PHYLOCOM

SOFTWARE FOR THE ANALYSIS OF PHYLOGENETIC COMMUNITY STRUCTURE AND CHARACTER EVOLUTION (WITH PHYLOMATIC AND ECOVOLVE)

USER'S MANUAL

VERSION 4.2

CAM WEBB
ARNOLD ARBORETUM OF HARVARD UNIVERSITY
CWEBB@OEB.HARVARD.EDU

DAVID ACKERLY
UNIVERSITY OF CALIFORNIA, BERKELEY
DACKERLY@BERKELEY.EDU

STEVE KEMBEL
UNIVERSITY OF OREGON
SKEMBEL@UOREGON.EDU

CONTENTS

1 INTRODUCTION

PHYLOCOM is a command-line application for manipulating ecological and phylogenetic data, calculating various metrics of phylogenetic and phenotypic community structure, and measuring trait conservatism and trait correlations.

We have developed a system to help take the evolutionary ecologist easily through the steps needed in an analysis of phylogenetic community structure or trait evolution. PHYLOMATIC can be used to rapidly develop a phylogeny for any plant community. This phylogeny can then be input into the PHYLOCOM program to measure phylogenetic relatedness among species occurring together in samples, test hypotheses of community structure, or quantify patterns of trait evolution. If estimates exist for the ages of any node, these can be incorporated, as can branch lengths from other sources. ECOVOLVE is a phylogeny growth simulator, using the same file formats as the other tools. These tools will remain command-line programs, so that they can easily be used inside other programs and shell-scripts.

1.1 New in Version 4.2

- Phylomatic no longer outputs a branch length for the root node; the presence of this BL (allowed by the Newick definition) was causing parsing errors in other applications.

1.2 New in Version 4.1

- Renamed `comdistnn` function to `comdistnt` for consistency
- Added null model testing to `comdist`/ `comdistnt` functions
- Windows version now compiled with MINGW32, rather than DJGPP. This should help with some of the memory issues some Windows users were experiencing. A `phylocom.bat` file is also included to assist in opening a `CMD.EXE` console window.

1.3 New in Version 4.0

- Added code to detect line endings (Mac/Windows/Unix) and adjust automatically.
- Added `rao` function to calculate phylogenetic diversity.
- Updated `comstruct` and `comdist` to use -a switch to incorporate abundance into phylogenetic distance calculations. NB: see **?** for a discussion of important issues concerning abundances and Type I and II errors in detection of significant phylogenetic structure.
- `comtrait` function calculates trait dispersion within communities.
- Modified calculation of phylogenetic signal.

2 INSTALLATION

2.1 Mac OS X

Universal binaries for OS 10.5 are included in the `mac` directory. If these do not run (perhaps if you are using an older version of OS X), you may want to build from scratch: acquire the OS X Developer Tools Installer from the Apple website (you may have to register as a developer), or from your installation discs, and install. Then follow the instructions below for a UNIX build.

This is **command-line software**. You need to run it in a terminal window (`/Applications/Utilities/Terminal.app`).

To make PHYLOCOM available anywhere (i.e., not always requiring the executable in the same directory as your data files), create a `bin/` directory under your home directory, and add this line to your `.bash_profile` file:

```
PATH=$PATH:$HOME/bin:. ; export PATH
```

or this line to your `.tcshrc` file, if you are running the `tcsh` shell:

```
set path=(. ~/bin $PATH)
```

2.2 Linux/Other Unix

Rather than provideing precompiled binaries for each architecture, unix users should compile locally. These commands should work:

```
$ cd Desktop          # or to wherever you save the zip file
$ unzip phylocom.zip
$ cd phylocom-X       # replace X with version no.
$ cd src
$ make
$ ./phylocom
```

If your system does not use `gcc`, edit the `Makefile` to reflect your C-compiler.

To make PHYLOCOM available anywhere (i.e., not always requiring the executable in the same directory as your data files), create a `bin/` directory under your home directory, and add this line to your `.bash_profile` file:

```
PATH=$PATH:$HOME/bin:. ; export PATH
```

or this line to your `.tcshrc` file, if you are running the `tcsh` shell:

```
set path=(. ~/bin $PATH)
```

2.3 Windows

Windows 32-bit binaries are included in the `w32` directory. These binaries were compiled with mingw32 under Linux (see `Makefile`). The binaries run in the Windows console (usually found at: `c:\windows\system32\cmd.exe`); note that this shell is no longer the same as MS-DOS, although most of the commands behave as before (see http://commandwindows.com/ for an excellent introduction to the Windows console).

If you need to recompile in Windows, use `mingw32`, or you can install and use the `cygwin` tools.

In order to access PHYLOCOM from anywhere in your directory tree, create a directory where you want `phylocom.exe` to live (e.g., `C:\PHYLOCOM`). Right click the My Computer icon, choose Properties, then click on advanced system settings, then click on environment variables. From there, add the name of the directory where you have placed the PHYLOCOM executable to the end of the list of directories in the path. Copy the executables (`.exe` files) to this new directory, open a new command prompt window, and you should be able to type `phylocom` and have the program run. This is the same as adding these lines to a batch file:

```
path = %PATH%;C:\phylocom
```

This is **command-line software**. You need to run it in a console window. You can acces the console via (one of):

1. Start Menu → Applications → Accessories → Command prompt
2. Start Menu → Run, and type `CMD`
3. Double clicking on the `phylocom.bat` file. This must be in the same directory as the executables, unless you have set the path, as above.

If you are frequently using a particular PHYLOCOM function, you can also make custom shortcuts (a tip from an anonymous reviewer):

1. Place your executable `phylocom.exe` in a standard location (e.g., `"C:\PROGRAM FILES\PHYLOCOM\"`)
2. Create a shortcut to `phylocom.exe` and rename it (e.g., `phylocom_comstruct`).
3. Opening the properties of the shortcut, and in the Target box, enter `"C:\PROGRAM FILES\PHYLOCOM\PHYLOCOM.EXE" COMSTRUCT > OUT.TXT`
4. Leave the 'Run In' box blank

The shortcut can now be copied and dropped into any directory where you want to run an analysis (with data files named to the default names). Alternatively, you can put the following (all in one line) in the 'Target' box, to run the program in a console, which will stay open after the program has run:

```
C:\WINDOWS\system32\cmd.exe /k
"c:\program files\phylocom\phylocom.exe"
comstruct > out.txt
```

3 INPUT FILE FORMATS

PHYLOCOM uses plain text files as input (i.e., not propriety binary formats such as `.xls` and `.doc`). This facilitates connecting the software as part of an analysis or simulation chain, because text processing tools (e.g., `sed`, `awk`, `perl`) can be used to re-format files, etc. However, because one has access to the internals of the data files in this way, one must be extra careful with accuracy of formatting: no extra spaces, tabs, etc. Learning to use a good text editor will greatly help; we recommend TextWrangler for Mac and Notepad-plus for Windows.

Note: As of version 4.0, PHYLOCOM recognizes the line-endings (UNIX, Mac, Windows) used in each file. This should save a lot of wasted time! However, your line-endings must be consistent within each file.

3.1 Tree preparation

PHYLOCOM reads Newick-format phylogenies directly. See:

http://evolution.genetics.washington.edu/phylip/newicktree.html, and

http://evolution.genetics.washington.edu/phylip/newick_doc.html

for definitions of the Newick standard. The default file name is `phylo`, but other phylogeny files can be specified using the `-f` filename option. Plain Newick format phylogenies are also used by PHYLIP.

The basic Newick format used by PHYLOCOM is:

```
((A,B),C);
```

The full complexity Newick format that can be read by PHYLOCOM is:

```
((A_sp:1.1,B2-sp:2.2)clade1:1.0[a comment],c_sp:0.5);
```

Please note:

- Taxa and interior names must begin with `A-Z` or `a-z`, not a number,
- Branch lengths will be assumed to be 1.0 if not present,
- Branch lengths *must not be in exponential format* (i.e., N.NNN, not N.NNNe-XX),
- Comments are ignored,
- The root node should not have a branch length if other branch lengths are given,
- There should be no whitespace or line-endings within the tree,
- Trailing spaces/line-endings at the end of the phylo file are OK, but not in any other file format used by PHYLOCOM,
- Multiple Newick strings (as in PHYLIP `intree` files) are currently not allowed in the `phylo` file,
- The basal node must be a dichotomy, not a polytomy.

3.1.1 Newick and NEXUS

One of the standard file formats for phylogenies is NEXUS. If you open a NEXUS file with a text editor, you will see one or more Newick strings in the TREES section. You may be able to simply cut out the Newick string and paste into a new `phylo` file. However, some programs write by default a translation of the taxon names into numbers. These translated Newick strings need to be 'un-translated' before using in PHYLOCOM. A simple way to get a Newick string out of a NEXUS file is to open the NEXUS file in Mesquite, open a tree window (Taxa&Trees → New Tree Window) for the stored tree (Stored Trees), click on the 'Text' tab, and save the text page to a file (File → Save Window as Text...). In the saved file, edit out everything but the Newick string.

