Feedback — Quiz 4: Distance Matrix Methods (computer)

Help Center

You submitted this quiz on **Tue 16 Jul 2013 7:43 AM PDT**. You got a score of **16.00** out of **16.00**.

Overview

In today's exercise we will reconstruct phylogenetic trees using a variety of distance-based methods. Specifically, we will explore two different optimality criteria (least squares and minimum evolution), and one clustering method (neighbor joining).

Getting started

1. **Copy files for today's exercise:**

   Make sure you're still in today's working directory (condist) and that you already have the hcv.nexus file there. Now, copy the following file to the dir also:

   ```
   cp /home/student/data/simple.nexus simple.nexus
   ```

   ```
   ls -l
   ```

   simple.nexus is an artificial data set that I have constructed. It is identical to the one you analyzed by hand in Quiz 3. We will use it to convince ourselves that PAUP gets the same result as you.

Question 1

Analysis of the Simple Data Set

- **Start PAUP* and load the simple data set:**

```
paup simple.nexus
```

- **Select distance-based tree-reconstruction:**

```
set criterion=distance
```

- **Select uncorrected distances under the un-weighted least squares criterion:**

```
dset distance=p objective=lsfit power=0
```

The `dset` command is used to set various options for the distance-based methods. Option "`distance=p`" specifies the use of "uncorrected sequence distances", i.e., we do not want to correct the observed distances for multiple substitutions. Note that distances are here reported as "substitutions per site". This simply means that the number of differences has been divided by the length of the sequence. You can think of this distance as the fraction of sites that are different between two sequences.

The option "`objective=lsfit`" specifies that we want to reconstruct trees using the least squares optimality criterion. Recall that under least squares we are trying to find the tree that has the smallest possible deviation between the observed pairwise distances and the pairwise distances measured along the tree. (The distance between two taxa measured along the tree is called the "patristic" distance). The overall fit of the tree is found by (1) computing the difference between each observed distance and the corresponding patristic distance, (2) squaring this difference (this way we are sure to obtain a positive number, regardless of whether the observed or the patristic difference is larger), and (3) adding all the squared differences. The option "power=0" specifies that we do not want to weight the squared differences according to branch lengths when computing this fit.

- **Inspect distance matrix**

```
showdist
```

This command shows the distance matrix as evaluated under the current criteria.

- **Question:** Enter the pairwise distances, separated by spaces, in this order: BA, CA, DA, CB, DB, DC (that corresponds to reporting numbers top-to -bottom, one column at a time from the PAUP output, starting with the left-most column)

You entered:

```
0.60000  0.53333  0.13333 0.33333 0.60000  0.53333
```

Your Answer		Score	Explanation
0.60000	✔	0.17	
0.53333	✔	0.17	
0.13333	✔	0.17	
0.33333	✔	0.17	
0.60000	✔	0.17	
0.53333	✔	0.17	
Total		1.00 / 1.00	

Question 2

- **Find best tree using exhaustive search:**

```
alltrees
```

This data set is sufficiently small that we can search through all possible trees.

Q2: how many different, unrooted trees with 4 leafs is it possible to construct?

You entered:

3

Your Answer		Score	Explanation
3	✔	1.00	
Total		1.00 / 1.00	

Question 3

- **Inspect best tree:**

```
describetrees all/plot=phylogram brlens=yes label=yes
```

- **Question:** We now want to investigate whether the fitted branch lengths correspond to the observed pairwise distances. First, draw a sketch of the tree (note that in the PAUP output, this unrooted tree may look a bit weird - just draw it in the normal unrooted way you also used for the manual exercise, i.e., the tree should have a total of 5 branches). Second, label each branch with the branch length as listed in the table you just produced with `describetrees`. Finally, compute the patristic distance between each pair of species on the tree by adding up the branch lengths of branches lying on the path between the two taxa. Do the observed pairwise distances (from the distance matrix in the previous question) correspond to the patristic distances in this case?

