**labellllllHow to make genetic trees**

Download the free **MEGA** (Molecular Evolutionary Genetics Analysis) program for PC from [www.megasoftware.net](http://www.megasoftware.net). You probably want the Windows Graphical Mega 11 (64 bit) version (the latest as of this writing). If there has been an update since I wrote this, the directions might need altering a bit.

Thanks to *Phylogenetic Trees Made Easy* by Barry Hall which provided me some helpful insights.

**1. Chromatogram (AB1) and FASTA files**

**Chromatograms**

When you get a DNA sequence back from the lab, they may give you a pair of **chromatogram** files with the extension .AB1 (because Applied Biosystems built some of the sequencing machines). These are the forward and reverse reads.

Somebody has to interpret that picture and create the DNA sequence. Hopefully the lab did that for you, and also gave you a **FASTA file** (perhaps with the extension .FAS or .FASTA) (but they probably charged you at least a dollar to a few dollars more if they did). If so, great, use that FASTA file. But if you ever find that sequence is suspicious, and need to double check that it is a healthy sequence, you may need to check the original chromatogram, by dragging the .AB1 file onto the MEGA program.

If you were not given a FASTA file, you will have to create one yourself out of the chromatogram.

Even if you have a FASTA file, you might want to double check it and correct errors in it.

Graphical user interface, application

Description automatically generated

A **healthy chromatogram** has a clear winner for which colour belongs in which place. In this case, I trust that the sequence was correctly read from locations 25 to 110.

Graphical user interface

Description automatically generated

An **unhealthy chromatogram** has many locations where there is more than one colour peak. Something went wrong. Perhaps one of the two alleles of the gene had an indel. Perhaps there was contamination in the vial and more than one species of mushroom got inside. Who knows. In this case, I do not trust this sequence from location 110 to 200.

Maybe you’ll get lucky and the tallest peaks were the correct peaks and the DNA sequence is still good. That happens. But if you have reason to mistrust it (see section 3 bullet point 9 of my “Analyzing DNA sequences” document) take a look at the chromatogram (if you have it). Unfortunately, GenBank sequences do not provide you chromatograms so you are out of luck unless you know the person who had the sequence made.

**Trim the sequence**

The **beginning and end of the chromatogram are always corrupt** and have to be trimmed. (There may be a bug in MEGA where you have to maximize the chromatogram window and then restore it before you can scroll the window horizontally).

Graphical user interface

Description automatically generated

In this case, the first 18 characters are unclear, and show up as “N” (unknown). We have to correct those that we can and erase the rest. In this case, reading backwards from 18, I’m pretty sure 18 is a T.

But then, is there one G or two of them? That black peak is a little wider than the peaks further on. But the horizontal size of the peaks is not linear near the ends of the sequence, so you expect the peaks to get wider at the ends (many of the early peaks are wide). Look at the pair of G’s at position 29 and 30, there’s a bump showing you that there are really 2 black peaks. Position 17’s black peak kind of has a little shoulder on the left side. Does that mean it’s two G’s in a row? I think so. But I can’t be sure. This is why in section 3 bullet point 7 of my “Analyzing DNA sequences” document I caution you against believing that a discrepancy between 1 or 2 of the same letter at one extreme end of a sequence is a real difference.

Continuing to read backwards we have C, T, G, C and now I’m going to stop because I don’t know how many T’s that wide red peak is, nor any clue what comes before that.

So I decide to erase the beginning characters and start my sequence with CGTCGGT and then the pre-existing read takes over at position 19.with GA.

You could also be forgiven for not having the confidence to figure out anything before 19, and **you could just erase (trim) all the N’s in the first 18 positions** to make sure you don’t interpret anything wrong to avoid making your sequence look more different from other sequences than it actually is.

Trim the other end of the sequence in the same way.

**Check and Correct the Entire Sequence**

**Graphical user interface

Description automatically generated**

Look through the whole thing and **correct any N’s you find**. Position 21 could not be read (N) but I can tell it is a C. Position 32 could not be read but I know it is a T. Position 51 could not be read but I know it is a C.

Also **deal with double peaks**. Position 67 was read as a G but the T peak was almost as tall. This mushroom’s two alleles had slightly different sequences, so position 67 is a G in one allele and a T in the other. **MEGA does not deal with double peaks, while more expensive software might do this for you**. Looking up the “nucleotide” Wikipedia page, I see that G plus T is the secondary nucleotide K, so I put a K at position 67. The occasional double peak is not uncommon. More than a handful of them (like my unhealthy example above) means there is a problem.

If you don’t put the secondary nucleotides in, two sequences may appear more different than they are. See chapter 3 bullet point 3 of my “Analyzing DNA sequences” document.

**Combine the forward and reverse reads**

**MEGA does not combine forward and reverse reads, while more expensive software might do this for you.** You may have to create both a forward and reverse FASTA file, and then create the reverse complement of the reverse read (write it backwards swapping A<->T and C<->G. See how to do this in section 2 later in this document). Compare the two and create one FASTA file that uses all the parts of both (one may be longer than the other on one end) and find any place that they differ and decide which one is correct in those places. This is not practical to do with MEGA alone.

**Create the FASTA**

From the DATA menu, choose “**Export FASTA file**”. Now you have a FASTA file that you can use for making trees (keep reading) or analyzing (see my “Analyzing DNA sequences” document).

Edit the file to **put a meaningful name for the sequence** after the “>”, with your best guess as to what you thought it was when you collected it, being sure to admit that you haven’t analyzed the sequence yet and don’t know for sure what it is. I like to use the term “cf” (confer) to mean it looks like something but it hasn’t been proven, eg. “**>Cortinarius cf violaceus**”.