3.2 Sample preparation

PHYLOCOM accepts various kinds of sample data. Samples may represent subsets of the phylogenetic tree, or ecological data matrices (measurements of species abundance or occurrence in samples of some sort). The default file name is `sample`, but other sample files can be specified using the `-s filename` option.

The sample file has the following format:

- 3 columns, tab delimited, sorted by column 1
- one row per taxon:
  1. Sample (plot, quadrat, trap, etc.) name (character string, no spaces, should begin with `[A-Za-z]`)
  2. Abundance (integer; leave as 1 for presence/absence data)
  3. Species code (string, same as in `phylo`, should begin with `[A-Za-z]`)
- all species in this table MUST be included in `phylo`

You can wrestle your data into this format pretty easily, using a stats package or spreadsheet. Look at the included example file `sample` for an idea of what the file should look like. The current version has been tested to work with fairly large ecological data sets as sample files (e.g., 400 species and 5,000 samples).

3.3 Traits preparation

Any number of characters can be included in the `traits` file. Note that missing trait values are not allowed, and the list of taxa in `traits` must exactly match the terminal taxa in the phylo file. The default file name is `traits`, but other trait files can be specified using the `-t filename` option.

The first line of traits must read:

```
type<TAB>n<TAB>n<TAB>... [up to the number of traits]
```

where `n` indicates the type of trait in each of the four columns:

`0` for binary (only one binary trait may be included, and it must be in the first column)

`1` for unordered multistate (no algorithms currently implemented)

`2` for ordered multistate (currently treated as continuous)

`3` for continuous

Optional: The second line can start with the word `name` (lower case only) and then list the names of the traits in order. These will appear in the full output file.

Subsequent lines should have the taxon name, which must be identical to its appearance in `phylo`, and the data columns separated by tabs. See the example traits file distributed with the program.

4 USING PHYLOCOM

NOTE: Investing in a basic guidebook for UNIX or DOS/Windows console will be very worthwhile in the long run. Alternatively, the web is full of useful pages, e.g., Google: 'introduction to unix' or 'commandline Windows'.

Try this software first with the included example files. Either i) copy these files to the same directory as the executables, or ii) make the executables universally findable on your system (see § 2.3), and just `cd` to where your (demo or real) input files are. Then just type:

```
$ phylocom
```

(NOTE, the `$` symbol in this manual indicates the *command prompt*; do not type this symbol, only what follows it). If you are using OS X/UNIX and haven't placed the PHYLOCOM executable in your path (see § 2.3), type:

```
$ ./phylocom
```

In Windows console, just double-click on the `phylocom.bat` file, or manually `cd` to the correct directory, and type:

```
C:\SOME\PATH>PHYLOCOM.EXE
```

or just:

```
C:\SOME\PATH>phylocom
```

A welcome screen should appear. The format for the various options is:

```
$ phylocom method [optional parameters]
```

Basic information can be obtained at any time by calling:

```
$ phylocom help
```

Output from the programs is written to the screen (generally `/dev/stdout`, although some warnings are sent to `/dev/stderr`). In order to capture the output into a file, it must be redirected to a filename, e.g.:

```
$ phylocom comstruct > myoutput.txt
```

or:

```
$ phylocom bladj > mytree.new
```

This syntax works on both UNIX systems and in the Windows console.

5 BASIC DATA EXTRACTION AND MANIPULATION

A number of algorithms have been included to assist in summarizing data and converting between formats. These are generally simple to write, and we welcome suggestions for new facilities.

5.1 AGENODE

Outputs the ages for each node in the phylogeny, calculated from the branch lengths. The numbering system is the same as throughout this application. The root node is node 0, and nodes are numbered incrementally reading across the parentheses in the Newick input tree.

5.2 AGETERM

Outputs the stem age of each terminal taxon (age of each taxon's most recent ancestor node).

5.3 BLADJ

What do you do if you have a phylogenetic topology, with some nodes aged, but no branch lengths to smooth the rates of (with `r8s`)? You can still use `r8s` without branch lengths to force an ultrametric tree. Or you can use `bladj`!

This is a simple utility that takes a phylogeny, fixes the root node at a specified age, and fixes other nodes you might have age estimates for. It then sets all other branch lengths by placing the nodes evenly between dated nodes, and between dated nodes and terminals (beginning with the longest 'chains'). This has the effect of minimizing variance in branch length, within the constraints of dated nodes. It thus produces a pseudo-chronogram that can be useful for estimating phylogenetic distance (in units of time) between taxa for, for instance, the analysis of phylogenetic community structure. Even with only a few nodes dated, the resulting phylogenetic distances can be a marked improvement on simply using the number of intervening nodes as a phylogenetic distance (see **?**).

BLADJ takes as its input a phylogeny (the `phylo` file), with named internal nodes, and a simple table of interior node names and ages (the `ages` file, format: `name<TAB>age<RETURN>`; **NB**: node names need to match exactly, including case, between the `ages` and `phylo` files). It returns a new phylogeny with adjusted branch lengths. IMPORTANT: the root node of the phylogeny must be named and given an age.

Included in the distribution is a simple ages file (called `wikstrom.ages`) with angiosperm nodes aged according to **?**. I fully acknowledge that these ages are not the maximum age for, e.g. a family, but simply an estimate of the MRCA of the two most distant taxa in a clade included in Wikstrom's analysis. The correct statement is that the clade represented by this node is at least as old as the age given, and no older than the age of the next older node dated in the list. We all await an online database of fossil-based estimates of node age!

Make sure a file named `ages` is present in the same directory as the `phylo` file. Then run:

```
$ phylocom bladj
```

or:

```
$ phylocom bladj > output_tree.new
```

The output Newick tree will be ultrametric, with branch lengths scales to time.

Please Note: If a name in the `ages` file matches a *terminal* taxon in the tree, that terminal taxon will be positioned at the corresponding age, and not at an age of 0. The resulting tree will not be ultrametric. If this non-ultrametric tree is used as an input tree to phylomatic, the resulting tree will also not be ultrametric. To avoid this problem, build your phylomatic trees first, and apply BLADJ to the final tree.

5.4 CLEANPHY

Removes 'one-daughter nodes' from a `phylo` phylogeny, and the branch length of the root node. One-daughter nodes are allowed in the Newick definition, and are useful for storing information about hierarchical taxonomic classes. PHYLOMATIC includes numerous one-daughter nodes, and because most other phylogenetic applications do not accept one-daughter nodes, these need to be 'cleaned out.' Root 'tails' are also allowed by the definition, but many other applications choke on them. Run:

```
$ phylocom cleanphy -f phylomatic_out.new -e > clean.new
```

to remove one-daughter nodes from a file `phylomatic_out.new`. The `-e` switch suppresses the creation of automatic branch lengths of 1.0.

5.5 COMNODE

A simple consensus algorithm that finds the common nodes in two trees, named `tree1` and `tree2`. Creates common names for the matching internal nodes and outputs a simple Nexus format tree readable by most tree-viewing software (e.g., TreeView).

This tool can be used to add branch lengths to a supertree:

1. Let `tree1` be a phylogeny with branch lengths from, e.g., molecular analysis, including a few of the species in the supertree.

2. Let `tree2` be a supertree, without branch lengths, containing some of the taxa in `tree1`.

3. Run `phylocom comnode > out.nex`

4. Extract 'tree1' from the output file into a new `phylo` file

5. Run `phylocom agenode > ages`

6. Edit the `ages` file to only leave the lines begining ‘match.’
7. Extract ‘tree2’ from the output file into a new `phylo` file
8. Run `phylocom bladj`. The output tree contains all the taxa in the supertree, with branch lengths constrained by `tree1`.

See the included files `tree1` and `tree2` and run `phylocom comnode` to test the algorithm. See **?** for an example.

5.6 MAKENEX

Reads a `phylo` file, a `sample` file, and a `traits` file and outputs a NEXUS file readable by Mesquite. It includes up to four CHARACTER blocks with:

1. taxa presence or absence in the various samples coded as 0 or 1.
2. taxa abundance in the various samples as a continuous variable. Hint: want to know whether there is a phylogenetic signal in abundance? Make a sample unit in a `sample` file with all taxa in the `phylo` file. Run `phylocom makenex`. Open in Mesquite. Choose ‘Trace Character History.’ Test significance of any conservative trend in abundance by making a continuous trait that is abundance, and running `aot`.
3. Any discrete characters in the traits file.
4. Any continuous characters in the traits file. Use ‘Trace Character History’ to view the distribution of traits and/or species presence/absence in samples on the pool phylogeny.

Note that currently all three input files are needed. Create a dummy `traits` or `sample` file if needed.

5.7 NEW2NEX

Converts a Newick file (the `phylo` file) to a Mesquite-readable NEXUS-format file. Note that this function is very similar to the MAKENEX function, but does not require `sample` and `traits` files as input.

