

Branch support

Instructions for use

P. Hovenkamp

15-2-2005

Caution

All procedures to compute branch supports rely on either fairly large numbers (minimally, say 100) of replications of the complete analysis, or on extensive swapping around the optimal tree. They take time!

Bremer support

Bremer support (decay analysis) is the number of extra steps needed to "collapse" a branch. A collapsed branch is a branch that is not present in the entire set of allowed cladograms. Thus, when only optimal cladograms are allowed (all shortest cladograms), the branches that do not occur in the strict consensus are "collapsed", and can be said to have a Bremer support of 0. Usually, the term Bremer support is not used for these branches, and the branches are often not shown in the tree. However, sometimes some of these branches are displayed in a "majority rule" consensus. Then, by definition, the Bremer support for all branches with a frequency of < 100 in a majority consensus is 0.

If the set of all cladograms is extended to include not only the shortest ones, but also those with one extra step, the increased number of cladograms of course results in fewer branches common to all of them. Accordingly, more branches collapse if the strict consensus of all these cladograms is taken. Thus, a branch may be present in the consensus of all MPT's, while a search of all trees one step longer reveals a cladogram in which this branch is not present. Such a branch has a Bremer support of 1. Similarly, if all cladograms 2 steps longer are searched, even more branches will be collapsed. They have a Bremer support of 2, etc. Ultimately, if the search is expanded widely enough, all branches will collapse and the consensus of all cladograms will be fully unresolved.

There are two ways of computing Bremer support:

1. compute all cladograms for different degrees of suboptimality, and use the strict consensus of those to track at which level of suboptimality a clade "disappears".
2. compute all shortest cladograms that do *not* contain a particular branch, and find out the length of those ("reverse constraints" method).

In both methods, you need to repeat the analysis a number of times.

In method 1 you need to compute all cladograms of minimal length + 1, then all of minimal length +2, etc.;

In method 2 you need to repeat the analysis for each branch of the cladogram.

N.B. All methods of computing Bremer support are rather sensitive to the completeness of the search - if the fraction of all possible trees that are retained during the search is too small, the consensus will be overresolved, and Bremer support values will be too high. Thus it is *absolutely necessary* to set the number of trees to be held to a fairly high value (especially for the suboptimal trees!), with MAXTREES (Paup) or HOLD (Nona); and to do a fairly intensive search using a large number of replications.

How?

Paup

Using Paup, you can use AutoDecay (Macintosh only), which applies the reverse constraint method for each branch of a specified cladogram.

Alternatively, you can set the program (Macintosh and Windows) to keep all

trees of a particular length, and compute a strict consensus of those. There is no option to evaluate the resulting consensus trees – this must be done by hand.

Paup + AutoDecay (Macintosh)

In Paup:

1. Open the file containing a datamatrix
2. Analyze the matrix
3. Save the tree to a tree-file.

In AutoDecay:

1. Select the name of a file containing the tree-file you want to use
2. Give the length of the shortest cladogram you have found
3. Give the search parameters - *do not accept the default value for Nreps*, this value is too low for a sufficiently intensive tree search and therefore will artificially inflate the Bremer support values
4. Accept the name for the ADC-file that AutoDecay now produces

In Paup:

1. Run the ADC-file (**File - Open/Execute**)
2. Wait... until finished

Back to AutoDecay:

1. Select the name of the ADC-file
2. AutoDecay will now compute the Bremer support values and save them to a specified file.

It is possible to save the results to a tree-file that can be displayed by TreeView.

Paup only (Macintosh and Windows)

1. Open the file containing the datamatrix
2. Analyze the matrix
3. Compute the strict consensus
4. Note down the length of the MPT
5. Run the analysis again, this time with the parameter KEEP=length + 1
N.B. this value must be set to the total length of the tree, *not* to the number of extra steps.
6. Compute the strict consensus
7. Rerun step 5, each time increasing the KEEP-value, until the strict consensus is completely collapsed.
8. Compare all consensus-trees to obtain the Bremer support values.

Nona

In Nona, you can keep all trees up to a particular length, and then compute Bremer support values based on those trees.