**Collect all your sequences from a certain genus together in one file**. Depending on the genetic diversity found within a certain genus, you may be able to have one file with more than one genus in it, or you may only be able to have one section of one genus in a file. If you use BLAST (chapter 2) instead of MEGA (chapter 3) it is much, much quicker and easier to use for preliminary analysis, but BLAST becomes useless if all the sequences in your file aren’t closely related enough. ITS sequences are meant for determining if two collections are the same species or not, not for determining genus or family, so even if you use MEGA your trees will not be correct if your sequences are not closely related enough, although it can do a much better job than BLAST.

**Optional - Trim every sequence to just ITS**

Some sequences get more than one gene at a time. Most common is ITS followed by LSU, since they are both popular genes and right next to each other. If you are only interested in ITS because all of the rest of your sequences are ITS only, it might help to isolate just the ITS part of the larger sequence. For large files, the tree building could become faster without the unnecessary extra data (see below for how slow the tree building can be). Also, with all the sequences a common length, the tree building software can sometimes do a better job.

For example, to isolate just ITS from a combined ITS/LSU sequence: The end of ITS and the beginning of LSU is usually delimited by TGACCT and then CAAAT (when searching for this pattern, remember you won’t find it if there is a line break in the middle of it). You can delete everything after that. However, the first part might be TGATCT, TGAGCT, or TGTCTT instead. The second part might be CAGAT or CGGAT. You might have to look for some permutation of those.

**2. Quick BLAST trees**

**.FASTA file format for BLAST**

For BLAST, **your fasta file can have GenBank numbers as well as full sequences** that start with “>”, which is handy. You must start with a list of (usually) 8 digit GenBank numbers, which must all be together at the top of the file, and once you start putting in explicit sequences, you cannot go back, you can no longer use GenBank numbers in the file. In your tree, BLAST will use the long, confusing GenBank name for the sequence. I prefer not to use GenBank numbers in my files, but to write them all out explicitly, for two reasons: I can control the string that gets used for the name of the sequence and I can use MEGA to make a tree, which does not support using GenBank numbers.

Here is an example fasta file with 4 genbank sequences and then two explicit ones of my own:

KM253741

HQ204707

MK386834

FR852106

>Cortinarius violaceus from my backyard

AGGGTATATAGATCGCGCGCGATATATA...etc.

>Aspropaxillus septentrionalis iNat 65425700

AGTGCTTTCGAGAGAGAGGATCTCTGAG…etc.

**Don’t attempt more than one genus**

ITS is what we have the most sequences of, so chances are you're making an ITS tree. That is usually only good for genus rank or lower, so don't try to make a tree with more than one genus in it, it may be wildly inaccurate. Since the rank of genus is an arbitrary human construction, sometimes a few related genera can work in the same ITS tree, and sometimes, one genus is too much and you can only get a good tree at the subgenus or section level.

**Making a BLAST tree**

Using BLAST to make trees is much quicker and easier than almost any other way, but it’s not always as good, neither in aligning nor in tree building. It’s worth a try though. Visit BLAST at <https://blast.ncbi.nlm.nih.gov> and click on “Nucleotide BLAST”.

* Click on “Align two or more sequences” below the entry box to get a page with two boxes, a Query box and a Subject box. Open up the “Algorithm Parameters” near the bottom and change the “Max Target Sequences” from 100 to the number of sequences in your fasta file if there are more than 100.
* Put your entire fasta file in the bottom box. Pick one sequence to put in the top box as the one to compare to all the other sequences. Choosing one is tricky. If you just want to see specifically how one new sequence compares to the others, that is the obvious one to put in the top box, but if you are making a tree, choosing one can be tricky. You may have to try several choices and see which tree looks most correct. Unfortunately, the quality of the tree will vary wildly depending on your choice of top box sequence.
  + Choose **a sequence that has many similar sequences in the file**, not one that stands out on its own, for the **top box**.
  + Choose one of your **longest sequences** for the **top box**; choosing a short one will not look at any of the regions outside of the regions sampled in your short sequence and you will be ignoring much of the available information when making the tree and it will be worth little.
* Do not change “Highly similar sequences (megablast)” if you are using ITS sequences (which you probably are if you are getting data from me) as that is the best choice for ITS trees, a region meant for comparing similar sequences. If your sequences are not all in the same genus (or sometimes even the same section of a genus), ITS might not work that well anyway, so you won’t get any real benefit from playing around with this option.
* Click on “Show results in a new window” near the bottom and then the big BLAST button.
* On the new second page, click on “Distance tree of results” at the top of the right-hand columns.
* On the new third page, unless your tree is very small, you won’t see all of it. Click on “Tools/Expand all” and then “Tools/Download/PDF file”. Click “Create PDF-file” and then click “View”
* **Don’t bother to play with “Tree method”** on the third page or any other tree options, they’re all bad methods and if you really care about getting a good tree it’s better to spend your time using MEGA which I describe next than mucking around with the options in BLAST. Not having a good tree building method is one of the problems with BLAST. The other is alignment.

**Alignment**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| A | C | G | G | G | T | A | T | C | G | A | A | T | C | C | A | G | T | C | T | T | G | C |
| A | C | G | G | T | A | T | C | G | A | A | T | C | C | A | G | T | C | T | T | G | C |  |

Consider these two sequences. **They appear to only be the same for the first 4 characters**, and then they are completely different from each other in almost every position from 5 on. But if you look closely, the bottom one just lost a G near the beginning. Once you notice that and write them down this way:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| A | C | G | G | G | T | A | T | C | G | A | A | T | C | C | A | G | T | C | T | T | G | C |
| A | C | G | G | - | T | A | T | C | G | A | A | T | C | C | A | G | T | C | T | T | G | C |

You realize **the sequences really are almost identical except for one small difference**. Figuring this out is what is meant by aligning. And BLAST does a terrible job if the difference is above 20% or so; it just gives up.

