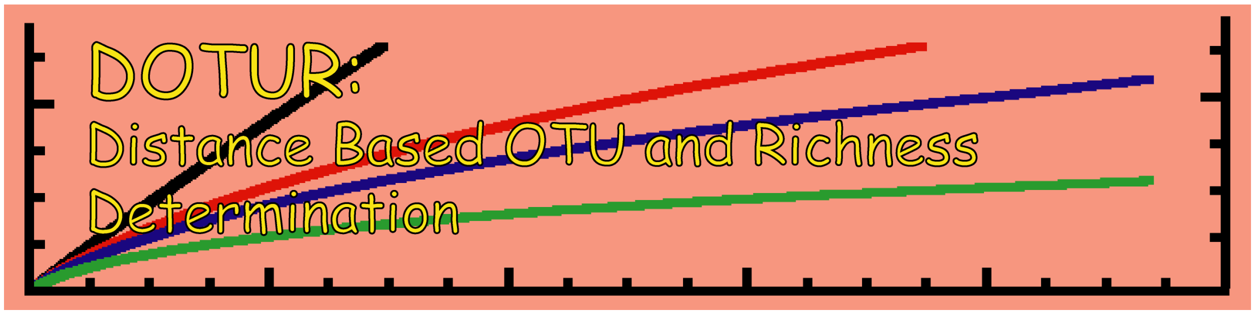


Introduction

DOTUR is a computer program that uses a distance matrix as the input file and assigns sequences to operational taxonomic units for every distance level that can be used to form OTUs using either the nearest, furthest, or average neighbor clustering algorithms. These are also called, single-linkage, complete-linkage, and UPGMA, respectively. Once sequences are assigned to OTUs, the frequency data for each distance level is used to construct rarefaction and collector's curves for the number of species observed, Shannon's and Simpson's diversity index, and Chao1, ACE, Jackknife, and Bootstrap richness estimators as a function of sampling effort and the distance used to define an OTU. DOTUR is freely available as C++ source code and as a windows executable.

This manual is designed to achieve five goals:

1. Describe the difference between each of the three sequence assignment algorithms
2. Show how to use DOTUR
3. Describe output files and equations used to calculate each parameter
4. Validate output by theory and making calculations by hand
5. Answer frequently asked questions

If you have any questions, complaints, or praise, please do not hesitate to contact Dr. Patrick D. Schloss at pds@plantpath.wisc.edu

Clustering Algorithms

Previous attempts at assigning sequences to OTUs have relied on gazing at a distance matrix and manually assigning sequences to OTUs, assigning OTUs based on BLAST results to the nt database, and using Forrest Rowher's program FASTGROUP. These methods have two main flaws. First, they are typically only used to obtain OTU data for one distance level. Second, they are somewhat arbitrary and prone to error (see our paper). For example, if you have three sequences A, B, and C. A is 2% different from B and C, but B and C are 3% different from each other. How do you define an OTU? Hmmm….

DOTUR, is a rapid, consistent, and objective way of assigning sequences to an OTU for all possible distance levels. It uses one of three rules: nearest neighbor, average neighbor, and furthest neighbor. Also, if you are interested in studying OTUs at cutoffs of 3, 5, 10 and 20% difference you will get these all in one execution of DOTUR (and everything in between as well!).

Here is how DOTUR can assign sequences to OTUs…

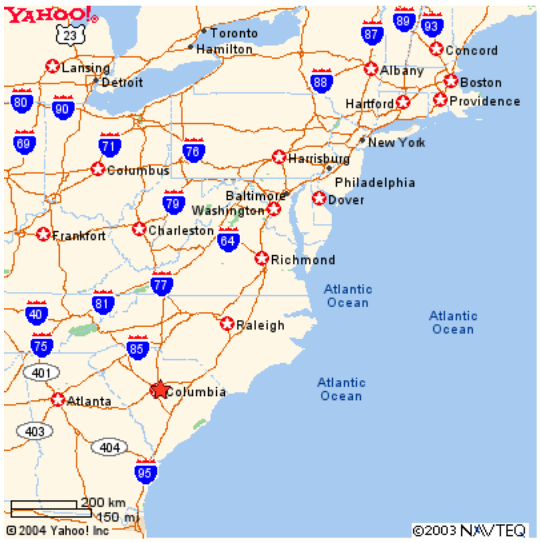
Nearest neighbor: Each of the sequences within an OTU are at most X% distant from the most similar sequence in the OTU.
Furthest neighbor: All of the sequences within an OTU are at most X% distant from all of the other sequences within the OTU.
Average neighbor: This method is a middle ground between the other two algorithms.

A Cartoon Example

The default is the furthest neighbor for reasons that will be explained below after considering a simplified example.