Your Answer		Score	Explanation
Yes	✔	1.00	
No			
Total		1.00 / 1.00	

Question 4

- **Compare to manually constructed tree:**
- We now want to investigate whether the tree that PAUP has found here, corresponds to the one you constructed manually in Quiz 3. To do this you should convert all the fractional ("per-site") distances reported by PAUP, to absolute distances. This is done simply by multiplying the fractional distance by the length of the alignment (15 positions, in this case).
- **Question:** Is your tree and the PAUP tree identical?

Your Answer		Score	Explanation
No			
Yes	✔	1.00	
Total		1.00 / 1.00	

Question Explanation

The alignment has a length of 15 positions (have a look in the file simple.nexus). If the distance reported by PAUP was, say, 0.2, the corresponding absolute difference would therefore be: *[Math Processing Error]*

Question 5

Analysis of HCV Data Set using Neighbor Joining

- **Set up analysis for HCV data set**

  Still from within PAUP:

  ```
  execute hcv.nexus
  ```

  ```
  set criterion=distance
  ```

  ```
  dset distance=p objective=lsfit power=0
  ```

  ```
  outgroup  2_1_1 2_1_2 2_1_3 2_1_4 2_1_5 2_1_7 2_1_8 2_1_9 2_1_10
  ```

  ```
   set root=outgroup outroot=monophyl
  ```

  These commands will: load the file hcv.nexus (say yes when asked whether you want to reset the active datafile), select distance-based tree-reconstruction, select uncorrected distances, define patient 2 sequences as the outgroup, set outgroup rooting, and ensure outgroup is printed as monophyletic sister group to ingroup.

- **Construct a neighbor joining tree based on the HCV data:**

  ```
  nj
  ```

  This will construct a neighbor joining tree using the active distance measure (currently set to uncorrected).

- **Print tree and table of branch lengths:**

```
describetrees 1/plot=phylogram brlens=yes
```

The neighbor joining tree resembles the trees you previously constructed using parsimony. Importantly, you should see that the viral sequences from different patients form distinct clusters. Note that only a single tree is produced. This is characteristic of clustering methods, which work by following a deterministic algorithm for constructing a tree from distance data. Clustering algorithms such as neighbor joining do not have any measure of tree-goodness and therefore are not able to identify sets of equally good trees.

- **Question:** The present neighbor joining tree was computed without correcting the observed distances for multiple substitutions. In the phylogram, identify the internal node that is ancestral to the patient 5 sequences (you will see that internal nodes are labeled with consecutive numbers), and also the internal node that is one level further down in the tree (i.e., ancestral to the ancestral node). You will note that the branch connecting these two nodes is relatively long. Locate the branch in the list of branch lengths, which is printed above the tree. What is the length of this branch?

You entered:

0.10171

Your Answer		Score	Explanation
0.10171	✔	1.00	
Total		1.00 / 1.00	

Question Explanation

The internal node that is ancestral to the patient 5 sequences is node 74. The node ancestral to this node is node 75. The distance between node 74 and node 75 is found from the list of branch lengths, which is printed just above the tree. The length is 0.10171.

Question 6

- **Select correction of multiple substitutions using the Jukes and Cantor model:**

```
dset distance=jc
```

This causes all observed distances to be corrected using a formula based on the Jukes and Cantor model of evolution. Recall that under the Jukes and Cantor model all base frequencies are assumed to be equal (at 0.25), and all base substitution rates are also assumed to be equal.

- **Construct a new neighbor joining tree using corrected distances:**

```
nj
```

- **Print tree and table of branch lengths:**

```
describetrees 1/plot=phylogram brlens=yes
```

In this tree all branch lengths have been corrected for (unobserved) multiple substitutions. That means they are slightly longer than the uncorrected distances, and this correction is more noticeable for longer branches.

- **Question:** Again locate the internal node that is ancestral to the patient 5 sequences and also the immediate ancestor of this node (the node labels are not necessarily the same as before). Now find the corresponding branch in the table and make a note of the length. Is the corrected branch length longer than the uncorrected one?