**Here are the signs that you need to use MEGA** because BLAST made a useless, terrible tree.

* Some sequences are clearly in the wrong place in your BLAST tree.
* When the “Query cover” numbers in one of the right-hand columns down the **second page** are below 50% for a significant number of sequences. This number tells you how much of that sequence overlapped with the sequence in your top box. If they are both ITS only sequences of about the same length, that number should be around 100%. If one was a whole ITS sequence but the other only had ITS1 or ITS2 instead of both, then the number will be around 50%. If one sequence was an ITS+LSU sequence twice the length of an ITS only sequence and the other was only ITS1, the number would be about 25%, and that would be normal and acceptable. The problem with using “your longest sequence with many nearby similar sequences” as the top box sequence (as I recommend you do, above) is that will often make the “Query cover” numbers low, as most other sequences will be shorter than it, so it will be harder to tell if there’s a problem.
  + *Quick method*: For the purposes of judging the quality of a BLAST tree, you can put a normal sized ITS sequence in the top box (600-700 characters long) and then look for **“Query cover” results that are <50%**. This provides a “quick guess” of whether or not a BLAST tree will be any good.
  + *Long but accurate method*: To determine with confidence if low “Query cover” percentages are a problem or not, you need to take the time to click on the **“Alignments” tab on the second screen**.

Text, table

Description automatically generated

This only had a 38% query cover, but it might be OK. Most of the short subject sequence was used, starting at position 1 up to 398 which is near its end. The first page tells us the subject sequence was 427 long, so 29 characters are missing at the end. The first page also tells us that the query sequence was 948 long, so there were 29 characters after 854 to possibly align with the subject sequence, but BLAST could not do it. So 326 to 368 and 399 to 427 did not align, but 83% of the subject sequence did, it’s just much shorter than the query (the first 518 characters of the query were from a region earlier than the region that was sequenced for the subject, so of course they can’t be compared). Now, missing 17% of the data is going to hurt the quality of the placement of this sequence in the tree, for sure. But it’s not necessarily a deal breaker.

Text, letter

Description automatically generated

This only had an 18% query cover, and it is definitely not going to be of any use. BLAST was unable to align the first 250 characters of the subject, nor the last 262 characters (the first page tells us the subject was 683 characters long). So only 25% of the data of this sequence are aligned and being used in the tree. There’s no way this sequence is going to be placed anywhere near its proper place in the tree with 75% of the data missing.

There’s a quick and dirty way to estimate fairly accurately if a sequence will not work in a BLAST tree without going to as much trouble as I just did. Either the beginning or end of one of the two sequences must be used at either the beginning or the end of the comparison, or you are only comparing their middle sections and all four ends are missing. In other words, **check the first and last query and subject numbers, and at least one of those four numbers needs to be the number “1”** (or close to 1, say <5). In the first example, “Sbjct 1” shows that the beginning of the subject sequence is used, so it passes the test. In the second example, the smallest of the four numbers (“Query 486”, “Sbjct 251”, “Query 656” and “Sbjct 421”) is found at the beginning of the subject, and it’s 251, and that’s terrible. We check the numbers at the end as well as the beginning because one sequence might be a “reverse complement” (backwards compared to the other). More on that below.

If you’re not OK with having the sequences that fail this test show up in the wrong place in your tree, it’s time to step up and use MEGA. Not only will BLAST not align sequences that are too far apart, their tree building methods “Fast Minimum Evolution” and “Neighbor Joining” are inferior to the “Maximum Likelihood” trees that MEGA can make.

Don’t get me wrong. **BLAST is awesome** and I use it far more often than I use MEGA. A super computer is doing the work for you. If you used MEGA with the same quick but inaccurate alignment and tree building algorithm that BLAST uses, MEGA on a typical laptop would take ONE HUNDRED TIMES LONGER to do exactly what BLAST does (BLAST can align and then build a tree of 500 sequences in 30 seconds instead of 30 minutes). If I didn’t have BLAST, my entire last year’s worth of work would have resulted in me having something to say about 10 PNW mushroom sequences instead of 1,000 of them.

**3. MEGA - Make a compatible .FASTA file**

**Is MEGA the best?**

No, but it’s pretty darn good and it’s free. Educational institutions use *Sequencher* or other similar program, but they can be so expensive they won’t even tell you how much it is before a sales person meets with you. Think hundreds of dollars a month.

As I will explain below, **BEAST** should be investigated with its **Bayesian Inference algorithm**. That has the potential to make better trees than MEGA. The accuracy test for BEAST is not the same kind of bootstrap support algorithm, it has its own technique.

**MEGA fasta file limitations**

* You can’t use GenBank numbers in your fasta file
* You can’t use comments that begin with a semicolon “;”

BLAST lets you have FASTA files with the above, but MEGA does not, so if yours looks like this, with some GenBank numbers and comments:

;this is the neotype sequence from Sweden.

KM253741

>Cortinarius violaceus from my backyard

AGGGTATATAGATCGCGCGCGATATATA...etc.