5.8 NEW2FY

Converts a Newick file (the `phylo` file) to a simple tabular format, with each node as a row. Tab-delimited columns are:

- `nodeID`
- parent node `nodeID`
- number of daughter nodes
- *partial* list of daughter `nodeIDs`
- depth of node (number of edges from root)
- branch length to parent node (a float)
- node name

5.9 PHYDIST

Calculates the simple pairwise matrix of phylogenetic distances among terminal taxa for the whole phylogeny pool (`phylo`). This could be useful even if you are not interested in community structure. The column and row headings are terminal names in the `phylo` file.

5.10 PHYVAR

Calculates the phylogenetic variance-covariance matrix: approximately the 'inverse' of the of phylogenetic distance matrix—taxa that are closely related have high phylogenetic covariance.

5.11 NAF

Convert all data files (`sample`, `phylo`, `traits` all needed) into a 'node-as-factor' table, for analysis of trait values (or sample abundance values) by simple or hierarchical ANOVA. All taxa subtending to a particular daughter node are coded with a similar value in a column for each node. Hence variance in a trait for terminals in one clade can easily be compared to variance in terminals in the sister clade.

5.12 RNDPRUNE

Randomly prunes the `phylo` phylogeny. Two switches control the output:

`-r` N: performs the randomization N times,

`-p` N: includes N terminals.

The randomization simply selects randomly (from an even distribution) from the names of the terminals in `phylo`.

5.13 SAMPLEPRUNE

Prunes a `phylo` phylogeny by the members of each sample unit in the `sample` file.

5.14 VERSION

Outputs the version of PHYLOCOM.

6 PHYLOGENETIC COMMUNITY STRUCTURE

6.1 Phylogenetic community structure metrics

6.1.1 PD

Calculates Faith's (**?**) index of phylogenetic diversity (PD) for each sample in the phylo. Faith's PD index (total branch length among all taxa in a sample, including the root node of the tree) is reported, as are the total branch length in the phylogeny, and the proportion of the total branch length in the phylogeny associated with the taxa in each sample.

6.1.2 COMSTRUCT

Calculates mean phylogenetic distance (MPD) and mean nearest phylogenetic taxon distance (MNTD; aka MNND) for each sample, and compares them to MPD/MNTD values for randomly generated samples (null communities) or phylogenies.

This function accepts the switch `-a` to weight phylogenetic distances by taxa abundances. This changes the interpretation of MPD from the average distance among two random taxa chosen from the sample (default) to the average distance among two random individuals drawn from the sample (`-a` argument). Similarly, it changes the interpretation of MNTD from the average distance to closest relative for each taxon in the sample (default) to the average distance to closest non-conspecific relative for each individual in the sample (`-a` argument).

For each run, the samples or phylogeny are randomized using one of several null models (described below). The mean and standard deviation of MPD/MNTD for the randomly generated null communities are reported for each sample. The rank of observed MPD/MNTD values relative to the values in the null communities are reported as rankLow (number of null communities with MPD/MNTD values less than or equal to observed) and rankHi (number of null communities with MPD/MNTD values greater than or equal to observed). These ranks can be used to calculate P-values (e.g. for a one-tailed P-value, divide a rank by the $numberofruns + 1$). Note that if the sum of rankLow and rankHi for MPD or MNTD is not close to the number of runs, there must be a large number of ties between observed and null community values and results should be interpreted with caution. This situation may arise when using very small phylogenies or numbers of samples.

Two measures of 'standardized effect size' of phylogenetic community structure are calculated: the Net Relatedness Index (NRI) and Nearest Taxon Index (NTI) describe the difference between average phylogenetic distances in the observed and null communities, standardized by the standard deviation of phylogenetic distances in the null communities. NRI and NTI are calculated for each sample in a manner similar to that described in **?**:

$$NRI_{sample} = -1 \times \frac{MPD_{sample} - MPD_{rndsample}}{sd(MPD_{rndsample})}$$

$$NTI_{sample} = -1 \times \frac{MNTD_{sample} - MNTD_{rndsample}}{sd(MNTD_{rndsample})}$$

6.1.3 Null models

Choosing an appropriate null model and species pool to measure phylogenetic community structure requires careful consideration. Every null model makes different assumptions, and using two null models or different species pools to analyze the same data can give radically different results. See **?** or **?** for an evaluation of the assumptions and shortcomings of the different types of null models implemented in this software, and **?** for an example of these null models applied to ecological data.

Specify which null model to use with comstruct using the `-m` command line option plus the number correponding to one of the following null models:

`0` *Phylogeny shuffle*: This null model shuffles species labels across the entire phylogeny. This randomizes phylogenetic relationships among species.

1 *Species in each sample become random draws from sample pool*: This null model maintains the species richness of each sample, but the identities of the species occurring in each sample are randomized. For each sample, species are drawn without replacement from the list of all species actually occurring in at least one sample. Thus, species in the phylogeny that are not actually observed to occur in a sample will not be included in the null communities.

2 *Species in each sample become random draws from phylogeny pool*: This null model maintains the species richness of each sample, but the identities of the species occurring in each sample are randomized. For each sample, species are drawn without replacement from the list of all species in the phylogeny pool. All species in the phylogeny will have equal probability of being included in the null communities. By changing the phylogeny, different species pools can be simulated. For example, the phylogeny could include the species present in some larger region.

3 *Independent swap*: The independent swap algorithm (**?**); also known as 'SIM9' (**?**) creates swapped versions of the sample/species matrix. It constrains the swapped matrices to have the same row and column totals as the original matrix (i.e. number of species per sample and frequency of occurrence of each species across samples are held constant as species co-occurrences in samples are randomized). The algorithm searches the presence/absence matrix for 'checkerboard' cells (pairs of species/samples of the form `(0..1), (1..0)` or vice versa) and swaps these cell contents when it finds them. Number of swaps per run can be set at the command line with the `-w` argument. Number of swaps per run defaults to 1000, but please note that the number of swaps must be large relative to the number of occupied cells in the species/sample matrix to ensure the community is properly randomized. This null model can be very computationally demanding when dealing with large numbers of species or samples. Note also that this null model randomizes patterns of species co-occurrence in samples, but not abundances, and it does not introduce species from the phylogeny pool into the samples.

Functions incuding randomization test all accept several additional switches:

- `-r` X to set the number of runs (X) to randomize over. Can be zero. Otherwise the default value (999 runs) is used.
- `-a` to use abundance data in calculations. When this switch is used, all results reflect phylogenetic distances among individuals (abundance-weighted distances) as opposed to distances among taxa.

Examples include:

```
$ phylocom comstruct -m 0 -r 9999
$ phylocom comstruct -m 0 -r 9999 -a
$ phylocom comstruct -m 3 -w 100 -r 999
```

6.1.4 SWAP

Swaps the sample a number of times using the null model algorithm specified by the `-m #` option (described in the `comstruct` documentation) and outputs the resulting swapped matrix to console in the same format as the input file (each line contains `sampleId<TAB>abundance<TAB>species`). Note that the null model algorithms work with presence/absences in samples, not abundances, so all abundances are equal to 1 in the swapped matrix. For the independent swap, change the # of swaps at the command line with `-w`.

```
$ phylocom swap -m 2 -w 100 -r 999
$ phylocom swap -m 0 -r 9999
```

6.1.5 LTT *and* LTTR

For an ultrametric, time-calibrated phylogeny, the shape of the curve of number of extant lineages against time, from the root (one lineage) to the present (terminal lineage number), gives an indication of whether the tree contains many old lineages (a 'museum'), or just recent lineages (a 'cradle'). Within this species pool, the relative shape of such curves (analagous to 'lineage-through-time' plots, hence LTT) for sub-communities of samples gives an indication of their phylogenetic community structure. The LTT algorithm claculates these curves for a number of time slices. The LTTR algorithm calculates the rank of the number of the observed number of lineages against a null model of community membership, hence indicating if samples are phylogenetically even or clustered. Note that these analyses only 'make sense' for ultrametric trees.

6.1.6 NODESIG

Tests each node for overabundance of terminal taxa distal to it, so that the position of phylogenetic clumping/overdispersion in a community sample can be determined. Observed patterns for each sample are compared to those for random samples using null model 2 (random draws of s taxa from the phylogeny terminals where s is the number of taxa in a sample). A Nexus file is created with the input phylogeny reprinted once for every sample unit, with SIGMORE or SIGLESS appended (as a note) to each internal node with significantly more or less taxa than chance.

Open the resulting file in Mesquite. Make a new tree window. Select 'Show Notes on Tree' by command-clicking the hexagonal, note-tool symbol. Scroll through the trees, each of which corresponds to a different sample unit.

Note, if you have trouble finding files in Classic under OS X, you will have to open and resave the Newick file in a GUI text editor in OS X to add a resource fork to the file, needed for OS 9 software to 'see' it.