1. Open the file containing the datamatrix
2. Enter the command
SUBOPTIMAL N, where N = the number of extra steps you want to allow
N.B. this is different from Paup, where N is the total length of the trees you want to allow.
3. Run the analysis

4. Give the command

BSUPPORT

If the displayed tree contains "overflow" values, it is necessary to increase the value for N and rerun the analysis.

Pablo Goloboff suggests the following procedure using the FIND command:

1. Open the file containing the datamatrix
2. Find the shortest tree (you can read in a tree found in an earlier search)
3. Give the commands:

OUT *FILENAME*

to make sure that the results are saved (otherwise, the supports will be displayed to the screen only)

4. Sequentially find longer trees with the commands:

HOLD 1000 ; SUB 1 ; FIND * ;

HOLD 2000 ; SUB 3 ; FIND * ;

HOLD 4000 ; SUB 5 ; FIND * ;

This sequential increase of search depth is to prevent the memory being filled with highly suboptimal trees before the less suboptimal trees are found. Depending on the size and quality of the dataset it may be necessary to increase the numbers for HOLD and SUBOPTIMAL.

5. Then compute Bremer supports:

BSUPPORT ;

TNT

Bremer supports can be computed with TNT in the same way as with Nona.

Interpretation of the results

Bremer support figures indicate the degree to which a clade is present in (slightly) less parsimonious trees. Thus:

- A support of 1 means that the clade is absent in at least 1 tree only one step longer than the most parsimonious tree.
- Similarly, a support of 2, 3, 4 etc. means that the clade is present in at least 1 tree 2, 3, or 4 steps longer.
- If the support for a clade is indicated as >4 this indicates that the clade was present in all trees 4 steps longer, in the case that trees more than 4 steps longer have not been examined.

What do these supports mean? A support of a particular value (say, 2) may mean either of two things:

- A branch is supported by two uncontradicted characters. Therefore, it can only "cost" two steps not to have that branch. *Bremer support has an upper bound in the number of characters on a branch!*
- A branch is supported by n characters, but contradicted by n-2. Therefore, to prefer the contradictory branch "costs" only 2 steps.

Standard Bremer support does not distinguish between the two possibilities. In Nona, you have the option to compute Relative Bremer support (command BSUPPORT *;), which does take into account the proportion of supporting and contradictory characters.

Note also that, as a result of the limitation by the number of characters, Bremer support values are not comparable across different datasets. In a dataset with very many characters (e.g., many molecular ones), Bremer support values may attain higher values than in datasets with fewer characters (as in most morphological ones). It would be unwise to interpret the lower values in the morphological cladogram as indicating a lower degree of reliability.

Bemer support values are not exact

The values that result of an analysis are usually only estimates of the true values. Why?

To find out whether a tree is absent in one tree of a particular length, *all* trees of that length must be examined. For most datasets, this is impossible. It is often already impossible to examine all trees of minimal length, let alone all trees one or more steps longer.

What does all this mean for the interpretation of the tree?

- It is not wise to draw strong conclusions from the presence of clades with a support of only 1.
- If any of the poorly supported clades appear to be interesting, the arguments to support these clades must come from the actual characters that support the clade. You must be prepared to say “Even if there is only one character supporting this clade, the character is so important that I tentatively accept this clades as well-supported”

Further reading:

Bremer, K., 1994. Branch support and tree stability. *Cladistics* 10: 295-304

Morgan, D.R., 1997. Decay analysis of large sets of phylogenetic data. *Taxon* 46: 509-517

Bootstrap

Bootstrap values are perhaps the most frequently used type of support values. They can be computed by resampling the characters randomly a number of times, and computing the frequency with which a node returns in the pooled results.

In some cases, bootstrap frequencies may have been computed on basis of the consensus trees for each replication - this will generally lead to lower frequencies. Most authors do not specify in which way bootstrap frequencies were obtained.

How?

Paup

For Paup, no helper programs are necessary. Computing bootstrap frequencies is a standard option.

Nona

Using Nona, bootstrap frequencies can be computed with the aid of the macro programming facilities. A file with macro instructions was posted some time ago to the Cladistic Software

(http://henry.ento.cornell.edu/CUIC/cladsoft_1st.htm) mailing list by Martín J. Ramírez. This file, BOOTFQ.RUN, computes bootstrap frequencies on basis of

all trees for the replications. A modified version, BOOT.RUN, is present in the Branch supports folder. This modified version uses a standard search protocol, saves only consensus trees for each replication and saves the tree for the unmodified data first.