...then you'll have to go to GenBank [www.ncbi.nlm.nih.gov/nuccore](http://www.ncbi.nlm.nih.gov/nuccore) and type KM253741 into the search box to get its actual sequence. When the page comes up, click on FASTA (found on the left underneath the long name of the entry) and then copy and paste everything from ">Cortinarius violaceus blah blah" until the end of the sequence. Replace KM253741 with this longer string. Remove the comment. Now your FASTA file will look like this:

>KM253741.1 Cortinarius violaceus var. violaceus voucher MM1974/0208 18S ribosomal RNA gene, partial sequence; internal transcribed spacer 1, 5.8S ribosomal RNA gene, and internal transcribed spacer 2, complete sequence; and 28S ribosomal RNA gene, partial sequence

ATTTAGAGGAAGTAAAAGTCGTAACAAGGTTTCCGTAGGTGAACCTGCGGAAGGATCATTATTGAAATAAACCTGATAAGTTGCTGCTGGCTCTCTAGGGAGCATGTGCACACTTTGTCATCTTTATATCTTCACATGTGCACCTCTTGTAGACTTTGGATATCTTTCTGAATGTAATTCAGGTTTTGAGGATTGACTTTTTGGTCTCTCCTTACATTTCCAAATCTATGTTCCTTCATATACACTTATGTTATAGAATGTAATAAAATAGGCCTTTTTTGCCTTACAAAAACCTATACAACTTTCAGCAACGGATCTCTTGGCTCTCGCATCGATGAAGAACGCAGCGAAATGCGATAAGTAATGTGAATTGCAGAATTCAGTGAATCATCGAATCTTTGAACGCACCTTGCACTCCTTGGTATTCCGAGGAGTATGCCTGTTTGAGTGTCATTAATATATCAACCTCTTCAAGCTTTTGCTTGTCGAGTGTTTGGATTTGGGGTTGTGCTGGTCTTATTTTCTGAAGGTCAGCTCCCCTGAAATGTATTAGCAGAACAATTTGTTGACCGTTCATTAATGTGATAACTATCTACATTATTGATGTGTGAGGCAGTAGTTCAGCTTCTTAACAGTCCATTGACTTGGACAAATTTTCATTAATGTGACCTCAAATCAGGTAGGACTACCCGCTG

>Cortinarius violaceus from my backyard

AGGGTATATAGATCGCGCGCGATATATA…etc.

**Naming your sequences**

I don't like the long messy names GenBank gives to sequences, so I usually rename them to something short and more useful. I take away the ".1" that isn't necessary at the end of the GenBank #, and I indicate why I trust the name given to this sequence (in this case saying that it is the neotype sequence) and I put where it is from. The string after the ">" is what you are going to see in the tree, so if you want to understand what the sequences in your tree are, you'll want to make them as short and descriptive and possible. If they're too long, they will run off the side of the page, and that's always frustrating. Here is what I did:

>KM253741 Cortinarius violaceus NEOTYPE Sweden

ATTTAGAGGAAGTAAAAGTCGTAACAAGGTTTCCGTAGGTGAACCTGCGGAAGGATCATTATTGAAATAAACCTGATAAGTTGCTGCTGGCTCTCTAGGGAGCATGTGCACACTTTGTCATCTTTATATCTTCACATGTGCACCTCTTGTAGACTTTGGATATCTTTCTGAATGTAATTCAGGTTTTGAGGATTGACTTTTTGGTCTCTCCTTACATTTCCAAATCTATGTTCCTTCATATACACTTATGTTATAGAATGTAATAAAATAGGCCTTTTTTGCCTTACAAAAACCTATACAACTTTCAGCAACGGATCTCTTGGCTCTCGCATCGATGAAGAACGCAGCGAAATGCGATAAGTAATGTGAATTGCAGAATTCAGTGAATCATCGAATCTTTGAACGCACCTTGCACTCCTTGGTATTCCGAGGAGTATGCCTGTTTGAGTGTCATTAATATATCAACCTCTTCAAGCTTTTGCTTGTCGAGTGTTTGGATTTGGGGTTGTGCTGGTCTTATTTTCTGAAGGTCAGCTCCCCTGAAATGTATTAGCAGAACAATTTGTTGACCGTTCATTAATGTGATAACTATCTACATTATTGATGTGTGAGGCAGTAGTTCAGCTTCTTAACAGTCCATTGACTTGGACAAATTTTCATTAATGTGACCTCAAATCAGGTAGGACTACCCGCTG

>Cortinarius violaceus from my backyard

AGGGTATATAGATCGCGCGCGATATATA…etc.

I have written an entire chapter on how I name my sequences in FASTA files to be optimally useful. See the appendix of my “Analyzing DNA sequences” tutorial.

**One genus (or section of a genus) at a time**

The same restrictions apply to MEGA that applied to BLAST. ITS inherently doesn’t have enough information to make an accurate tree above the genus level (to generalize) or sometimes even above section of a genus, no matter how much math you do.

For large genera or files with >100 sequences in the FASTA file, this process might be very slow, so you might want to only do one subgenus or section of the genus at a time. Of course, if you don't know which actual generic molecular section each species is in (which may have nothing to do with the section the species is officially in) then you can't do that, so you'll have to make a large tree of the whole genus first to try and figure out which species are in which section. Then you can try and do only one part of the genus at a time.

**Reverse Complement**

You will sometimes see a species clearly in the wrong place, often outside of a group containing every other sequence, indicating it is not even in the right genus. This could be because that particular sequence was written backwards (compared to the universally accepted forward direction). BLAST doesn't care, but MEGA does.

A picture containing text

Description automatically generated

The bottom sequence stands alone, but it is supposed to be the same species as the one pointed to by the red arrow. In fact, I remember looking at the two sequences in BLAST and they were identical. What gives?

