for each community, the PD and the species richness (SR).

```
> pd.result <- pd(samp, phy, include.root = TRUE)
> pd.result

       PD SR
clump1  16  8
clump2a 17  8
clump2b 18  8
clump4  22  8
even    30  8
random  27  8
```

Looking at these results, we can see that the communities where taxa are clumped on the phylogeny tend to have a lower PD, because the species in these communities capture only a small part of the total phylogenetic diversity present in the phylogeny.

## 4.2 MPD, MNTD, $SES_{MPD}$ and $SES_{MNTD}$

Another way of thinking about the phylogenetic relatedness of species in a community is to ask 'how closely related are the average pair of species or individuals in a community?', and relate the patterns we observe to what we'd expect under various null models of evolution and community assembly. These types of questions are addressed by the measures of community phylogenetic structure such as MPD, MNTD, NRI and NTI described by Webb et al. (2002) and implemented in Phylocom (Webb et al. 2008).

The function `mpd` will calculate the mean pairwise distance (MPD) between all species in each community. Similarly, the `mntd` function calculates the mean nearest taxon distance (MNTD), the mean distance separating each species in the community from its closest relative. The `mpd` and `mntd` functions differs slightly from the `pd` function in that they take a distance matrix as input rather than a phylogeny object.

A `phylo` object can be converted to a interspecific phylogenetic distance matrix using the `cophenetic` function. Since the `mpd` and `mntd` functions can use any distance matrix as input, we could easily calculate trait diversity measures by substituting a trait distance matrix for the phylogenetic distance matrix.

Measures of 'standardized effect size' of phylogenetic community structure can be calculated for MPD and MNTD by compared observed phylogenetic relatedness to the pattern expected under some null model of phylogeny or community randomization. Standardized effect sizes describe the difference between phylogenetic distances in the observed communities versus null communities generated with some randomization method, divided by the standard deviation of phylogenetic distances in the null data:

$$SES_{metric} = \frac{Metric_{observed} - mean(Metric_{null})}{sd(Metric_{null})}$$

Phylocom users will be familiar with the measures NRI and NTI; $SES_{MPD}$ and $SES_{MNTD}$ are equivalent to -1 times NRI and NTI, respectively, when these functions are run with a phylogenetic distance matrix.

Several different null models can be used to generate the null communities that we compare observed patterns to. These include randomizations of the tip labels of the phylogeny, and various community randomizations that can hold community species richness and/or species occurrence frequency constant. These are described in more detail in the help files, as well as in the Phylocom manual. Let's calculate some of these measures of community phylogenetic structure for our example data set. We will use a simple null model of randomly shuffling tip labels across the tips of the phylogeny. For a 'real' analysis we'd want to use a much higher number of runs:

```
> phydist <- cophenetic(phy)
> ses.mpd.result <- ses.mpd(samp, phydist, null.model = "taxa.labels",
+     abundance.weighted = FALSE, runs = 99)
> ses.mpd.result

        ntaxa  mpd.obs mpd.rand.mean mpd.rand.sd mpd.obs.rank
clump1      8 4.857143      8.338384   0.3122826            1
clump2a     8 6.000000      8.308081   0.3010890            1
clump2b     8 7.142857      8.347042   0.2902357            1
clump4      8 8.285714      8.329004   0.2860454           40
even        8 8.857143      8.294372   0.3539614          100
random      8 8.428571      8.309524   0.3490333           58
```

```
        mpd.obs.z mpd.obs.p runs
clump1  -11.1477280      0.01   99
clump2a  -7.6657767      0.01   99
clump2b  -4.1489883      0.01   99
clump4   -0.1513398      0.40   99
even      1.5899205      1.00   99
random    0.3410781      0.58   99

> ses.mntd.result <- ses.mntd(samp, phydist, null.model = "taxa.labels",
+     abundance.weighted = FALSE, runs = 99)
> ses.mntd.result

        ntaxa mntd.obs mntd.rand.mean mntd.rand.sd mntd.obs.rank
clump1      8        2       4.623737    0.6882521           1.0
clump2a     8        2       4.611111    0.6490006           1.0
clump2b     8        2       4.702020    0.7263945           1.5
clump4      8        2       4.694444    0.6493281           1.0
even        8        6       4.646465    0.6757555          99.5
random      8        5       4.739899    0.5845799          65.0
        mntd.obs.z mntd.obs.p runs
clump1  -3.8121747      0.010   99
clump2a -4.0232798      0.010   99
clump2b -3.7197697      0.015   99
clump4  -4.1495885      0.010   99
even     2.0029956      0.995   99
random   0.4449366      0.650   99
```