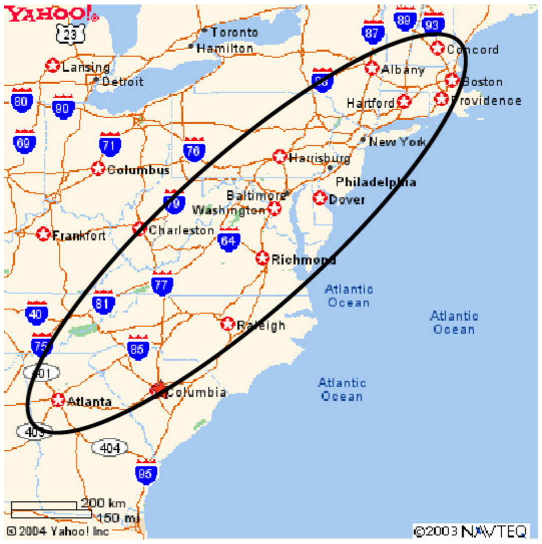
For the moment, let's assume that instead of being interested in bacterial 16S rRNA sequences, we are instead interested in finding ways to cluster major cities along the eastern seaboard of the United States. This analysis isn't meant to be precise, but to make several points about how we can assign sequences to OTUs. Perhaps we have a theory that all of the cities within a certain distance of each other share some cultural heritage. How will we cluster these?

In the map to the right, there are 15 major cities and state capitols.

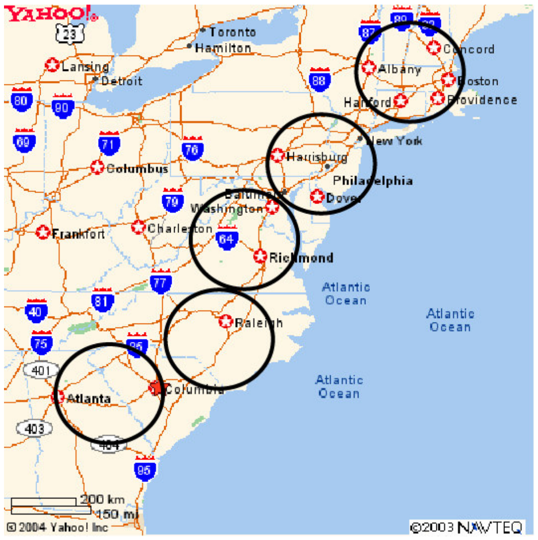


By the nearest neighbor method, you start by picking a city and looking for any city that is within 100 miles of it. So let's pick Concord. Boston is within 200 miles of it so Boston joins this group. However, Providence is within 100 miles of the nearest neighbor in this group. You can go on and on down the seaboard until you reach Atlanta, continually finding cities that are within 200 miles of the nearest neighbor in the group. Charleston, Frankfort, and Columbus would form their own OTU and Detroit and Lansing their own. But if your sampling intensity increased to include Toledo, OH and Blacksburg, VA, all of the US cities shown in the map would be in one group – hardly a regional grouping!

FASTGROUP would take the first city, Concord, and determine what cities are within 200 miles of it and lump all of those cities into the same group. Then it would take the next city, not in a group and search for all those cities not already in an OTU, but within 200 miles of it. This could result in decent assignment of cities to groups, or it might not. It all depends on what reference cities are picked.



If on the other hand we say that every city in a group is at most 200 miles from any other city in the group, then we are using the furthest neighbor approach. So, we see that Albany, Hartford, Concord, Boston, and Providence are all within the same group and Harrisburg, New York City, Dover, and Philadelphia are in separate group and so on. DOTUR starts by finding the minimum distance between any two sequences and putting those together in an OTU. As the distance is increased the original sequences remain in the OTU, but new sequences must meet the requirement that all they are within a known distance of all the other sequences in the OTU.



From this cartoon example, it is hopefully clear why we prefer the furthest neighbor approach to the nearest neighbor approach. In addition, FASTGROUP is a quasi-nearest neighbor/furthest-neighbor approach whose results are dependent on which sequence is picked as the reference sequence. The average neighbor approach would create groupings that are a compromise between the nearest and furthest neighbor approach. We prefer the furthest neighbor to the average neighbor because it is the more conservative approach and is not sensitive to the number of sequences sampled as the average neighbor would be.

DNA Distance Example
During the first step, DOTUR will produce three files: *.list, *.otu, and *.rank. The *.list file tells the user which sequences are in each OTU and the *.otu file tells the user how many sequences are in each OTU. The *.rank file tells the user how many OTUs contained a certain number of sequences. The format of these files will be addressed later in this manual.