BLAST the two sequences (as described in my BLAST tutorial at the beginning of chapter 2), with one in the top box and one in the second box and compare them by clicking on the only sequence showing in the description list.

Text, table

Description automatically generated

Look at that. While the Query goes UP from 250 through 676, the Subject goes DOWN from 427 to 1. They’re backwards! Well, not backwards, one is actually the **reverse complement** of each other. If you imagine a double helix of DNA and you read from the top down of one strand you get the forward read. If you start at the bottom of the other strand and read up, you get the reverse read and the string will be backwards but also A’s are switched to T’s and vice versa and C’s are switched to G’s and vice versa (if you understand what the actual molecule looks like, this will make sense, and is covered in my

Do a BLAST with your entire .FASTA file in the bottom box and just one of them in the top box (perhaps one you suspect of being backwards). Look through the entire “Alignments” tab page and see if anything is backwards from anything else to identify which sequences might be backwards. Or, just wait until you make your first MEGA tree and identify anything suspiciously in the wrong place and double check those.

To fix a reverse complement problem, Google “Reverse complement dna webtool” or visit <https://www.bioinformatics.org/sms/rev_comp.html> and type your sequence into the box and click submit. The next page has the reverse complement sequence you can copy and paste into your fasta file to replace the backwards read with a forward read so that MEGA will be able to put it in the tree properly.

You may get an error on that web page if your sequence has any ambiguous nucleotides. For instance, K means one allele has a G and the other allele has a T, so K kind of means “either G or T can go here”. But once you make a reverse complement, G becomes C and T becomes A, so you need the ambiguous character that means “either C or A can go here” and that is M. So if you get an error, you will have to replace K with M, W with S, and R with Y (and vice versa) by hand, because this web page is not clever enough to do that for you. See the Wikipedia page on “nucleotide” for a full description of what each letter means: <https://en.wikipedia.org/wiki/Nucleotide> (scroll down to the chart at the very bottom). Rarely, you may see other letters and need to know what to do with them. N (unknown) can stay as N, and the web page handles that.

If the error is due to a dash “-“, you can safely ignore the error and let the page omit the dashes, as those are used to show sequences that have already been at least partially aligned (see above), but you’re going to align them again anyway.

Diagram

Description automatically generated with low confidence

Oh look! The sequence is now where it belongs with the other Woo 6 sequence once I changed it to its reverse complement.

**3. Using MEGA**

**How long will this take? O(n^2)**

Making a tree has Order(n^2), meaning if you put four times as many sequences in your file to make a bigger tree, it might take 16 times as long to make the tree, so short trees are way less frustrating. Here are some approximate times with files of ITS sequences that sometimes include other neighbouring regions to make for longer sequences (with average length of 500-1000 characters). These are times from my (average?) laptop with an i3-8130U CPU @2.2GHz and 8G of RAM. Note that these times can be 100x slower than BLAST, so you are paying for accuracy with time. It may take half an hour to make a tree from a file with 500 sequences.

|  |  |  |  |
| --- | --- | --- | --- |
| **# Sequences** | **Align** | **ML Tree** | **Total** |
| **25** | :03 | :03 | :06 |
| **50** | :09 | :14 | :23 |
| **100** | :26 | :29 | :55 |
| **200** | 2:07 | 1:35 | 3:42 |
| **500** | 13:30 | 16:15 | 29:45 |

**Trim your sequences - Isolating ITS from SSU and LSU**

If you really need to have 500 or so sequences in your tree at the same time, it might be worth your while to trim your sequences so they are short sequences, since 500 short sequences will align way faster than 500 long sequences. Another good reason is that you get a better tree when all your sequences are the same length showing the same genes. A few longer sequences with extra data doesn’t mean a better tree, it could be a worse tree.

You can often tell by looking at a sequence that it’s much longer than the others, but MEGA can help you find sequences that are longer than the others. More details are in the “Alignment” section below this one.

Many ITS sequences contain parts or all of the 18s SSU (short subunit) and 28s LSU (large subunit) genes (28s is called 25s in plants). This is how this part of the genome looks:

18s-SSU **:::** ITS1 **:::** 5.8s **:::** ITS2 **:::** 28s-LSU

If you aren’t deliberately collecting multiple genes of each sequence, and you intend to align and create a tree for hundreds of sequences at the same time, get rid of the SSU and LSU parts of the ITS sequences to make them shorter. After a while, you’ll get a feel for how long an ITS sequence is, and the really long ones that contain extra genes will stand out to your eye. Trim those.

You will more commonly find ITS+LSU sequences as of this writing than SSU+ITS sequences. LSU starts with TGACCT (although occasionally some genera have TGATCT or TGAGCT). This will be immediately followed by CAAAT for basidios, or either CAGAT or CGGAT for ascos. Search your long basidio sequence for TGACCT, ensure that it is followed by CAAAT, and then delete everything after that point. Note that if there is a line break in the middle of TGACCT, search will not find it, so you should search for CAAAT and then verify that it is preceded by TGACCT.

If you do have an SSU+ITS sequence, ITS starts with ATCATTA, which follows AAGG. You can trim away anything before that in the sequence. Remember, if there is a line break in the middle of the ATCATTA, search will not find it.

For completeness, 5.8s begins with CAACTTT, with an optional C in front of that.

ITS2 begins with AATTCTC (or possibly AATCCTC).

For all of these, be careful, because it’s possible for these patterns to exist elsewhere.

**Alignment**

MEGA uses two main methods for aligning: CLUSTALW and MUSCLE. Some have said that **MUSCLE** might be a little better, so I’ve been using MUSCLE, but feel free to try both as I can’t personally vouch for one over the other.