6.1.7 nodesigl

A text printout of nodal significance. For each sample and each node the number of dependent taxa is shown. The median of the random distribution is given, as is the rank of the observed. A mark is added to show nodes that have either more or less dependent taxa than expected.

6.2 Inter-sample phylogenetic distance

6.2.1 COMDIST *and* COMDISTNT

Outputs the phylogenetic distance between samples, based on phylogenetic distances of taxa in one sample to the taxa in the other.

COMDIST uses the mean pairwise distance (MPD)—for each taxon in a sample it finds the average distance to all taxa in the other sample, and calculates the mean. COMDISTNT uses the nearest taxon method (MNTD)—for each taxon in sample 1 it finds the nearest phylogenetic neighbor in sample 2, records this and calculates the mean.

Both functions accept the switch `-a` to weight phylogenetic distances by taxa abundances. This changes the interpretation of the measures from the average distance among a random taxon chosen from each of two samples (default) to the average distance among random individuals drawn from each of two samples (`-a` argument).

Use the output matrices from these functions in an ordination or clustering package (e.g., NMDS) for a 'phylordination' (**?**).

Measures of standardized effect size (β NRI and β NTI) can be calculated for these measures of inter-sample phylogenetic distance using the `-n` argument. Observed inter-sample MPD/MNTD are compared to the values expected under several null models (the same null models and arguments described in section 6.1.3 are available).

$$\beta NRI_{i,j} = -1 \times \frac{MPD_{observed} - MPD_{random}}{sd(MPD_{random})}$$

$$\beta NTI_{i,j} = -1 \times \frac{MNTD_{observed} - MNTD_{random}}{sd(MNTD_{random})}$$

The randomizations required to calculate these measures can be time consuming for large datasets, and are disabled by default. Use the `-n` argument to enable null model testing.

Examples include:

```
$ phylocom comdist
$ phylocom comdist -a
$ phylocom comdist -m 0 -r 999 -a -n
$ phylocom comdistnt -m 2 -r 999 -n
```

6.2.2 ICOMDIST

Outputs the phylogenetic distances between each taxon and all members of other samples. In the output, `AV` refers to mean distance, `NT` refers to mean nearest taxon distance.

6.2.3 RAO

Calculates Rao's quadratic entropy, a measure of community diversity weighted by phylogenetic distances among species (**?**). Calculations and notation follow **?**. The non-phylogenetic diversity metrics are equivalent to Simpson's diversity (the probability that two individuals from the community belong to different species), while the phylogenetic diversity metrics are interpretable as the expected phylogenetic distance between two randomly drawn individuals from different species. This method requires an ultrametric phylogeny (**?**). **?** has demonstrated that diversity components do not measure differentiation among vs. within communities properly, especially when communities have unequal weights (i.e. when communities contain unequal numbers of individuals).

Phylocom will issue a warning when your community data contain unequal numbers of individuals, in which case you should ignore calculations of alpha, beta and total diversity components.

Output sections and headings are as follows:

Diversity components

Reports overall alpha (within-site), beta (among-site), and total diversity, as well as the Fst statistic (beta / total) for diversity and phylogenetic diversity.

Within-community diversity

Plot Plot name

NSpp Number of species

NIndiv Number of individuals

PropIndiv Proportion of all individuals found in this plot

D Diversity (= Simpson's diversity)

Dp Phylogenetic diversity (= Diversity weighted by interspecific phylogenetic distances)

Among-community diversity

Among-community phylogenetic diversity

7 TRAIT-BASED COMMUNITY STRUCTURE: COMTRAIT

Calculate measures of trait dispersion within each community, and compare observed patterns to those expected under a null model. The comtrait function works exactly like the comstruct function, but instead of phylogenetic distances within each community, a measure of trait dispersion is calculated.

Several metrics of trait dispersion within communities can be calculated. Specify the metric to use with the `-x` switch plus is one of the following options:

`1` Variance: Variance of trait values

`2` MPD: Mean pairwise trait distance among taxa

`3` MNTD: Mean distance to nearest neighbor trait distance

`4` Trait range

Specify the number of randomizations with the `-r` switch, and specifiy the null model to use with comtrait using the `-m` switch plus the number correponding to one of the following null models:

`0` Trait shuffle: This null model shuffles trait values across species.

`1` Species in each sample become random draws from sample pool This null model maintains the species richness of each sample, but the identities of the species occurring in each sample are randomized. For each sample, species are drawn without replacement from the list of all species actually occurring in at least one sample.

`2` Species in each sample become random draws from traits data This null model maintains the species richness of each sample, but the identities of the species occurring in each sample are randomized. For each sample, species are drawn without replacement from the list of all species with trait values. This function is redundant since by definition the sample and trait species must match, but is included for consistency with the comstruct function.

`3` *Independent swap* Works as described in section 6.1.3.

Output column headings:

Trait Trait name

Sample Sample name

NTaxa Number of taxa in sample

Mean Mean value of trait in sample

Metric Observed metric in sample

MeanRndMetric Mean value of metric in null models

SDRndMetric Standard deviation of metric in null models

SESMetric Standardized effect size of metric

$$SES = \frac{Metric_{observed} - Metric_{random}}{sd(Metric_{random})}$$

rankLow Number of randomizations with metric lower than observed

rankHigh Number of randomizations with metric higher than observed

runs Number of randomizations

Usage:

```
$ ./phylocom comtrait -m 0 -x 2 -r 999
```

8 PHYLOMATIC

PHYLOMATIC is a tool for attaching members of a user-supplied list of taxa to a master, or 'mega' phylogeny at as terminal a position as possible, using the internal node names of the megatree. Please see **?** for more information on the goals of the tool.

PHYLOMATIC is also command line software; please see Section 4 for general information on using the command line. PHYLOMATIC requires two files: a `phylo` file in the same format as that used by PHYLOCOM, and a `taxa` file. The program is run by simply typing:

```
$ ./phylomatic
```

if all three files (`phylomatic`, `taxa`, `phylo`) are present in the directory, or, by specifying the appropriate data files:

```
$ ./phylomatic -f myphylo.new -t mytaxa.txt
```

Other switches include:

`-n` Label all nodes with default names

`-h` Help information

`-l` Convert all chars in `taxa` file to lowercase

8.1 *The `taxa` file*

Each [RETURN]-delimited line of the file lists a set of hierarchical taxon names (delimited by '`/`'), which will be sought for as either terminal or internal node names in the megatree. An example:

```
annonaceae/annona/Annona_cherimola
annonaceae/annona/Annona_muricata
fagaceae/Quercus_robur
dipterocarpaceae/shorea/Shorea_parvifolia
```

The last name on each line is the name that will be spliced into the returned tree. Note that current megatrees from the phylomatic2 project have lowercase internal names, and the `taxa` file should therefore also have lowercase names, unless the `-l` option is used. Note also that `phylomatic` will not match a taxon `Z` in `x/y/Z` where `Z` is an *internal* node in the megatree (reference) phylogeny. In the case where, for instance, an output tree of just genera is desired, but some of the genera appear in the megatree as internal node names, 'dummy species' names can be used genus (e.g., `betulaceae/alnus/alnus_sp`); a text editor can be used to remove the '`_sp`' from the output tree.

This '`/`'-delimited format allows the creation of unlimited user-defined phylogenetic structure. The program reads the string from right to left, matching the taxon at the first position it can in i) the 'megatree,' or ii) the growing user-defined tree. Hence, a `taxa` containing:

```
annonaceae/g1/s1
annonaceae/g2/s2
annonaceae/g2/s3
annonaceae/g2/s4/ssp1
annonaceae/g2/s4/ssp2
```

will produce a tree containing:

```
((s1)g1,(s2,s3,(ssp1,ssp2)s4)g2)annonaceae
```

See also the `phylomatic_example` directory in the distribution.

8.2 Branch lengths

PHYLOMATIC will use any branch lenghths in the input phylogeny to constrain the branch lengths of the output. However, if the input phylogeny has no branch lengths, PHYLOMATIC will attempt to create branch lengths to give a simple ultrametric tree. This algorithm sometimes fails, and a non-ultrametric tree is produced. These false branch lengths should be cleaned:

```
$ phylomatic > phylomatic_out.new
$ phylocom cleanphy -f phylomatic_out.new -e > clean.new
```

The cleaned tree can then be 're-ultrametricized' using PHYLOCOM BLADJ. Alternatively, the original input phylogeny to PHYLOMATIC can be ultrametricized using PHYLOCOM BLADJ.

9 ECOVOLVE

ECOVOLVE generates a phylogeny via a random birth and death process (output to screen), and generates a traits file (written directly to `ecovolve.traits`) with five randomly evolving, independent traits. A `sample` file (`ecovolve.sample`) is also written with a single sample unit ('alive') containing all extant members of the phylogeny.