Consider the following input file constructed from DNADIST (obtained from the DNADIST documentation):

```
5
A              0.000000  0.303893  0.857546  1.158921  1.542897
B              0.303893  0.000000  0.339731  0.913519  0.619666
C              0.857546  0.339731  0.000000  1.631729  1.293707
D              1.158921  0.913519  1.631729  0.000000  0.165882
E              1.542897  0.619666  1.293707  0.165882  0.000000
```

First, we'll consider the nearest neighbor algorithm. The smallest non-diagonal distance is 0.165882 between D and E. These two sequences enter into the first OTU at a distance of 0.165882. Since we want to retain the minimum distance, we retain the minimum values between the alpha and beta distances and we can rewrite the matrix as follows:

```
A              0.000000  0.303893  0.857546  1.158921
B              0.303893  0.000000  0.339731  0.619666
C              0.857546  0.339731  0.000000  1.293707
D,E            1.158921  0.619666  1.293707  0.000000
```

Now we have a doubleton containing D and E and three singletons. Incidentally, if we were interested in a definition where all sequences within an OTU were identical, then there were five singleton OTUs.

Continuing on, we again look for the smallest distance value and find that between A and B the distance is 0.303893. We repeat the clustering step and find:

```
A,B            0.000000  0.339731  0.619666
C              0.339731  0.000000  1.293707
D,E            0.619666  1.293707  0.000000
```

So at a distance of 0.303893 there are two doubleton OTUs and a singleton OTU. The next smallest distance is 0.339731 between the OTU containing A and B and the singleton OTU containing C:

```
A,B,C           0.000000  0.619666
D,E             0.619666  0.000000
```

Now we have a tripleton and a doubleton OTU when the distance between any two sequences in an OTU is at most 0.339731. Finally, there is one OTU when the shortest distance between any two sequences in an OTU is 0.619666.

The *.nn.otu file would look like this:

```
Unique      5     1     1     1     1     1
0.165882    4     2     1     1     1
0.303893    3     2     2     1
0.339731    2     3     2
0.619666    1     5
```

And the *.nn.list file would look like this:

```
Unique      5     A     B     C     D     E
0.165882    4     A     B     C     D,E
0.303893    3     A,B   C     D,E
0.339731    2     A,B,C D,E
0.619666    1     A,B,C,D,E
```

Finally, the *nn.rank file would look like this:

```
Unique      5     5
0.165882    4     3     2
0.303893    3     1     2
0.339731    2     0     1     1
0.619666    1     0     0     0     0     1
```

If we repeat the analysis for the furthest neighbor we again look for the smallest distance in the original distance matrix, but this time, we retain the maximum distance between the two sequences we are joining. So for a distance of 0.165882 between D and E, the following matrix would result:

```
A               0.000000  0.303893  0.857546  1.542897
B               0.303893  0.000000  0.339731  0.913519
C               0.857546  0.339731  0.000000  1.631729
D,E             1.542897  0.913519  1.631729  0.000000
```

The next smallest distance is between A and B at 0.303893 leading to the following matrix:

```
A,B             0.000000  0.857546  1.542897
C               0.857546  0.000000  1.631729
D,E             1.542897  1.631729  0.000000
```

In the next step, the smallest distance value (0.857546) is between the OTU containing A and B and the singleton OTU with only C in it:

```
A,B,C           0.000000  1.631729
D,E             1.631729  0.000000
```

Finally the five sequences join the same OTU with a distance of 1.631729 between them.

The resulting *.fn.otu, *.fn.list, and *.fn.rank files would look like:

```
Unique       5      1      1      1      1      1
0.165882     4      2      1      1      1
0.303893     3      2      2      1
0.857546     2      3      2
1.631729     1      5

Unique       5      A      B      C      D      E
0.165882     4      A      B      C      D,E
0.303893     3      A,B    C      D,E
0.857546     2      A,B,C  D,E
1.631729     1      A,B,C,D,E

Unique       5      5
0.165882     4      3      1
0.303893     3      1      2
0.857546     2      0      1      1
1.631729     1      0      0      0      0      1
```

The final method is the average neighbor approach. Instead of picking the lowest or highest distance between the two sequences being joined, it averages the distances in the two columns and rows begin combined. This is an unweighted average so it takes into account the number of sequences in each OTU when making the average. Again using the smallest distance to start the process, the D and E distances are merged:

```
A               0.000000  0.303893  0.857546  1.350909
B               0.303893  0.000000  0.339731  0.766593
C               0.857546  0.339731  0.000000  1.462718
D,E             1.350909  0.766593  1.462718  0.000000
```

The next smallest distance is 0.303893 between A and B:

```
A,B             0.000000  0.598639  1.058751
C               0.598639  0.000000  1.462718
D,E             1.058751  1.462718  0.000000
```

The next smallest distance is 0.598639 between the doubleton OTU containing A and B and the singleton OTU containing C. But remember that since this is an unweigthed approach, when we average the A,B and C columns and rows, we must multiply the A,B data by two and the C by one:

```
A,B,C           0.000000  1.193407
D,E             1.193407  0.000000
```

Finally, all five sequences join the same OTU at a distance of 1.193407. We would obtain *.an.otu, *.an.list, and *.an.rank files containing data like this:

```
Unique      5    1    1    1    1    1
0.165882    4    2    1    1    1
0.303893    3    2    2    1
0.598639    2    3    2
1.193407    1    5

Unique      5    A    B    C    D    E
0.165882    4    A    B    C    D,E
0.303893    3    A,B  C    D,E
0.598639    2    A,B,C D,E
1.193407    1    A,B,C,D,E

Unique      5    5
0.165882    4    3    1
0.303893    3    1    2
0.598639    2    0    2    3
1.193407    1    0    0    0    0    5
```