Drag your corrected fasta file onto MEGA (it must end in the .fasta extension to be recognized), then choose “Align” (not “Analyze”). If you can’t drag and drop, click the “Align” icon then “Edit/Build Alignment” then “Retrieve a sequence from a file” and point it to your fasta file.

Once the new window opens up, choose the “Alignment” menu item and “Muscle” (or ClustalW if you want). **DO NOT CHOOSE A CODON OPTION** if you are using ITS, that is for the parts of DNA that encode for proteins, where only certain strings are possible (to indicate which protein to make) instead of any random DNA letter (A, C, G or T). ITS is “junk DNA” which does not do anything, and certainly doesn’t encode for proteins, and is certainly allowed to have more than a few possible combinations of the letters. It may ask if you want to “select all”, if so, yes, you do.

All the default parameters are pre-set for accuracy over speed, and it won’t be tangibly faster if you reduce the accuracy, so leave everything alone and just press OK.

**If you get an error** – make sure you don’t have any GenBank numbers in your file, any comments, or the characters ; : or an unmated double quote “.

From a few seconds to a few hours later, depending on how big your file was, the pop window will close and it will be done. Choose Data/Export Alignment/MEGA format” and save the .MEG file. Now you can close the Alignment Explorer window.

The window of aligned sequences should be filled with coloured squares in most rows and columns. If you see mostly blanks, or you see one or more rows filled with coloured squares, but the rest are blank, you have some sequences much longer than the others. If this tree is important it might be worth the trouble to “Trim your sequences” as described in the section above this one to get the best tree possible.

If you want a possibly even better alignment done, there may be free online tools that use a “**Guidance2**” algorithm. Perhaps you can find one that will allow you to save the result in a .MEG file format which you can then import into MEGA to build a tree. I haven’t figured out how, but nor have I really tried yet.

**How good was that alignment?**

In the main MEGA window click the “Distance” icon and choose “**Compute Overall Mean Distance**”. If it doesn’t know about your .MEG file yet, point it to it. **If the resulting number is >0.3, your alignment will probably produce a somewhat crappy tree**. That might mean:

* Your sequences represent too diverse a group. Limit yourself to one genus, or split even further into sections if you can.
* You have sequences that are isolated too far from the others, without any “missing links” between them to show how one evolved into the other.

“Distance/Compute Pairwise Distances” brings up a table where you can see how far apart any two sequences are. Wherever a sequence is more than 0.3 apart from another sequence, that can be a problem; it will not know where to put them relative to each other in the tree. But if there is an in between sequence that is <0.3 from both, it can use that as a stepping stone to figure out the tree. So it’s not necessarily a problem if you can find two sequences >0.3 apart, as long as you can find a path between them using other sequences with low numbers between each step.

**Building a Tree**

**NOTE**: Some characters from the name of your sequence that I use regularly, like **?, # and %, won’t print in MEGA trees**. This makes it hard to read the trees and know exactly what the sequence was trying to tell you. (e.g. “2% different than type” becomes “2 different from type”, and you can’t tell when I’m not sure about something because “EU?” becomes “EU”).

In the main MEGA window, press the “Phylogeny” icon. Choose “Construct/Test Maximum Likelihood Tree”. This is the best quality tree option, the others are not much faster yet produce worse results. If you wanted to save time and sacrifice accuracy, you’d just use BLAST which can do the poorer tree options 100x faster than MEGA. Now point the app to the .MEG file you saved in the Alignment section above.

**TIP - MEGA Crashing Bugs** – MEGA seems prone to crashing when making Maximum Likelihood trees. Often, if I just try a second time, it will work, but if you continue to be plagued by this problem, change the last parameter in the box “Number of Threads” to 1 instead of 4. That should fix it, although the tree may take a little longer to build.

If you’re interested, here’s how the different algorithms stack up:

* Quickest but least accurate
  + UPGMA – outdated and bad, nobody should use this anymore as others are just as quick and better
  + Neighbor Joining – BLAST’s other option (not the default), it’s the choice for those who really need speed and can sacrifice some accuracy. It makes a distance matrix, joins two nodes and makes another distance matrix, whatever that means. If you’re going to try this with MEGA make sure NOT to select the “complete deletion” option for “Gaps/Missing Data Treatment” on the parameters page.
* Mid-tier algorithms (a little slower and a little more accurate, but not terribly slow). They assume that DNA evolved from one string to another in the least possible number of steps (substitutions). If one organism has an A in one place, and another has a C, perhaps over a million years that location mutated from A to C to A again then to C again, but these algorithms won’t acknowledge that possibility, they will assume that spot mutated once from A to C. That’s the limitation that is going to keep these algorithms from being truly useful.
  + Minimum Evolution – BLAST’s default that you used above when playing with BLAST, the best BLAST can do.
  + Maximum Parsimony – an algorithm with similar quality/speed tradeoffs, but these trees do not have branch lengths and if you want them, they will be very slow to calculate. SeaView does a fair job of using this algorithm, but apparently MEGA doesn’t.
* Best algorithms (slowest but most accurate)
  + Maximum Likelihood – the best MEGA can do, use this one
  + Bayesian Inference – better, but MEGA can’t do it. You’ll need BEAST, which is also free, but I do not have instructions on how to do that yet.

**Making a Maximum Likelihood (ML) Tree - Options**

The “Models” icon and “Find Best DNA/Protein Model” option can help you figure out the right parameters to use based on your specific alignment. More on this later, but right now it seems to be pretty much known what choices are usually the best, and here they are:

Test of Phylogeny – choose “none” for now, then read “Calculating Branch Support” below.