Full instructions for using the Nona macro language are included with the registered version of Nona.

Bootfq.run

1. Using any editor, prepare a procedure file for Nona.
2. After the xread and ccode statement, include the following statement:
RUN BOOTFQ.RUN N %1 %2 ... %9;
with parameters as follows:
N - the number of bootstrap replications
%1 %2... - the Nona search options you want to use.

The bootstrap frequencies will be collected in a file BOOTFQ.TMP, which can be used as input for the program FQ:

1. Start Dos
2. Give the command:
FQ BOOTFQ.TMP or
FQ BOOTFQ.TMP OUTFILE

Boot.run

1. Using any editor, prepare a procedure file for Nona.
2. After the xread and ccode statement, include the following statement:
RUN BOOT.RUN OUTFILENAME;

Bootstrap frequencies based on 100 replications will now be collected in the specified outfile.

You can view these frequencies in the same way using FQ, using the command:

Fq = outfile

This will display the bootstrap frequencies for all nodes present in the original tree.

Winclada

Bootstrap support is implemented in Winclada.

TNT

A number of resampling supports is implemented in TNT, among which “standard” bootstrapping is one. For an explanation of the other measures, see:

Goloboff, P., J. Farris, M. Källersjö, B. Oxelmann, M. Ramirez, and C. Szumik. 2003. Improvements to resampling measures of group support. *Cladistics* 19:324-332

Interpretation of the results

Bootstrap frequencies can be **reported** in two ways:

1. As a “bootstrap majority rule” cladogram
Here, a tree is computed on basis fo the frequencies with which clades occur in the set of trees for all bootstrap replications. As with other majority rule trees, frequencies < 50% are by definition not present.

2. As support figures for the selected cladogram. Here, the clades in the original tree are taken as base, and their frequencies in the replicated trees are examined. Frequencies of < 50 % are now theoretically possible, but in practice a cut-off value of 50 % or sometimes 70 % is used.

There is no consensus about the significance of bootstrap frequencies. Theoretically, a cut-off value of 95 % can be defended with an analogy to the standard statistical analysis using a significance level of 5 %.

The use of a cut-off level of 70 % is often defended by reference to Hillis & Bull (1993), but these authors caution that “values cannot be directly compared among studies”.

In some recent papers, values of 80 % or more (approximately) are accepted as reliable, on basis of the predictive value of these values for the re-occurrence of the clades after taxon addition or deletion, but there are no data published that support this claim.

What happens when taxa are added or deleted can be more directly investigated using the Taxon Jackknife.

Taxon Jackknife

Taxon Jackknife support frequencies are the frequency of a particular node in all the cladograms produced by deleting one taxon in turn. This statistic was first proposed by Lanyon, and further promoted by Siddall, who included a helper program in his Random Cladistics program. As with bootstrapping, it may make a difference whether the frequency is based on consensus trees for all replications or whether all trees are used.

Taxon jackknifing is not a very frequently reported statistic, but should be, as it deals directly with the problems raised by the taxon sampling effect.

How?

Paup

No helper program is available.

Nona

A windowed interface to computing Taxon Jackknife values using either Nona or Hennig is provided in WinSupport.

WinSupport will read in nearly any type of file (Hennig, Nona, unformatted data or Nexus-files), and allows you to set a number of preferences as to how the data should be treated. It will then run Nona, and after the analysis collect the results to a logfile, as well as to a Nexus-format treefile which can be viewed with TreeView.

TNT

There is no option to compute Taxon Jackknife values in TNT, but it is probably possible to develop one using the Macro language. Currently, WinSupport does not support TNT as a search engine.

Carp

The Carp support (unpublished) measures the frequency with which a node returns if the data are changed minimally (1 step in 1 cell of the datamatrix). In

many cases, this will make no differences, but sometimes this reveals that support for a particular node is so weak that after making such small changes the node is often no longer present.

How?

Paup

No helper program is available.

Nona

In the program WinSupport there is an option to use matrix/Carp permutations to compute a frequency support value.

TNT

There is no option to compute Carp support values in TNT, but it is probably possible to develop one using the Macro language. Currently, WinSupport does not support TNT as a search engine.