The output includes the following columns:

- ntaxa Number of taxa in community

- mpd.obs Observed mpd in community

- mpd.rand.mean Mean mpd in null communities

- mpd.rand.sd Standard deviation of mpd in null communities

- mpd.obs.rank Rank of observed mpd vs. null communities

- mpd.obs.z Standardized effect size of mpd vs. null communities (equivalent to -NRI)

14

- `mpd.obs.p` P-value (quantile) of observed mpd vs. null communities (= mpd.obs.rank / runs + 1)

- `runs` Number of randomizations

Positive $SES$ values (`mpd.obs.z` $> 0$) and high quantiles (`mpd.obs.p` $> 0.95$) indicate phylogenetic evenness, or a greater phylogenetic distance among co-occurring species than expected. Negative SES values and low quantiles (`mpd.obs.p` $< 0.05$) indicate phylogenetic clustering, or small phylogenetic distances among co-occurring species than expected. MPD is generally thought to be more sensitive to tree-wide patterns of phylogenetic clustering and eveness, while MNTD is more sensitive to patterns of evenness and clustering closer to the tips of the phylogeny. For example, community 'clump4' contains species that are spread randomly across the entire tree ($SES_{MPD}$ close to zero) but phylogenetically clustered towards the tips (negative $SES_{MNTD}$ and `mntd.obs.p` in the low quantiles of the null distribution).

All of these measures can incorporate abundance information when available using the `abundance.weighted` argument. This will change the interpretation of these metrics from the mean phylogenetic distances among species, to the mean phylogenetic distances among individuals.

## 4.3   Phylogenetic beta diversity

We can measure patterns of phylogenetic relatedness among communities in a manner similar to the within-community measures described above. The `comdist` and `comdistnt` functions measure the among-community equivalent of MPD and MNTD, the mean phylogenetic distance or mean nearest taxon distance between pairs of species drawn from two distinct communities.

Phylogenetic beta diversity measures can be used with any method based on measuring among-community distances. For example, they could be used in a cluster analysis or phyloordination to group communities based on their evolutionary similarity, or they could be compared with spatial or environmental distances separating communities using a Mantel test. The code below calculates MPD between pairs of communities, and uses these phylogenetic distances to cluster communities based on their phylogenetic similarity:

```
> comdist.result <- comdist(samp, phydist)
> comdist.result

        clump1 clump2a clump2b  clump4     even
clump2a 6.12500
```
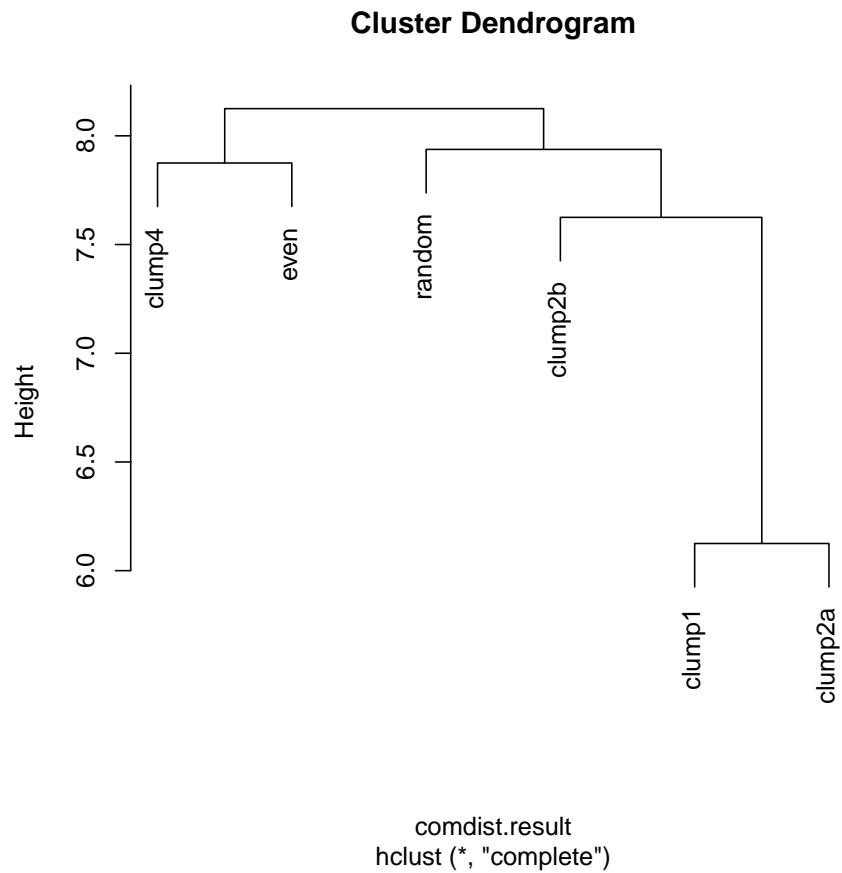
15

```
clump2b 7.12500 7.62500
clump4  8.06250 7.62500 7.62500
even    8.06250 8.06250 8.06250 7.87500
random  7.81250 7.68750 7.93750 8.12500 8.03125

> library(cluster)
> comdist.clusters <- hclust(comdist.result)
> plot(comdist.clusters)
```