Model/Method

* Maximum composite likelihood - assumed that any mutation was as equally likely as any other, and that they happened at a constant rate over time. If you have specific knowledge of an era when a sunspot storm caused faster mutations and know which species were created during that time and can integrate that knowledge into building a tree, power to you, you are much smarter than I am, but the rest of us are going to have to go by some of those assumptions.
* Jukes-Cantor – they were the first to have an algorithm that understood that two random sequences that had nothing to do with each other would still be 25% similar, not 0% similar. Just on pure luck, one out of every 4 characters will match in a sequence, as there are only 4 possibilities – A, C, G or T. Infinite time passing will mutate one sequence into another that is only 75% different. They use this fact to create a curve that says, for instance, that if two sequences are 50% different, that doesn’t mean that 50% of the locations mutated (some of them will have mutated back to their original value and not look like a mutation). On their curve, two sequences that are 50% different indicate that 79.45% of the locations have probably changed, it’s just that some of them changed back to their original value so it only looks like 50% of them actually changed. This greatly enhanced the quality of the trees.
* Tamura-Nei – they added another insight, that A switches to G (and vice versa) and C switches to T (and vice versa) (more easily than A can switch to C or T, C can switch to A or G, G can switch to C or T, and T can switch to A or G). The reason is probably biological, having to do with the exact size and shape of the molecules that actually have to break off and float around and stick to a new place.

Tamura-Nei has the most clever insights, so use that one.

Rates among Sites – Uniform Rates. This may not be true for all areas of DNA, but for ITS, a junk area, supposedly changes here are pretty random.

Gaps/Missing Data Treatment - Use All Sites. Sometimes you don’t want to consider the insertions and deletions that made sequences hard to align the same way you consider a change from one nucleotide to another, but again, in the ITS junk area, it is assumed that it is all to be considered.

Tree Inference Options – leave these alone as I don’t understand them yet.

Number of Threads – leave at 4, unless you keep crashing trying to make your tree. Then you might want to try changing this to 1, but it will be four times slower.

**If you get an error** – make sure you don’t have any GenBank numbers in your file, any comments, or the characters ; : or an unmated double quote “.

**TIP – MEGA Tree Width/Height Bug** – After MEGA builds the tree, I find it does not set the tree width properly, so it can be hard to read. I go into the **LAYOUT** options and increase the **TREE WIDTH** to maximum, and then back to minimum, and that fixes it. If your tree is very small, it may also not get the height of the tree correct, so I increase the **TREE HEIGHT** to maximum, and then back to minimum again to fix that.

After the tree is built (which may take a while), **save your work** so you don’t have to spend all that time building the tree again later with the same data. Choose **FILE/SAVE CURRENT SESSION**. Also, you can save an image of the tree using the **“Image”** menu option and you’ll probably want **“Save as PDF file”**.

**How accurate is the tree?**

What are the odds that your tree is correct? Pretty low. Every time you so much as change one sequence or even one letter in one sequence, and re-make the tree, you’ll be surprised how different a tree you get the next time.

You can get MEGA to calculate the percentage odds that any given branch is correct (that the species inside that group are more closely related than anything found outside that branch), but it’s VERY SLOW. Set “Test of Phylogeny” to “Bootstrap Method” and choose the **“No. of Bootstrap Replications”** you want it to make. To get accurate probabilities for a publication, they say to choose a number between 100 and 2000, but that may take days. I find that even **using 10-25 gives a lot of valuable information**. It is worth doing 10 at some point. The program will then create that many different trees by shuffling the columns in the alignment, with replacement, which means it won’t necessarily try that many different trees, there could be duplicates, so that’s why the lower number of repetitions might not work so well. At the end of the process, for each branch, it will tell you the percentage of trees that had that branch together as a part of the tree, which is a pretty good reflection of the odds that the species inside belong together (in fact they say the true odds are probably higher than shown). But here’s the problem: The tree building process will be AT LEAST THAT MANY TIMES SLOWER. Remember how it takes 30 minutes to make a tree with 500 sequences? If you choose 1000 trees, it will take 500 hours to build your tree instead of 30 minutes. That’s the better part of a month. That’s why I usually choose a small number, which still gives you valuable. If you can stick to 100 sequences, it will take less than an hour to do a 50 tree version, and you’ll get percentages quantized to the nearest 2%.

This is where it matters that Maximum Likelihood trees are slower to create than other trees. If you just make the tree, with no accuracy test, you might not even notice the difference in the time it takes, and so you might as well choose the best tree, Maximum Likelihood. But if you are going to do the “Test of Phylogeny”, you might be tempted to make a faster tree, like “Neighbor Joining” and do that 50 times instead of making 50 Maximum Likelihood trees.

I cannot stress enough that **TAKING AN EXTRA WEEK TO BUILD YOUR TREE DOES NOT MAKE A BETTER TREE**. YOU STILL GET THE SAME CRAPPY TREE. But now you have an idea of exactly HOW crappy it is.

Tip: If you have a large tree and don’t want to wait a month for it to give you some support values, you can start a tree with 50 iterations and **cancel it after at least 25 iterations** have run. It will show you what it has so far. It won’t be very good, but it might be better than waiting a lot longer for more iterations.

**Interpreting the tree**

A **node** (where branches meet) represents a hypothetical ancestor of both branches, which may belong to (and often does belong to) an extinct species.

The **horizontal length of a branch** represents the number of mutations that happened, which probably represents how many thousands of years went by. The longer the branch, the less likely it is to be in the right place.