Your Answer		Score	Explanation
Yes	✔	1.00	
No			

Total	1.00 / 1.00

Question 7

- **Question:** What is the ratio of the corrected to the uncorrected branch length? (Divide the corrected branch length by the uncorrected one)

You entered:

1.175

Your Answer		Score	Explanation
1.175	✔	1.00	
Total		1.00 / 1.00	

Question Explanation

If the corrected distance is longer than the uncorrected one, then the ratio is greater than 1. Otherwise it is less than 1.

Question 8

- **Prepare table of model fit measures**

You are currently using neighbor joining to reconstruct the phylogenetic tree. Below you will also explore the use of least squares and minimum evolution methods. In order to compare the performance and characteristics of these methods we want to record some informative numbers. On a piece of paper, construct a small table with two columns (labeled "SSE" and "tree length"), and three rows (labeled "NJ", "least

squares", and "minimum evolution").

- **Question:** At the end of the list of branch lengths (printed with the `describetrees` command), you will find the sum of all branch lengths. This is often called the "length" of the tree. What is the length of the tree? (also enter this number in your table, under the column "tree length" in the row "NJ")

You entered:

0.62279

Your Answer		Score	Explanation
0.62279	✔	1.00	
Total		1.00 / 1.00	

Question 9

- **Compute fit of NJ branch lengths to observed branch lengths:**

```
dscores 1/objective=lsfit power=0
```

The `dscores` command computes the fit of the tree in memory according to the specified criterion. In this case we are computing the fit between the observed branch lengths and the branch lengths found by neighbor joining. The measure used is the sum of squared deviations mentioned above.

- **Question:** What is the sum of squared errors? (it is indicated by "S.S" which is an abbreviation for sum of squares).

You entered:

0.02152

Your Answer		Score	Explanation
0.02152	✔	1.00	
Total		1.00 / 1.00	

Question 10

Analysis of HCV Data Set Using Least Squares

- **Select JC corrected distances under the unweighted least squares criterion:**

```
dset distance=jc objective=lsfit power=0
```

- **Find the best tree using heuristic searching:**

```
hsearch start=nj swap=tbr
```

As we have seen previously, the HCV data set is far too big for exhaustive searching, and we therefore have to resort to heuristic techniques when we are using a phylogenetic reconstruction method that is based on an optimality criterion. In this case the starting tree is constructed by neighbor joining, i.e., it should be identical to the tree we just inspected. The heuristic search (which again uses re-arrangements of the "tree-bisection and reconnection" type) should result in a small set of equally good trees.

- **Inspect trees:**

```
contree all/strict=no majrule=yes percent=50
```

This constructs a consensus tree from the set of equally good best trees. Again you should see that the set of best trees have individual patients clustered separately. Note that while the Neighbor Joining tree also showed this feature, it did not indicate that there might be any uncertainty as to the details of the tree. However, by using a method that has an explicit measure of tree goodness (least squares in this case) you have now learned that there are several equally good reconstructions of the branch order within the individual patient clusters.

- **Compute fit of least squares branch lengths to observed branch lengths:**

```
dscores 1/objective=lsfit power=0
```

Again, we are computing the sum of squared deviations between observed and patristic pairwise distances. Arbitrarily we have chosen to only do this for tree number 1 ("`dscores all`" would have done it for all trees in memory), but recall that all trees in memory are equally good, so the results would have been identical to what you now get.

- **Question:**What is the sum of squares? (Also enter the numbers in your table)

You entered:

0.02130

Your Answer		Score	Explanation
0.02130	✔	1.00	
Total		1.00 / 1.00	

Question 11

- **Find total length of tree:**

```
describetrees 1/plot=no brlens=yes
```

- **Question:** What is the sum of all branch lengths when using the least squares criterion? (Remember to also enter the numbers in your table).

You entered:

0.62177

Your Answer		Score	Explanation
0.62177	✔	1.00	
Total		1.00 / 1.00	

Question 12

- Now, compare the results from this analysis with the number you obtained from the neighbor joining tree above. Has the fit improved? (Recall that for both sum of squares and tree length, smaller is better).