The difference between the *.fn.otu, *.an.otu, and *.nn.otu files in this test case is the distance definition of the OTU. In other applications, the clustering of sequences may be different between the two methods. But hopefully you will see how the in the nearest neighbor method all of the sequences within an OTU are at most 0.619666 from any sequence in the final OTU and in the furthest neighbor method all of the sequences in the OTU are at most 1.631729 from every member of the OTU and the average neighbor method is between the two. If you inspect the original distance matrix, this should make some sense.

You should also note that these examples are presented with 10^6 precision. If you can achieve this with 16S rRNA sequences (or any sequence!) let me know! One of the settings in DOTUR is the precision flag which allows you to set it to 10, 100, 1,000, or 10,000. As I alluded to, you are probably unlikely to really have precision up to 10,000 in the average gene sequence. Considering the average sequencing project only obtains about 500 bp from a gene, a more realistic level of precision would be between 100 and 1,000. The default precision is 100, which means that OTUs are reported every 0.01 unless you change the "-p" flag (see below). Because of DOTUR's use of the precision flag, a distance of 0.03 should not be interpreted as the same as 0.030 or 0.0300. The largest distance that could fall under 0.03 would be 0.0349. If you want the largest distance to be 0.03049, then set the precision to 1,000 and so on.

How to Run DOTUR

To compile DOTUR in LINUX type the following:

>g++ dotur.C –O4 –o dotur

DOTUR is run from the command line prompt. If you have a PHYLIP formatted "square" matrix then the following command will initiate the program assuming dotur (or dotur.exe) and amazon.dist (the distance matrix) are in the same folder:

```
>dotur amazon.dist
```

Many people may wish to use ARB to align sequences and make phylogenetic trees. Using ARB's neighbor joining method, you can create and export a distance matrix. ARB generates a "lower triangular" matrix. If you use a distance matrix constructed in ARB the following option must be set (-l, "el"):

```
>dotur.exe –l amazon.dist
```

The default setting is to use the furthest neighbor sequence assignment method the following flags will set it for the furthest neighbor, average neighbor, or nearest neighbor:

```
>dotur –c f amazon.dist              [this is the default]
>dotur –c a amazon.dist
>dotur –c n amazon.dist
```

Another default setting is to not calculate rarefaction curves for anything but the number of OTUs observed. Rarefying these parameters has no value and only confuses the issue of when to stop sampling. If you have any questions about this, please see the AEM paper or contact me. Because this substantially increases the length of the run, you must tell the program to calculate the estimator:

```
>dotur –r amazon.dist
```

The precision of the reported distances used to define an OTU is by 100ths (i.e. 0.01, 0.02, etc.). If you desire a greater level of precision, this can be achieved by setting the –p flag. However as the level of precision increases, so does the number of distances that curves must be constructed for, increasing the length of execution. Precision options are as follows:

```
>dotur –p 10 amazon.dist             [0.1]
>dotur –p 100 amazon.dist            [0.01, this is the default]
>dotur –p 1000 amazon.dist           [0.001]
>dotur –p 10000 amazon.dist          [0.0001]
```

For many applications, it is not necessary to calculate OTU groupings beyond a specific distance cutoff. For example if you were interested in the OTU groupings for distances less than or equal to 0.10, it would be worthless to calculate out to 0.50. This is possible in DOTUR using the "-stop X.XX" flag.

```
>dotur –stop 0.10 amazon.dist
```

The precision of the bootstrapping procedure can be altered by increasing or decreasing the number of iterations that the sampling without replacement procedure is run. The default value is 1,000 but can be changed as follows:

```
>dotur -i 10000 amazon.dist
```

Rarefaction curves of the number of OTUs observed is calculated by sampling without replacement. Some advocate obtaining 95% confidence intervals by actually sampling with replacement. This method of sampling can be performed in DOTUR using the "-wrep" flag. This results in the production of a file with the "*.wr_rarefaction" extension.

```
>dotur -wrep amazon.dist
```

Some may be interested in altering the input order of their sequences or they may be pooling data from multiple libraries that have no temporal collection context. The "-jumble" flag performs one ordering randomization and then carry's out the clustering and analysis.

```
>dotur -jumble amazon.dist
```

Although I advise against using a raw similarity score without correcting for multiple substitutions, by setting the "-sim" flag a similarity matrix can be used as input to dotur. Similarity is converted to a distance by subtracting the similarity score from 1.0.

```
>dotur -sim amazon.dist
```

Finally, execution in Windows and Linux (and Mac OSX) is essentially the same. In Windows, you cannot merely double click on the icon to get the program to execute. You must use the "Command Prompt" program found by going Start -> Program Files -> Accessories -> Command Prompt. Then you must type in the path of DOTUR and your distance file to execute the program:

```
C:\> "Documents and Settings\pds\Desktop\dotur.exe" "Documents and
Settings\pds\Desktop\amazon.dist"
```

Alternatively, you can change the root path to move to the desired directory and execute DOTUR from there:

```
C:\PATH\> dotur.exe amazon.dist
```

Be forewarned that DOTUR does not seem to run as quickly in Windows as it does in Linux and I would encourage everyone to align their sequences in ARB, which uses Linux or OSX, and to run DOTUR in the same operating system.