For instance, in the tree below, at node 6 (the 88% node) the species represented by “*Deconica cf crobula UNITE EU*” that is the bottom half of that node may be the ancestor of the entire top half of that node, since there is no horizontal distance between it and node 6.

You can also see that #3 is on one of the longest branches, and indeed its likelihood of actually being there is pretty low (34%) compared to most of the shorter branches that have much higher probabilities. We might have guessed that without taking the time to calculate support values.

#5 is interesting. It is on a long branch but it was placed accurately (92%) anyway. This is a case where the % could not be inferred and taking the time to calculate the branch support added value.

#1 is a *Cortinarius* outgroup sequence, deliberately in a different genus, explained below in “Rooting Your Tree”.

We can’t tell if #2 is a *Deconica* species or not. It’s outside of the rest of them, so we won’t know if it’s inside the genus or not unless we add other sequences known to be *Deconica* that are even more outside of #2 (then #2 would be in *Deconica*) or we add other sequences that aren’t *Deconica* that end up grouping with #2 to show us its true genus.

Not much is known about #4 either. It only has a 24% chance of belonging there.

#6 is our best node. I think we found a section or subgenus of *Deconica*. With 88% confidence everything to the right belongs together in a section. It doesn’t matter that some of the numbers to the right of #6 are low (there’s only a 28% chance that *Deconica inquilina* is shown in the proper part of “section 6”), but it doesn’t matter. We can still confidently say that all of those species are in a section together, somehow.

#6 is the top branch of a node with 52% support. The bottom half of that 52% node may not be a section, as there’s only a 48% chance that #5 belongs with the others.

Diagram

Description automatically generated with medium confidence

**Rooting Your Tree - Outgroups**

Don’t be fooled, the tree is unrooted. That means you can’t tell which species evolved first and which evolved later, but you’re going to think you can, and you’ll be wrong (until you properly root your tree).

Diagram

Description automatically generated with medium confidence

For instance, in the above tree, it looks like #1 represents the beginning of time, and things evolved from there, with “*Deconica cf coprophila Vanc BC*” and “*Deconica cf merdaria 1821/coprophila 1793*” being one of the most recently evolved species (found to the right of point #2) since those are very far to the right. Maybe.

But what if… you thought of point #2 as the left most, oldest point, and moving left from #2 towards #1 was actually moving forwards in time? That would mean that “*Deconica cf coprophila Vanc BC*” was one of the *oldest* species, not the newest species, and… well… let’s redraw the tree to find out the rest.

A picture containing diagram

Description automatically generated

Now, it looks like *Deconica horizontalis* is one of the most recently evolved species, and so is *Deconica montana*, as they are furthest to the right. Which way is correct? **There is no way to know**. We can see the differences between species A and B, but there’s no way to know if B turned into A or if A turned into B. There’s no way of knowing which way time was flowing and which are the new species and which are the old species.

So you have to root your tree, by finding two sequences where you do know the direction of evolution. If you know that direction, you can figure out which direction other parts of your tree must have gone. But how are you ever going to know for sure when one mushroom evolved into another instead of the other way around? Well, consider a mushroom in a different genus than *Deconica*, like *Cortinarius*. We know that the common ancestor of *Deconica* and *Cortinarius* had to come first, before any extant species in either genus did. So we add one single extra sequence to our fasta file, an “outgroup” sequence from a group outside the group we are interested in for our tree, that is definitely more distantly related to every other mushroom in the tree than any possible combination of two mushrooms inside the tree.

Now let’s go back to the tree illustrated in the section “Interpreting the tree” that has a *Cortinarius* sequence on the bottom. The tree builder was able to figure out that since that sequence was so much more different than anything else, the true oldest point in time that deserves to be furthest left was the node between the *Cortinarius* and everything else. That tree actually shows which species evolved first and which came later. Unless… *Cortinarius* evolved from a *Deconica* mushroom. You need to know for sure (from microscopy or other analysis) that you picked an outgroup member distantly related enough that it truly could not have evolved from anything inside your tree. Here it is again.

Diagram

Description automatically generated with medium confidence

**Playing with your tree**

You can customize the fonts and graphics and spacing used for the nodes, branches, labels, etc. until it’s not obvious that MEGA generated this tree and it looks completely hand drawn to your own graphical tastes. There are lots of options you can play with.

If can estimate how long ago different branches occurred. It can even give you a guess as to what the DNA could have been for one of the extinct ancestors (if you chose “Use All Sites” for the “Gaps/Missing Data Treatment” option above) so you can try and recreate an extinct species in your lab.

You can even add icons of your mushrooms to show pictures.

Compute/Compute Condensed Tree (cutoff 50%) will remove numbers <50% and collapse their nodes (making an n-ary tree instead of a binary tree) to show low probability branches a little more vaguely (which is being honest) instead of showing them in a particular place but admitting that they’re in the wrong place. This tree doesn’t show actual length of time between evolution of species.

A picture containing diagram

Description automatically generated

**What arguments DNA can and can’t settle**

NOTE: DNA will never settle some arguments. Do you think of a group of collections as all one species or split them into 5 species? That is, and may always be, a matter of opinion. DNA won’t help you decide at what branch to declare a genus, family or species. It can only tell you how related two organisms are and let you use your judgement about where to draw the boundary lines.

**Other shapes of trees**

There’s no reason your tree has to look like a traditional tree. Try VIEW/TREE-BRANCH STYLE and choose a Radiation tree or a Circle tree; can you see how they show the same thing?

Diagram

Description automatically generated with medium confidenceDiagram

Description automatically generated