**Cluster Dendrogram**



comdist.result
hclust (*, "complete")

# 5 Comparative analyses

## 5.1 Phylogenetic signal

The idea of phylogenetic niche conservatism (the ecological similarity of closely related species) has attracted a lot of attention recently, for example in the widely used framework of inferring community assembly processes based on knowledge of community phylogenetic structure plus the phylogenetic conservatism of traits. (Webb et al. 2002).

Phylogenetic signal is a quantitative measure of the degree to which phylogeny predicts the ecological similarity of species. The $K$ statistic is a measure of phylogenetic signal that compares the observed signal in a trait to the signal under a Brownian motion model of trait evolution on a phylogeny (Blomberg et al. 2003). $K$ values of 1 correspond to a Brownian motion process, which implies some degree of phylogenetic signal or conservatism. $K$ values closer to zero correspond to a random or convergent pattern of evolution, while $K$ values greater than 1 indicate strong phylogenetic signal and conservatism of traits. The statistical significance of phylogenetic signal can be evaluated by comparing observed patterns of the variance of independent contrasts of the trait to a null model of shuffling taxa labels across the tips of the phylogeny.

These tests are implemented in the `Kcalc`, `phylosignal`, and `multiPhylosignal` functions. All of these functions assume the trait data are in the same order as the phylogeny tip.labels. Let's make sure the Phylocom trait data are in this order and then measure phylogenetic signal in these data.

```
> traits <- traits[phy$tip.label, ]
> multiPhylosignal(traits, phy)

              K PIC.variance.obs PIC.variance.rnd.mean PIC.variance.P
traitA 0.8905609       0.05396825             0.1248991          0.001
traitB 2.9340184       0.10920635             0.8335175          0.001
traitC 0.5149502       0.62222222             0.8403453          0.058
traitD 4.3536696       0.01103943             0.1251168          0.001
       PIC.variance.Z
traitA      -3.670123
traitB      -5.409741
traitC      -1.613841
traitD      -5.742619
```

The higher the $K$ statistic, the more phylogenetic signal in a trait. `PIC.variance.P` is the quantile of the observed phylogenetically independent contrast variance versus the null distribution, which can be used as a 1-tailed P-value to test for greater phylogenetic signal than expected. Traits with `PIC.variance.P` $< 0.05$ have non-random phylogenetic signal.

# 6  Literature cited

- Blomberg, S. P., T. Garland, Jr., and A. R. Ives. 2003. Testing for phylogenetic signal in comparative data: behavioral traits are more labile. Evolution 57:717-745.

- Faith, D.P. 1992. Conservation evaluation and phylogenetic diversity. Biological Conservation, 61:1-10.

- Webb, C., D. Ackerly, M. McPeek, and M. Donoghue. 2002. Phylogenies and community ecology. Annual Review of Ecology and Systematics 33:475-505.

- Webb, C.O., Ackerly, D.D., and Kembel, S.W. 2008. Phylocom: software for the analysis of phylogenetic community structure and trait evolution. Version 4.0.1. `http://www.phylodiversity.net/phylocom/`