Your Answer		Score	Explanation
○ The least squares tree has improved over the NJ tree only with respect to SSE.			
◉ The least squares tree has improved over the NJ tree both with respect to SS and tree length.	✔	1.00	
○ The least squares tree has improved over the NJ tree only with respect to tree length.			

The least squares tree has not improved over the NJ tree.

Total	1.00 / 1.00

Question 13

Analysis of HCV Data Set Using Minimum Evolution

- **Select JC corrected distances under the minimum evolution criterion:**

```
dset distance=jc objective=me
```

We now want to explore a different optimality criterion for distance-based analysis. Under minimum evolution we take the shortest tree to be the best one. This is very similar to parsimony, but in this case we are using pairwise, JC-corrected distances as the basis for reconstructing the tree. ME proceeds by searching through a list of possible trees; for each tested topology the best set of branch lengths are found by the least squares method, but instead of finally choosing the tree with the best fit, we instead end up by choosing the shortest tree.

- **Find the best tree using heuristic searching starting from a NJ tree:**

```
hsearch start=nj swap=tbr
```

- **Inspect trees:**

```
contree all/strict=no majrule=yes percent=50
```

Again you should see that the set of best trees have individual patients clustered separately.

- **Find total length of tree:**

```
describetrees 1/plot=no brlens=yes
```

- **Question:** At the end of the table listing branch lengths, you will again find the sum of all branch lengths. What is it?

You entered:

0.62148

Your Answer		Score	Explanation
0.62148	✔	1.00	
Total		1.00 / 1.00	

Question 14

Is the minimum evolution tree shorter than the other two trees?

Your Answer		Score	Explanation
○ The minimum evolution tree has the longest tree length.			
○ The minimum evolution tree is only shorter than the NJ tree.			
◉ Yes, the minimum evolution tree is shorter than both the least squares tree and the NJ tree.	✔	1.00	
○ The minimum evolution tree is only shorter than the least squares tree.			
Total		1.00 / 1.00	

Question 15

- **Compute fit of minimum evolution branch lengths to observed branch lengths:**

```
dscores 1/objective=lsfit power=0
```

- **Question:** Again, we are computing the sum of squared deviations between observed and patristic pairwise distances. What is the SS for minimum evolution? (Also note the result from this analysis in your table)

You entered:

0.02131

Your Answer		Score	Explanation
0.02131	✔	1.00	
Total		1.00 / 1.00	

Question 16

- **Compute fit of minimum evolution branch lengths to observed branch lengths:**

```
dscores 1/objective=lsfit power=0
```

- **Question:** Again, we are computing the sum of squared deviations between observed and patristic pairwise distances. Note the result from this analysis in your table and compare it with the numbers you obtained from the neighbor joining and least squares analyses above. How is the fit of the ME tree compared to those two **judged by the sum of squares**?

Your Answer		Score	Explanation
(•) The ME tree is better than the NJ tree but worse than the least squares tree	✔	1.00	
( ) The ME tree is better than the least squares tree but worse than the NJ tree			
( ) The ME tree is better than the other trees.			
( ) Both the least squares tree and the NJ tree are better than the ME tree.			
Total		1.00 / 1.00	

Question Explanation

Minimum evolution trees are explicitly optimized on tree length (shorter length is taken to be better - this is similar to the parsimony criterion). Least squares trees are explicitly optimized on the sum of squares (smaller is better). NJ is not optimized on anything, but can be seen as a heuristic (greedy) approximation of a minimum evolution tree.

From the analysis above we note that the performance of the three methods (neighbor joining (NJ), least squares and minimum evolution) is not too different both with respect to SSE and tree length for this data set. NJ always performs worse than the other methods because it does not have a measure tree goodness but the differences are fairly minor. This might not always be the case but will depend on the data. In general NJ will often give a good tree and it is often much faster than the other methods. Therefore, NJ is often a good first choice to get a general idea about the tree or as a starting point for further analysis.