Output Files

DOTUR produces 23 output files. Although this may initially seem like a lot, many of the files are a representation of data found in other files. For example, each richness estimator and diversity index can have four separate files if the "-r" flag is set. All files are tab delimitated and are easily imported in to Excel or your favorite spreadsheet. The user is encouraged to track down the original papers to better understand how they were derived. Most reprints are available online (try www.jstor.org first).

.fn.* or *.nn.* or *.an.
If you give DOTUR the distance file amazon.dist, depending on the sequence assignment algorithm you used you will get files with the root "amazon.fn.*", "amazon.nn.*", or "amazon.an.*". These correspond to files constructed using the furthest, nearest, or average neighbor algorithms.

.otu and *.list
These files contain the number of sequences (*.otu) and their identity (*.list) in each OTU as a function of distance. In the *.otu file the first column contains the distance used to define an OTU, the second is the number of OTUs and the remaining columns tell the number of sequences in each OTU. The same information is contained in the *.list file except that instead of the number of sequences in each OTU DOTUR gives the name of each sequence in that OTU separated by commas.

.rank
This file contains data for constructing a rank-abundance plot of the OTU data for each distance level. The first column contains the distance and the second is the number of OTUs observed at that distance. The successive values in the row are the number of OTUs that were found once, twice, etc.

.r_* and *.c_
DOTUR constructs data files to plot collector's curves by default and the estimator and diversity rarefaction curves if the "-r" flag is set. If your distance matrix is sorted so that the sequences are in the order you sampled them, then the collector's curve files (*.c_*) have meaning. The collector's curve files are constructed assuming that the sequences are imputed in the order they were obtained. The rarefaction curve files (*.r_*) are randomizations of a sampling without replacement process. The first column in these files is the number of sequences sampled followed by three columns for each distance representing the mean parameter value and the lower and upper 95% confidence interval bounds for each parameter. For parameters where there is no easily defined 95% confidence interval "0.000"s are inserted.

.ltt
These are data files that can be used to construct "lineage through time" figures. Each estimator has a *.ltt file that contains the distance in the first column, the parameter for the full dataset in the second column, the lower 95% confidence interval bound in the third, and finally the upper 95% confidence interval bound. For some parameters (e.g. ACE) it is not possible to calculate the confidence interval so these are presented as "0.0000" in all rows. The parameter for the full

dataset is the value you would obtain using all sequences in the collection, which is the last row of the *.c_* or *.r_* file.

.collect and *.r_rarefaction
These are the collector's curve and rarefaction curve data for the number of observed OTUs as a function of distance between sequences and the number of sequences sampled. This is merely a count of the number of OTUs observed at any given point in the sampling process.

By theory, the rarefaction curve should match the following expression:

$$S_n = S_T - \frac{\sum_{i=1}^{S_T} \binom{N - N_i}{n}}{\binom{N}{n}}$$

where,

S_n = Average number of OTUs observed after drawing *n* individuals
S_T = Total number of OTUs in sample of *N* total individuals

Below is a comparison of the DOTUR output and the theoretical for a distance of 0.03 from the Amazonian dataset where the total number of sequences was 98, and there were 84 total OTUs with 75 singletons, 6 doubletons, 1 tripleton, and 2 quadrupletons. The DOTUR output was obtained using 10,000 random iterations and shows that the absolute error between the two approaches is very small. As the number of iterations decreases or increases, the percent error increases or decreases, respectively.

n	Theory	DOTUR	Diff.	%Error
1	1.000	1.000	0.000	0.000
2	1.996	1.995	0.001	0.049
3	2.987	2.986	0.001	0.024
4	3.974	3.971	0.003	0.069
5	4.956	4.954	0.002	0.045
6	5.935	5.934	0.001	0.010
7	6.909	6.909	0.000	0.002
8	7.880	7.881	-0.001	0.014
9	8.846	8.847	-0.001	0.013
10	9.808	9.809	-0.001	0.011
11	10.767	10.768	-0.001	0.012
12	11.721	11.721	0.000	0.001
13	12.672	12.675	-0.003	0.028
14	13.619	13.623	-0.004	0.031
15	14.562	14.566	-0.004	0.027
16	15.502	15.510	-0.008	0.051
17	16.438	16.446	-0.008	0.046
18	17.371	17.378	-0.007	0.040
19	18.300	18.305	-0.005	0.027