Phylogeny growth is iterated over discrete time units until i) all lineages are extinct, ii) a maximum time-step has been reached, or iii) a maximum number of extant lineages has been generated. When speciation occurs, both daughters inherit the same value for a trait as the parent. Characters evolve via a pseudo-Brownian process: at each time step, a value for character change is drawn from a pre-assigned discrete probablility distribution (via the `-c` switch), plus or minus is assigned randomly, and the new trait value is calculated.

Ancestral competition can be simulated: if activated, the proximity in trait space (trait 1 only) of each lineage to the nearest lineage is calculated and proximity increases the probability of lineage extinction.

A number of switches determine behavior:

`-s` F (float): Probability of speciation per unit time. Default value = 0.05.

`-e` F (float): Probability of extinction per unit time. Default value = 0.01.

`-t` I (integer): Time units to simulate over. Default value = 100.

`-m` I (integer): Output mode (2 = LTT; 3 = newick). Default value = 3.

`-c` S (string of 10 integers): Probability envelope for character change. Default value = `3211000000`.

`-l`: Stop simulation after this number of extant lineages. Default value = 'no.'

-p: Output phylogeny pruned only for extant taxa. Default value = 'no.'

-d F (float): Taper character change by $e^{-time/F}$. This produces more conservatism in traits (see **?**).

-x: Simulate competition, with trait proximity increasing extinction. Default value = 'no.'

-h: Help.

The output can be viewed in Mesquite:

```
$ ecovolve > ecovolve.phylo
$ phylocom makenex -f ecovolve.phylo -t ecovolve.traits \
          -s ecovolve.sample > ecovolve.nex
```

Open `ecovolve.nex` in Mesquite, and trace characters (set BLs to be proportional).

10 TRAIT ANALYSES (BY DAVID ACKERLY)

The AOT module of PHYLOCOM conducts univariate and bivariate tests of phylogenetic signal and trait correlations, respectively, and node-level analyses of trait means and diversification. Phylogenetic signal is defined as the tendency for close relatives to resemble each other, and does not in itself indicate any particular process that may be responsible for the patterns of trait evolution (**?**). Trait correlations are tested using independent contrasts.

Like the rest of PHYLOCOM, the algorithms in AOT do not require branch lengths and can handle polytomies easily. Significance testing for the resulting patterns of trait conservatism is conducted by randomization of trait values across the tips of the phylogeny. Currently, AOT can assess phylogenetic signal for binary, ordered or continuous traits, and independent contrasts can be calculated between pairs of continuous traits, or continuous vs. binary traits. Multistate unordered traits cannot be included; one option for such traits is to code a set of $s-1$ binary dummy variables, where s is the number of states.

10.1 Trait means and variance by node: tip-based and node-based methods

Prior to examining phylogenetic signal and independent contrasts, AOT calculates two sets of statistics at each internal node of the tree, based on the average and standard deviation of trait values for all terminal taxa descended from a node, and the average and standard deviation of descendent values that are passed down the tree. Define:

N_i number of terminal taxa descended from node i

V_i number of child nodes descended from parent node i

$C_{i,j}$ character values for terminal taxa j descended from node i

$A_{i,j}$ trait values at child nodes j descended from node i

$b_{i,j}$ adjusted branch lengths subtending child nodes j descended from node i (see **?**, for the adjustment algorithm).

The following values are then calculated recursively, starting at the tips of the tree and working towards the root. Average trait value at node i based on all descendent terminal taxa:

$$T_i = \frac{\sum_{j=1}^{N_i} C_{i,j}}{N_i}$$

Standard deviation of trait values at all descendent terminal taxa:

$$S_i = sd(C_{i,j})$$

Trait value at node i based on values at next level higher nodes $1...j$:

$$A_i = \frac{\sum_{j=1}^{V_i} \frac{A_{i,j}}{b_{i,j}}}{\sum_{j=1}^{V_i} \frac{1}{b_{i,j}}}$$

Root mean square deviation of trait values at child nodes descended from node i, relative to value at parent node i (analogous to standard deviation, except that A_i at parent node is not equal to arithmetic mean due to weighting by branch lengths):

$$D_i = \left(\frac{\sum (A_{i,j} - A_i)^2}{V_i} \right)^{\frac{1}{2}}$$

The choice of A vs T as statistics to describe a node depends on the purposes of an analysis. The difference between them is in the relative weighting of terminal taxa. T weights each tip equally, regardless of the phylogenetic relationships between the focal node and the tips of the tree. A weights each child node that is descended from the focal node equally; as a result terminal taxa are weighted inversely in proportion to the diversity of successive descendent clades (Fig. 1). In the example below (with equal branch lengths), $A = 9$ at the root is the average of the two daughter nodes (7, 11) while $T = 8$ is the average of all terminal taxa ($64/8 = 8$). Similarly, the standard deviation of the terminal taxa (S) weights every tip equally, while the standard deviation of child node trait values (D) weights each descendent node equally.

The mean and standard deviation of the tip values (C) may be more appropriate for floristic and community studies, where the taxa in a particular clade are a biogeographic or ecological subset from the underlying phylogeny. The mean and variance of $A_{i,j}$ provide a more direct measure of evolutionary divergence and are appropriate for historical inference from data sets sampled from across entire clades. A values are also less sensitive to differences in sampling intensity across clades.

The significance of A, D, T, and S at each node are all assessed by randomization (see below). Significantly higher or low values of A and T indicate that the clade in question has lower or higher mean trait values that expected by chance. Significant results for D and S indicate low or high levels of diversification at a node (D) or overall diversification within a clade (S) (see **??**).

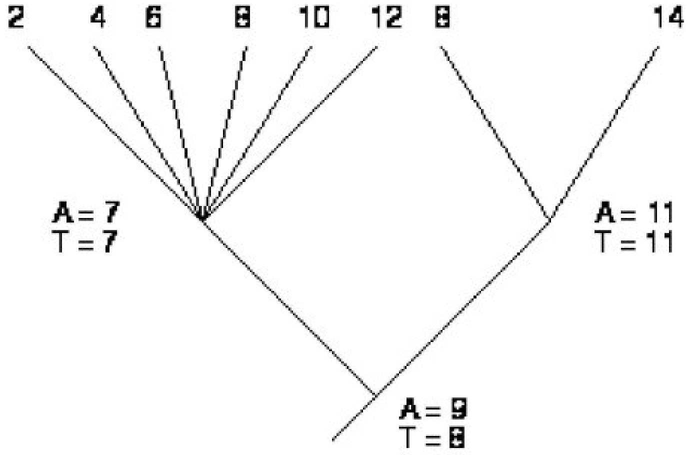

Figure 1 Illustration of calculation of weighted trait averages (A) and tip averages (T) on a phylogeny

10.2 The Contribution Index: Node-based partitioning of trait variance

? introduced new statistics to calculate the contribution of each divergence in a phylogeny to the overall variance in trait values across species. The following parameters are calculated to generate the 'Contribution Index' (see worked example in supplementary material to **?**. The following parameters are calculated and reported by PHYLOCOM, where the final three parameters correspond to Steps 2, 3 and 4 outlined in the **?** supplementary material.

NB: Previous users of AOT should note that the variables output by the program have been changed in v3.1 and onwards; output from previous versions did not exactly match the **?** *paper.*

SSTipsVNode total sum of squares for trait variance among species, relative to the nodal value for a clade (Ai). This is equivalent to the sums of squares of standard statistics, except deviations of each trait values are calculated relative to the nodal trait value at the base of each clade, rather than the grand mean. The result will necessarily be somewhat higher than the normal sums of squares.

SSAmongNodes sum of squares of the variance among nodal trait values of daughter nodes immediately subtending each node of the tree, weighted by the number of taxa in each node. This measures the among clade variance for the clades descendent from each node.

SSWithinNodes sum of squares of the variance among species trait values within each daughter clade descendent from a node. This is the sum of the individual SSTipsVNode values for those clades.

PercVarAmongNodes The percent of total trait variance in a clade that is explained by the divergence among daughter clades $= SSAmongNodes/(SSAmongNodes + SSWithinNodes)$.

PercVarAtNode The percent of total trait diversity that is contributed by the clade defined by this node $= SSTipsVNode/SSTipsVNode(root)$

ContributionIndex $PercVarAmongNodes \times PercVarAtNode$

10.3 Phylogenetic independent contrasts

AOT calculates evolutionary correlations between traits using independent contrasts. This method and its assumptions has been discussed extensively elsewhere and the user is advised to consult **??** for a thorough introduction. For pairs of continuous traits, AOT calculates standardized independent contrasts based on the branch lengths in the phylogeny (see below). Significance testing of the resulting correlation values can be done using tables of critical values for the Pearson correlation coefficient.

Independent contrasts are calculated from internal node averages (A values) of daughter nodes at each node. The direction of subtraction is set so that the sign of the contrast on trait 1 (X) is positive, and traits 2 and above (Y) are then compared in the same direction across the node. To handle polytomies, AOT uses the method introduced by **?** to obtain a single degree of freedom contrast at each polytomy, treating the polytomies as 'soft'. In this method, one trait must be designated as the independent or X variable, and all other traits are dependent or Y variables. The nodes arising from a polytomy are then ranked based on the values of trait X (see below for specifying which trait in a data set is set as X). The species are then split into two groups. For continuous traits, they are split at the median (if there are an odd number, the median value is assigned to the lower group if its value is lower than the mean for the entire set, and vice versa). For binary traits, they are split into groups by state. Then the mean values for all traits are calculated for the two groups, and the difference between these means is taken as a single contrast. This maximizes the difference in means for trait X, and the other traits fall out according to their distribution between the two groups.

AOT tests independent contrasts between continuous traits, or between continuous traits and a binary trait. Correlations between continuous traits are calculated over $N - 1$ internal traits. Contrasts on discrete traits can only be calculated on a limited set of nodes, where contrasting states of the binary trait occur on at least two of the descendent nodes (**?**; also see **?**). AOT identifies two sets of nodes for binary contrasts: the sister-taxa (ST) set and the paraphyletic (PT) set (Fig. 2). The ST contrasts involve nodes at which both binary states are observed in the daughter nodes, and no daughter clades contain a mix of the two states. Once a node is designated as ST, no deeper nodes on the path from that node to the root can be designated as an ST contrast. Some of these deeper nodes can be selected as PT contrasts, based on the rule that the paths connecting taxa that form a contrast cannot cross. PT contrasts are designated by pruning any node that has already been used as a contrast (ST or PT), and then continuing up the tree looking for bifurcations between the two binary states (Fig. 2). AOT reports results for contrasts calculated on the ST set alone, or on the combined ST+PT sets. I believe that the CAIC BRUNCH algorithm (**?**) also uses the combined ST+PT set of contrasts, but I have not confirmed this yet.

For pairs of continuous characters, AOT outputs the correlation of independent contrasts, the significance based on randomization, and the sample size (= number of internal nodes). For a continuous vs. a binary character, results are output for the ST and combined sets; for each one, the output consists of a paired t-test (the average magnitude of the continuous trait contrast for the available nodes and its significance), and a sign test (the number of positive contrasts for the continuous trait, the total number of contrasts, and the significance of the sign test).

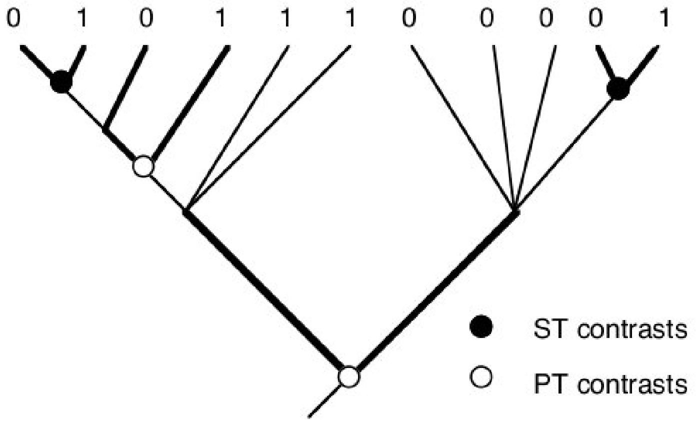

Figure 2 Illustration of nodes selected for ST and PT contrasts on a binary trait.

10.4 Branch lengths

The theory of independent contrasts relies heavily on the use of appropriate branch lengths on the phylogeny, in units of 'expected character evolution' (**?**). However, it is very difficult to justify the choice of BL distributions (e.g., based on molecular data) for analysis of ecological and morphological traits. Numerous simulation and empirical studies have shown that the results of most independent contrasts analyses are quite robust to different BL distributions (e.g., **??**). Several tests can be applied to test whether a particular data set fits the expectation of Brownian motion, as assumed by independent contrasts. One of these is to test for a lack of correlation between the absolute value of standardized contrasts and the underlying standardization term (square root of the sum of the subtending branch lengths, **?**). These terms are output by PHYLOCOM (`aotf` option) and can be imported and tested in any statistics package. The `Continuous` program (**?**) implements maximum likelihood tests of several different parameters, representing deviations from Brownian Motion; this program is also now implemented in `BayesTraits` (**?**). **?** discuss several other testable assumptions derived from Brownian motion.

The `-e` switch in PHYLOCOM will disable standardization of contrasts, and output unstandardized contrasts and their correlations. Note that this is not exactly the same as setting all branch lengths = 1. Under Felsenstein's algorithm, deeper branches are elongated during the calculation of contrasts, to reflect greater uncertainty at deeper nodes. This step can be applied to equal branch lengths, such that the deeper nodes will effectively have branch lengths slightly greater than 1. To use equal branch lengths with standardization in PHYLOCOM, branch lengths must be either deleted or set to 1 in the tree file.

10.5 Significance testing

Significance testing for all node-level and tree-wise conservatism measures is conducted by randomization of trait values across the tips of the phylogeny. Results of randomization are reported as the number of randomizations for which the statistic was less than or equal to the observed data (PL), and the number for which it was greater than or equal to the observation (PH), including the observation (see below). By default 999 randomizations are perfomed, plus the observed data for a total of 1,000. The number of randomizations can be set using the `-r` switch, to obtain higher precision in significance values; to obtain observed values only without significance testing, use `-r 0`.

Significance for one-tailed tests can be calculated as p = PL/R or p = PH/R, if testing for observations that are lower than expected or higher than expected, respectively. For two-tailed tests: p = minimum(2PL/R, 2PH/R).

For large sample sizes, the calculation of PL and PH separately is redundant, because PL+PH = R+1 (e.g., if the observation was lower than all 999 random replicates, PL = 1 and PH = 1000). However, for very small sample sizes, where randomization may result in many identical outcomes, PL and PH may sum to more than R+1. This occurs because all randomizations that result in values identical to the observation should be counted in both PL and PH (since significance is for randomizations less than or *equal to* the observation). PL and PH are both tabulated to address this situation.

10.5.1 Significance of independent contrasts

Randomization of tip values is not appropriate for testing independent contrasts, because it breaks down patterns of trait conservatism (see **?**). Significance of contrasts should be determined from appropriate parametric tests. For pairs of continuous traits, the correlation coefficient for independent contrasts can be tested for significance, using N-1 df (where N is the number of internal nodes providing contrasts) (**?**, Table R). For tests of a continuous trait vs. a discrete trait, the mean and standard deviation of of the contrasts can be used to conduct a one-sample t-test against the null hypothesis that the mean = 0. This is based on a t-statistic with N-1 df (where N is number of contrasts again) (see **?**, Table B), and is equivalent to a paired t-test conducted across the contrast nodes. (Note: currently PHYLOCOM only calculates unstandardized contrasts if trait 1 is binary. To obtain standardized contrasts, which may be more appropriate for the one-sample t-test approach, the analysis could be run once with `x 1` binary, to determine which nodes are identified as 'ST' and 'PT', and then run again with `x 1` set to continuous to obtain standardized contrasts for the remaining traits.) Finally, significance of a sign test can be tested against the binomial expectation (**?**, Table Q).

10.6 Phylogenetic signal

Phylogenetic signal is measured using a test based on the variance of standardized independent contrasts (**??**). If related species are similar to each other, the magnitude of independent contrasts will generally be similar across the tree, resulting in a small variance of contrast values. Observed contrast variances are compared to the expectations under a null model of randomly swapping trait values across the tips of the tree.

Note that in previous versions of Phylocom (<4.0), phylogenetic signal was calculated using a different method, based on the average magnitude of unstandardized independent contrasts over the tree. The current method will give different results than the previous method, especially since it is based on standardized contrasts which take branch length information into account. However, by taking branch lengths into account, the method provides a useful complement to other model-based measures of phylogenetic signal such as the *K* statistic of **?**. Phylogenetic signal calculations may result in NA values when the x variable is binary, since contrasts are only calculated for nodes at which there is a contrast in the binary trait. In this case, try setting the x variable to be continuous with the `-x` switch (see section 10.7.1).

10.7 Running trait analyses

Trait analyses can be run in three ways, each of which provides different output formats:

`aot` space-delimited output formatted for screen

`aotf` tab-delimited output for spreadsheet

`aotn` Nexus-formatted output for visualization in Mesquite

10.7.1 Switches

`-r INT` Modifies the number of randomizations (default = 999). E.g.,

```
$ phylocom aot -r 99
```

`-x INT` Specify which trait should be used as x variable for contrasts (default = 1). E.g.,

```
$ phylocom aot -x 2
```

`-e` Use equal branch lengths and unstandardized contrasts

To output results to a file, add a file name:

```
$ phylocom aotf > results.txt
```

Any number of switches can be combined in any order, e.g.:

```
$ phylocom aot -r 99 -e -x 3 > results.txt
```

10.8 Output format

The output has four different tables: individual node statistics; independent contrast values; tree-wide phylogenetic signal; correlations of independent contrasts. Option `aot` provides tables 3 and 4 only, formatted for viewing on the screen (space-delimited). Option `aotf` provides all four tables in a table-delimited table best viewed in a spreadsheet program. On a Mac, use the `open` command to open this file directly from the command line: `open -a Microsoft\ Excel results.txt`. Option `aotn` produces a Nexus-format file for opening in Mesquite.