n	Theory	DOTUR	Diff.	%Error
50	45.620	45.634	-0.014	0.031
51	46.461	46.477	-0.016	0.035
52	47.299	47.317	-0.017	0.037
53	48.136	48.154	-0.019	0.038
54	48.970	48.995	-0.025	0.051
55	49.802	49.822	-0.019	0.038
56	50.633	50.647	-0.014	0.028
57	51.461	51.478	-0.016	0.032
58	52.288	52.305	-0.018	0.034
59	53.112	53.126	-0.014	0.027
60	53.935	53.946	-0.011	0.020
61	54.755	54.767	-0.012	0.022
62	55.574	55.586	-0.012	0.021
63	56.391	56.401	-0.009	0.017
64	57.206	57.213	-0.007	0.012
65	58.020	58.026	-0.006	0.010
66	58.832	58.840	-0.008	0.014
67	59.642	59.647	-0.005	0.009
68	60.450	60.448	0.002	0.003

20	19.225	19.233	-0.007	0.037
21	20.148	20.158	-0.010	0.049
22	21.066	21.079	-0.013	0.061
23	21.982	21.995	-0.012	0.057
24	22.894	22.904	-0.010	0.042
25	23.804	23.814	-0.010	0.042
26	24.710	24.719	-0.010	0.039
27	25.613	25.623	-0.011	0.041
28	26.512	26.521	-0.008	0.032
29	27.409	27.417	-0.008	0.029
30	28.303	28.313	-0.010	0.034
31	29.194	29.204	-0.010	0.033
32	30.082	30.093	-0.010	0.034
33	30.967	30.984	-0.016	0.052
34	31.850	31.863	-0.013	0.041
35	32.729	32.745	-0.015	0.047
36	33.606	33.618	-0.012	0.035
37	34.481	34.498	-0.017	0.050
38	35.352	35.371	-0.019	0.052
39	36.221	36.241	-0.020	0.055
40	37.088	37.109	-0.021	0.056
41	37.952	37.973	-0.022	0.057
42	38.813	38.835	-0.021	0.055
43	39.672	39.687	-0.014	0.036
44	40.529	40.542	-0.013	0.032
45	41.383	41.391	-0.008	0.020
46	42.235	42.245	-0.010	0.023
47	43.085	43.093	-0.009	0.020
48	43.932	43.945	-0.013	0.029
49	44.777	44.792	-0.015	0.034
69	61.256	61.260	-0.004	0.006
70	62.061	62.057	0.004	0.007
71	62.865	62.861	0.004	0.006
72	63.666	63.667	0.000	0.000
73	64.467	64.469	-0.003	0.004
74	65.265	65.273	-0.008	0.013
75	66.062	66.070	-0.008	0.012
76	66.857	66.862	-0.005	0.007
77	67.651	67.654	-0.002	0.004
78	68.444	68.448	-0.004	0.005
79	69.235	69.241	-0.007	0.010
80	70.024	70.035	-0.011	0.015
81	70.812	70.824	-0.011	0.016
82	71.599	71.612	-0.013	0.018
83	72.384	72.391	-0.007	0.010
84	73.168	73.175	-0.007	0.009
85	73.950	73.958	-0.008	0.010
86	74.731	74.746	-0.015	0.020
87	75.511	75.530	-0.019	0.026
88	76.289	76.302	-0.013	0.017
89	77.066	77.081	-0.015	0.020
90	77.842	77.856	-0.014	0.018
91	78.616	78.619	-0.003	0.004
92	79.389	79.391	-0.002	0.003
93	80.161	80.161	0.000	0.000
94	80.931	80.930	0.002	0.002
95	81.700	81.700	0.000	0.000
96	82.468	82.473	-0.005	0.006
97	83.235	83.237	-0.003	0.003
98	84.000	84.000	0.000	0.000

chao

These files give the full bias corrected Chao1 richness estimates as described by Chao (2, 4) and modified by Colwell (http://viceroy.eeb.uconn.edu/estimates).

$$S_{Chao1} = S_{obs} + \frac{n_1(n_1 - 1)}{2(n_2 + 1)}, \qquad \text{when } n_1>0 \text{ and } n_2 \geq 0 \text{ and when } n_1=0 \text{ and } n_2=0$$

$$S_{Chao1} = S_{obs} + \frac{n_1^2}{2n_2}, \qquad \text{when } n_1=0 \text{ and } n_2 \geq 0$$

where,

S_{Chao1} = Richness Estimate
S_{obs} = Observed number of species
n_1 = Number of OTUs with only one sequence
n_2 = Number of OTUs with only two sequences