10.8.1 Output table 1: Trait conservatism by node (`aotf` only)

Column headings:

trait trait number

trait.name trait name if provided in line 2 of the traits file; otherwise `trait_#`

node node number

name node name (last column in screen formatted output)

age node age, based on branch lengths in phylo file

N.tax number of terminal taxa descended from node

N.nodes number of daughter nodes descended from node (= 2 for bifurcating nodes)

Tip.mn Mean value of trait across all descendent terminal taxa (T)

Tmn.rankLow Significance of mean, rank in low tail of null distribution (sig in screen output)

Tmn.rankHi Significance of mean, rank in high tail of null distribution

Tip.sd Standard deviation of trait values across all descendent terminal taxa ('tips')

Tsd.rankLow Significance of standard deviation, rank in low tail of null distribution

Tsd.rankHi Significance of standard deviation, rank in high tail of null distribution

Node.mn Mean trait value calculated under ancestral averaging algorithm (A)

Nmn.rankLow Significance of ancestral mean value, rank in low tail of null distribution

Nmn.rankHi Significance of ancestral mean value, rank in high tail of null distribution

Nod.sd Standard deviation of values at daughter nodes ('divergences'); measure of trait radiation at this node

Nsd.rankLow Significance of divergence deviation, rank in low tail of null distribution

Nsd.rankHi Significance of divergence deviation, rank in high tail of null distribution

SSTipsRoot Standard sum of squares of trait variance for all species

SSTips Standard sum of squares of trait variance for species descended from this node

percVarAmongNodes Percent of variance in clade contributed by divergence at this node

percVarAtNode Percent of all trait variance contributed by species at this node

Contribution Contribution of divergence at this node to overall species level variation in

Index trait (see **?**)

SSTipVNodeRoot Sums of squares of trait deviations from nodal trait value at root

SSTipVNode Sums of squares of trait deviations from nodal trait value at this node

SSAmongNodes Sums of squares of deviations of daughter node trait values from nodal trait values at this node

SSWithinNodes Sums of squares of deviations of species trait values within subclades descended from this node

10.8.2 Output table 2: Independent contrasts by node (`aotf` only)

The LowVal and HiVal columns allow plotting of paired trait values in the original trait space that are then used to calculate contrasts. Column headings:

node node number

name node name (last column in screen formatted output)

age node age, based on branch lengths in phylo file

N.nodes number of daughter nodes descended from node (= 2 for bifurcating nodes)

Contrast1 Magnitude of independent contrasts for trait 1

Contrast2 Magnitude of independent contrasts for trait 2

... columns for any additional traits

If x is binary:

NodeType PT: paraphyletic contrasts; ST: sister-taxon contrasts. See discussion of binary traits above.

If x is continuous:

ContrastSD Standard deviation of the contrast (square root of sum of subtending branch lengths)

LowVal1 Lower node value used for calculating contrast 1

HiVal1 Higher node value used for calculating contrast 1

LowVal2 Node value for trait 2 at node with lower value for trait 1

HiVal2 Node value for trait 2 at node with higher value for trait 1

... node values for additional traits

10.8.3 Output table 3: Trait conservatism—treewise results

Column headers:

Trait Trait name (or number in screen output)

NTaxa Total number of taxa in tree

VarCn Variance of standardized contrasts across tree

VarCn.rankLow Significance of contrast variance relative to null model, low rank

VarCn.rankHi* Significance of contrast variance relative to null model, high rank

10.8.4 Output table 4: Independent contrast correlations

Results for traits 2 and higher vs. trait 1. If trait 1 is continuous, column headers are as follows (*: 'aotf' only):

Trait Number for trait Y ('aot' only)

XTrait* Name for trait X (same for all rows)

YTrait* Name for trait Y

NTaxa* Total number of taxa

PicR Correlation of independent contrasts

nPos Number of positive contrasts for sign test

nCont Number of contrasts tested

If trait 1 is binary, then the columns are:

Tr Trait number (`aot` only)

XTrait* Name for trait X (same for all rows)

YTrait* Name for trait Y

NTaxa* Total number of taxa

MnConAll Average magnitude of independent contrasts across all contrasts (subtracting node with lower value for binary character from node with higher value)

SdConAll Standard deviation of independent contrasts across all contrasts (for use in one-sample t-test)

nPosAll Number of positive contrasts for sign test, across all contrasts

nContAll Total number of contrasts tested (ST+PT)

(Next 4 columns repeat 4 above, for ST contrasts only)

11 AFTERWORD

Please feel free to read the code for details of what is done—it's heavily commented. If you have any comments or questions (even trivial ones), feel free to email us. If you think something is not working right, PLEASE get in touch. We have tried to assure that the algorithms are correct, but some errors are hard to spot.

There is much still to be added to PHYLOCOM (incorporating tree conversions, link character value analyses to community structure, etc.). If you have an algorithm you'd like implemented, let us know.

12 CITING PHYLOCOM

If you use PHYLOCOM, please cite the software in any resulting publications:

> Webb, C. O., Ackerly, D. D., and Kembel, S. W. 2008. Phylocom: software for the analysis of phylogenetic community structure and character evolution. Bioinformatics 24: 2098-2100.

13 ACKNOWLEDGMENTS

Cam: The original idea for these analyses came from the work of researchers in the 60's and 70's on species/genus ratios. David Baum suggested adding nearest-phylogenetic-neighbor metrics. Michael Donoghue and David Ackerly have both influenced my software ideas greatly. Phylocom development has been supported by grants 0212873 and 0515520 from the US NSF, by The Arnold Arboretum of Harvard University, and the Yale Institute of Biospheric Studies.

David: Thanks to Michael Donoghue for his inspiration in the pursuit of 'tree-thinking', to Mark Westoby and Angela Moles for assistance with the 'Contribution Index', and to Cam for enthusiasm and support in development of AOT. The implementation of the AOT module was facilitated by an NCEAS workshop on Neotropical forest ecology, co-organized with Susan Mazer and Miguel Martinez-Ramos.

Steve: Thanks to Mark Dale, whose collection of BASIC code first inspired me to learn scientific programming. Thanks to NSERC for support.

14 LEGAL

REFERENCES

Ackerly, D. D. 2000. Taxon sampling, correlated evolution, and independent contrasts. Evolution **54**:1480–1492.

Ackerly, D. D., and R. Nyffeler. 2004. Evolutionary diversification of continuous traits: phylogenetic tests and application to seed size in the California flora. Evolutionary Ecology **18**:249–272.

Blomberg, S. P., and T. Garland, Jr. 2002. Tempo and mode in evolution: phylogenetic inertia, adaptation and comparative methods. Journal of Evolutionary Biology **15**:899–910.

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.

Champely, S., and D. Chessel. 2002. Measuring biological diversity using Euclidean metrics. Environmental and Ecological Statistics **9**:167–177.

Diaz-Uriarte, R., and T. Garland, Jr. 1996. Testing hypotheses of correlated evolution using phylogenetically independent contrasts: sensitivity to deviations from Brownian motion. Systematic Biology **45**:27–47.

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

Felsenstein, J. 1985. Phylogenies and the comparative method. American Naturalist **125**:1–15.

Garland, T., Jr, P. H. Harvey, and A. R. Ives. 1992. Procedures for the analysis of comparative data using phylogenetically independent contrasts. Systematic Biology **41**:18–32.

Gotelli, N., and G. Entsminger. 2003. Swap algorithms in null model analysis. Ecology **84**:532–535.

Gotelli, N. J. 2000. Null model analysis of species co-occurrence patterns. Ecology **81**:2606–2621.

Gotelli, N. J., and G. R. Graves. 1996. Null Models in Ecology. Smithsonian Institution Press, Washington, DC.

Hardy, O. 2008. Testing the spatial phylogenetic structure of local communities: statistical performances of different null models and test statistics on a locally neutral community. Journal of Ecology **in press**.

Jost, L. 2007. Partitioning diversity into independent alpha and beta components. Ecology **88**:2427–2439.

Kembel, S. W., and S. P. Hubbell. 2006. The phylogenetic structure of a neotropical forest tree community. Ecology **87**:S86–S99.

Kraft, N. J. B., W. K. Cornwell, C. O. Webb, and D. D. Ackerly. 2007. Trait evolution, community assembly, and the phylogenetic structure of ecological communities. American Naturalist **170**:271–283.

Lapointe, F.-J., and T. Garland. 2001. A generalized permutation model for the analysis of cross-species data. Journal of Classification **18**:109–127.

Losos, J. B., and D. B. Miles. 2002. Testing the hypothesis that a clade has adaptively radiated: iguanid lizard clades as a case study. American Naturalist **160**:147–157.