To calculate the 95% confidence intervals we assume a lognormal distribution of the variance:

$$\mathrm{var}(S_{Chao1}) = \frac{n_1(n_1-1)}{2(n_2+1)} + \frac{n_1(2n_1-1)^2}{4(n_2+1)^2} + \frac{n_1^2 n_2(n_1-1)^2}{4(n_2+1)^4}, \qquad \text{when } n_1>0 \text{ and } n_2>0$$

$$\mathrm{var}(S_{Chao1}) = \frac{n_1(n_1-1)}{2} + \frac{n_1(2n_1-1)^2}{4} - \frac{n_1^4}{4S_{Chao1}}, \qquad \text{when } n_1>0 \text{ and } n_2=0$$

$$\mathrm{var}(S_{Chao1}) = S_{obs}\exp(-N/S_{obs})(1-\exp(-N/S_{obs})), \qquad \text{when } n_1=0 \text{ and } n_2>0$$

$$C = \exp\left(1.96\sqrt{\ln\left(1+\frac{\mathrm{var}(S_{Chao1})}{(S_{Chao1}-S_{obs})^2}\right)}\right)$$

$$LCI_{95\%} = S_{obs} + \frac{S_{Chao1}-S_{obs}}{C}$$

$$UCI_{95\%} = S_{obs} + C(S_{Chao1}-S_{obs})$$

where,

LCI = Lower bound of confidence interval
UCI = Upper bound of confidence interval

Revisiting the Amazon dataset The S_{obs} was 84, n_1 was 75 and n_2 was 6 so S_{Chao1} would be 481.19 OTUs. With a G of 12.5 the variance of the estimate is 48,808.59 and the C value would be 2.77. This gives a lower bound to the confidence interval of 227.6 and an upper bound of 1,182.9. The total range of the 95% confidence interval is 955.3.

ace

These files give the ACE richness estimates as described by Chao and her colleagues (3, 4). DOTUR calculates the 95% confidence interval using an algorithm for the standard error estimate obtained through a personal communication with Anne Chao.

$$N_{rare} = \sum_{i=1}^{10} i n_i$$

$$C_{ACE} = 1 - \frac{n_1}{N_{rare}}$$

$$\gamma_{ACE}^2 = \max\left[\frac{S_{rare}}{C_{ACE}}\frac{\sum_{i=1}^{10} i(i-1)n_i}{N_{rare}(N_{rare}-1)} - 1, 0\right]$$

$$S_{ACE} = S_{abund} + \frac{S_{rare}}{C_{ACE}} + \frac{n_1}{C_{ACE}}\gamma_{ACE}^2$$

$$\mathrm{var}(S_{ACE}) \approx \sum_{j=1}^{n}\sum_{i=1}^{n} \frac{\partial S_{ACE}}{\partial n_i}\frac{\partial S_{ACE}}{\partial n_j}$$

$$\mathrm{cov}(f_i, f_j) = f_i(1 - f_i / S_{ACE}), \textbf{if } \mathbf{i = j}$$
$$\mathrm{cov}(f_i, f_j) = -f_i f_j / S_{ACE}, \textbf{if } \mathbf{i \neq j}$$

where,

n_i = The number of OTUs with i individuals
S_{rare} = The number of OTUs with 10 or fewer individuals
S_{abund} = The number of OTUs with more than 10 individuals

Returning to the Amazonian dataset, with the previously described distribution, there are no "abundant" OTUs so S_{abund} is 0 and S_{rare} and S_{obs} are 84 and N_{rare} is 98. Since there are 75 singletons, the coverage, C_{ACE}, is 0.235. Calculating the γ^2 value we obtain the value 0.581. This gives an S_{ACE} value of 543.69.

boot
These files give the bootstrap estimate as described by Smith and Van Belle (5) but implemented for a single "quadrant".

$$S_{Bootstrap} = S_{obs} + \sum_{i=1}^{S_{obs}} \left(1 - \frac{S_i}{N}\right)^N$$

where,

N = The number of individuals sampled
S_i = The number of sequences in the ith OTU

For the Amazonian dataset $S_{Bootstrap}$=112.33 and there is not a simple expression for the 95% confidence interval

jack
These files give the interpolated Jackknife estimate as describe by Burnham and Overton (1). Since this is a complicated calculation, it makes DOTUR run much longer. Therefore, if you want the rarefied interpolated Jackknife estimate calculated, you must tell DOTUR to do so by using the "-j" flag at the command line. If it is not calculated, DOTUR will still produce the *.r_jack* files, but they will be filled with zeros.