Maddison, W. P. 2000. Testing character correlations using pairwise comparisons on a phylogeny. Journal of Theoretical Biology **202**:195–204.

Moles, A. T., D. D. Ackerly, C. O. Webb, J. C. Tweddle, J. B. Dickie, and M. Westoby. 2005. A brief history of seed size. Science **307**:576–580. URL http://dx.doi.org/10.1126/science.1104863.

Oliver, M., D. Petrov, D. Ackerly, P. Falkowski, and O. M. Schofield. 2007. The mode and tempo of genome size evolution in eukaryotes. Genome Research **17**:594–601.

Pagel, M. 1999. Inferring the historical patterns of biological evolution. Nature **401**:877–884.

Pagel, M., and A. Meade, 2006. BayesTraits. Internet. URL http://www.evolution.rdg.ac.uk/BayesTraits.html.

Pagel, M. D. 1992. A method for the analysis of comparative data. Journal of Theoretical Biology **156**:431–442.

Pavoine, S., S. Ollier, and D. Pontier. 2005. Measuring diversity from dissimilarities with Rao's quadratic entropy: are any dissimilarities suitable? Theoretical Population Biology **67**:231–239.

Purvis, A., and A. Rambaut. 2003. Comparative analysis by independent contrasts (CAIC): an Apple Macintosh application for analysing comparative data. Bioinformatics **11**:247–251.

Rao, R. 1982. Diversity and dissimilarity coefficients: a unified approach. Theoretical Population Biology **21**:24–43.

Rohlf, F. J., and R. R. Sokal. 1995. Statistical tables. Third edition. W. H. Freeman and Co., New York.

Strauss, S. Y., C. O. Webb, and N. Salamin. 2006. Exotic taxa less related to native species are more invasive. Proc Natl Acad Sci U S A **103**:5841–5845.

Webb, C. O. 2000. Exploring the phylogenetic structure of ecological communities: an example for rain forest trees. American Naturalist **156**:145–155. URL http://www.phylodiversity.net/cwebb/pubs/webb2000_an.pdf.

Webb, C. O., D. D. Ackerly, M. A. Mcpeek, and M. J. Donoghue. 2002. Phylogenies and community ecology. Annual Review of Ecology and Systematics **33**:475–505. URL http://www.phylodiversity.net/cwebb/pubs/webb2002_ares.pdf.

Webb, C. O., C. H. Cannon, and S. J. Davies, 2008. Ecological organization, biogeography, and the phylogenetic structure of tropical forest tree communities. *in* W. P. Carson and S. S. A. Schnitzer, editors. Tropical Forest Community Ecology. Blackwell, Oxford. URL http://www.phylodiversity.net/cwebb/pubs/webb_trf_comm_ecol_chapter.pdf.

Webb, C. O., and M. J. Donoghue. 2005. Phylomatic: tree assembly for applied phylogenetics. Molecular Ecology Notes **5**:181–183. URL http://www.phylodiversity.net/cwebb/pubs/webb2005_men.pdf.

Wikstrom, N., V. Savolainen, and M. W. Chase. 2001. Evolution of angiosperms: Calibrating the family tree. Proceedings of the Royal Society of London, Series B - Biological Sciences **268**:2211–2220.

A APPENDIX: WORKED EXAMPLES

Notation: # begins a comment line. `$` indicates a command that can be run at a unix command prompt.

A.1 Make a phylogeny for a plant species list

```
# Copy a recent megatree from phylomatic (e.g. using curl)
$ curl http://svn.phylodiversity.net/tot/megatrees/R20080401.new > \
  R20080401.new

# Create a species list in the appropriate format, called taxa
$ cat taxa
Annonaceae/Annona/Annona_cherimola
Annonaceae/Annona/Annona_muricata
Fagaceae/Quercus/Quercus_robur
...

# Run phylomatic
$ ./phylomatic -f R20080401.new
$ ./phylomatic -f R20080401.new > myphylo
$ cat myphylo
```

B APPENDIX: FAQS

B.1 Running PHYLOCOM

"I called my phylogeny file `phylo`*, but* PHYLOCOM *says it can't find the phylogeny file?"* (Windows users) By default, Windows prevents users from seeing full filenames, hiding the (usually three-character) suffix. This default behavior can be turned off in the 'Folder options' of the Control Panel, which we recommend strongly for serious computer users. If you created a `phylo` file in an editor, there is good chance it is actually called `phylo.txt`, which phylocom will not find.

B.2 AOT

"In the AOT module, I get output that contains 'NaN' and/or 'Inf'?" This is usually due to having resolved a polytomy as zero length branches. Standardized contrasts are divided by the sum of subtending branch lengths, and this will result in 'Inf' if it divides by zero. Another possible

explanation is that the x variable (set with the `-x` switch) is a binary trait. In this case, contrasts are only calculated at nodes for which there is a contrast in the x variable, which can lead to NaN values when there are relatively few nodes with contrasts calculated.

"I still don't really understand the tree-wide statistic that AOT calculates." AOT implements an algorithm that is discussed by **?** in *Evolution*. It is not the K statistic, but in the earlier part of the paper they talk about using the variance of standardized contrasts as a measure of signal, and comparing it to a null hypothesis of random shuffling of tip values across the tree. Note that by this measure Brownian motion will provide traits with a high degree of signal. This randomization approach cannot tell you if your traits are more or less conserved than expected under Brownian motion. For that you need a statistic such as K or another approach to randomization than we have implemented.

"How does AOT compare to CAIC?" If you are analyzing variation in a continuous trait Y relative to a discrete trait X, then phylocom implements the `BRUNCH` algorithm of `CAIC`. This will ignore the variance in Y observed across contrasts where X is constant.

B.3 BLADJ

"How can I run BLADJ *on a set of randomly resolved trees?"* Answer: with a bit of (unix) shell scripting. E.g.

```
#!/bin/sh
# rndres2bl.sh
# Takes a randomly resolved set of trees as input, matches to a tree
#   with interior node names, and then runs bladj.  Assumes that the
#   comnode match codes (match0...n) always occur in the same place.
# Input files:
#   1. infile.nex, a NEXUS file, with trees beginning, e.g.:
#      TREE rnd_resol1 = [&R] (((A,B),C),((D,E),F));
#   2. tree1, an unresolved tree to match against, e.g.:
#      ((A,B,C)cladeA,(D,E,F)cladeD);
#   3. ages, a bladj ages file, e.g.:
#     match0    100
#     match1    60
#     match2    30
# Creating the ages file will take a bit of visual assessment to
#   work out which of the matchX nodes can be dated.

grep "^TREE\ rnd" infile.nex | sed \
  ’s/^TREE\ *rnd_resol[0-9]*\ *=\ *\[\&R\]\ *//g’ > infile2.nex
N=0
echo -e "#NEXUS\nBEGIN TREES;" > outfile.nex
for tree in `grep ";" infile2.nex`
do
```

```
    echo $tree > tree2
    phylocom comnode > matched.nex
    grep "^TREE\ tree2" matched.nex | \
      sed ’s/TREE\ tree2\ =\ //g’ > phylo
    let N=$N+1
    echo -n "TREE rnd_resol"$N" = [&R] " >> outfile.nex
    phylocom bladj >> outfile.nex
  done
  echo "END;" >> outfile.nex
  rm -f infile2.nex tree2 matched.nex
```

B.4 new2nex and makenex

“I get this error when importing a .nex file into Mesquite: Taxon name in translation table doesn’t correspond to name of known taxon (”,” [a])” This appears to be an issue with some new versions of Mesquite, and might be fixed with postprocessing the nexfile using this command (thanks to Adam Wolf):

```
  $ sed ’/TRANSLATE/,/TREE/ s/, /   /g’ infile.nex > outfile.nex
```

B.5 Phylomatic

“Why won’t the megatrees (e.g., R20081027.new) open in TreeViewX?” The phylomatic database allows single daughter nodes in Newick: ((A,B)C)D which are also allowed by the Newick standard. TreeViewX can also handle these, but seems not to like it when the files are large. The single daughter nodes can be cleaned out using phylocom, e.g.:

```
  $ phylocom cleanphy -e -f R20081027.new > clean.new
```

The resulting file will open in TreeViewX. I have found that ATV can open the files directly.

Some other programs may also have trouble reading single daughter nodes, or the `:0.0000` branch length of the root node, or even any root node branch length at all, although these are all allowed by the semi-formal Newick standard. If you have trouble reading a phylocom or phylomatic tree into another program, please open the file in a text editor and consider the above issues.

PLEAE NOTE: Issues of compatability fo the root `(...)euphyllophyte;` node and other software applications are very common. If you are having trouble, please remove this node (edit out the `(` and the `)euphyllophyte;`) first. This can be automated with:

```
  sed -e ’s/(//1’ -e ’s/)euphyllophyte:1.000000//1’ -e ’s/)euphyllophyte//1’
```