$$S_{jack,k} = S_{obs} + \sum_{i=1}^{k} (-1)^{i+1} \binom{k}{i} n_i$$

$$\mathrm{var}(S_{jack,k}) = \sum_{i=1}^{n_t} (a_{ik})^2 n_i - S_{jack,k}$$

$$a_{ik} = \left\langle \begin{array}{l} (-1)^{i+1}\binom{k}{i} + 1, i = 1 \ldots k \\ 1, i > k \end{array} \right.$$

where,

k = The order of the Jackknife estimate

n_t = The number of sequences in the largest OTU

To determine which order of the estimate to use it is necessary to calculate the test statistics, T_k:

$$T_k = \frac{S_{jack,k+1} - S_{jack,k}}{\left(var\left(S_{jack,k+1} - S_{jack,k} \mid S\right)\right)^2}$$

$$var\left(S_{jack,k+1} - S_{jack,k} \mid S\right) = \frac{S_{obs}}{S_{obs} - 1}\left[\sum_{i=1}^{n_t}\left(b_i^2 n_i\right) - \frac{\left(S_{jack,k+1} - S_{jack,k}\right)^2}{S_{obs}}\right]$$

where,

$b_i = a_{i,k+1} - a_{i,k}$

For each T_k value, calculate its two-sided p-value. Find the first k-value where $P_k > 0.05$ and calculate c and d:

$$c = \frac{0.05 - P_{k-1}}{P_k - P_{k-1}}$$

$$d_i = ca_{i,k} + (1 - c)a_{i,k-1}$$

With c and d, calculate the interpolated S_{jack} and its standard error:

$$S_{jack} = \sum_{i=1}^{n_t} d_i n_i$$

$$se\left(S_{jack}\right) = \left(\sum_{i=1}^{n_t}\left(d_i^2 n_i\right) - S_{jack}\right)^{0.5}$$

For the Amazonian dataset, you can calculate the following:

k	**$S_{j,k}$**	**var**	**T_k**	**P_k**
1	159	150	13.91	<0.0001
2	228	450	8.89	<0.0001
3	292	938	5.77	<0.0001
4	**350**	**1700**	**3.36**	**0.0008**
5	**399**	**2940**	**1.54**	**0.1235**
6	434	5250		

The p-value crosses 0.05 between a order of 4 and 5 and you can calculate a c-value of 0.40 and the interpolated S_{jack} of 369.64 with 95% confidence interval between 278.98 and 460.30. Note that programs like EstiamteS and various microbial ecology papers present either the first and/or second order Jackknife estimate. This method essentially uses a statistical procedure to determine which order results in the minimum bias (error).

shannon
These files give the classic Shannon-Weaver Index of diversity.

$$H_{\text{Shannon}} = -\sum_{i=1}^{S_{obs}} \frac{S_i}{N} \ln \frac{S_i}{N}$$

For the Amazonian dataset the Shannon Index is 4.35.

To obtain the 95% confidence interval we assume that the variance is normally distributed and can be calculated as

$$\text{var}\left(H_{\text{Shannon}}\right) = \frac{\sum_{i=1}^{S_{obs}} \frac{S_i}{N}\left(\ln \frac{S_i}{N}\right)^2 - H_{\text{Shannon}}^2}{N} + \frac{S_{obs} - 1}{2N^2}$$

The variance is 0.0020. This gives a lower and upper bound to the 95% confidence interval of 4.26 and 4.44.

simpson
These files give the classic Simpson Index of diversity.

$$H_{\text{Simpson}} = \frac{\sum_{i=1}^{Sobs} S_i\left(S_i - 1\right)}{N\left(N-1\right)}$$

For the Amazonian dataset, the Simpson Index is 0.0044.

Frequently Asked Questions

How do I cite DOTUR?
Schloss, P.D. & Handelsman, J. 2005. Introducing DOTUR, a computer program for defining operational taxonomic units and estimating species richness. Applied and Environmental Microbiology. **71**(3):1501-1506.

***When using Windows, I get an error that says "Error: could not open…"* What is going on?**
This generally happens because people do not have their input file in the correct folder. For ease of use, put both dotur.exe and your input file in the same folder. If you think you have done this and still get the error, type "dir" at the prompt and make sure you can see both files – you probably cannot.

In windows, why does the command window open and close quickly when I double click on the DOTUR icon?
This is because you haven't given DOTUR an input file and you will get an error message quickly followed by the screen closing. Please see the above section on how to run DOTUR.

I use XYZ program to construct distance matrices. Can I use DOTUR?
We are anxious to help people use DOTUR in a way that is easiest for them. Please contact me (pds@plantpath.wisc.edu) with an example distance matrix, and I will incorporate the format into DOTUR.

Why doesn't DOTUR calculate XYZ parameter?
There are probably many parameters not included in DOTUR. If you would like to see one incorporated, please contact me (pds@plantpath.wisc.edu).

Literature Cited

1. **Burnham, K. P., and W. S. Overton.** 1979. Robust estimation of population size when capture probabilities vary among animals. Ecology **60:**927-936.
2. **Chao, A.** 1984. Non-parametric estimation of the number of classes in a population. Scand. J. Stat. **11:**265-270.
3. **Chao, A., and S. M. Lee.** 1992. Estimating the number of classes via sample coverage. J Am Stat Assoc **87:**210-217.
4. **Chao, A., M. C. Ma, and M. C. K. Yang.** 1993. Stopping rules and estimation for recapture debugging with unequal failure rates. Biometrika **80:**193-201.
5. **Smith, E. P., and G. van Belle.** 1984. Nonparametric estimation of species richness. Biometrics **40:**119-129.