Phylogeny Tutorial:

Getting sequences – this example is for Daucus carota. I looked around and found that the rbcL (RuBisCO large subunit) gene is good for phylogeny, and that a segment from the 1st half of this gene is a candidate for the plant "barcode", so this probably suits our needs. The NCBI database is frequently updated, so the results may change slightly, but it should follow for the most part.

1. Google NCBI and go the main page.
2. At the top, change the search menu to "nucleotide". If you search all databases, you can still get to nucleotides, but you will also see the other results, which might be worth a look to acquaint yourself with NCBI
3. Search the organism name and gene (e.g., Daucus carota rbcL).
4. To the right of the result list, chose the correct organism from "Top Organisms".
5. Note that two results are for complete (chloroplast) genomes – ignore these – they are way too much trouble for our needs.
6. The remaining three results are all indeed for the correct gene, but one is 653bp and the other are >1000bp. The gene is actually about 1500 or 1600bp long, so none of these are complete sequences, but any of the 3 will work fine for what we are doing. For this example, I chose the cultivar Karodagosun just because it is the longer of the three (and I checked – these two long sequences are 99.9% similar – not a big deal for this early example). As you go through these searches for the other 9 plants, you will notice that there are lots of sequences for this gene that are about 600bp long. I looked around and it looks like this is the candidate region for the barcode. We will end up trimming the alignment to this length anyway, so we could just as easily pick the most reputable sequences of this length.
7. After you choose the cultivar result, there are a few things of note inside the entry. First, there is an accession number just below the title "GenBank: AF207689.1". This is an easy way to search for sequences – you can either search this number from the first search page or enter it into a BLAST search and it will take you directly here. Next, scan through the metadata and orient yourself with the sort of metadata that are entered for each sequence. This gives an indication of sequence quality or entry trustworthiness – but only so far. This one was published in Plant Cell Physiology – publication is generally a good sign but not always. The title indicates that there might be something funny about this entry, but it is fine for now. If you scroll down, you can see the nucleotide sequence, but it is worthless in this format. Scroll back to the top, and hit the <u>FASTA</u> link under the accession number.
8. This format is the most widely used for phylogeny work, and it is the format we will use. Files with .fasta extensions are essentially text files that phylogenetic software recognizes with a few special features. First is the carrot (>). This means that non-sequence information follows until a carriage return (hitting enter). After the return is the unbroken sequence.

9. At this point, open an empty Word document (mac users can use TextEdit), and copy and paste the entry starting with the ">" and including the entire sequence. It will look funny in a document with uneven line lengths, but that is not a problem. I am attaching a .txt document as a reference. Next (or you could do this when all 10+ sequences are in the document), trim the title line to just what you want to see at the branch tips of your tree – "AF207689_Daucus_carota". Use underscores instead of spaces to avoid truncation later in MEGA. All sequences should retain the accession and name.
10. Repeat this procedure for all 10 plants. Some will not have an rbcL sequence, and in these cases, I took a sequence for a close relative (Agrostis capillaries, Chrysanthemum maximum, Fragaria x ananassa). The Fragaria is a bit of a mystery, but it was the only complete Fragaria rbcL I could find. Maybe one of the intergenic spacers would work, but I suspect not. At least it works well for this example. NCBI also knows that Aster novae-angliae is now Symphyotrichum novae-angliae and will suggest such. Also, the only good rbcL sequence for Bromus inermis has a peculiar line of N's in the middle of the sequence. This is actually ok, because we will only end up using the beginning of the sequence. Clearly, if we wanted to use the whole region, this would not stand, but for now it is ok. And Solidago and Oenothera both have only short (<800bp) sequences as the only rbcL. Since these coincide with the informative barcode region, these are ok.
11. After all 10 taxa are represented, we need at least one outgroup sequence. An outgroup gives a phylogenetic tree something to work against – or a solid root that is very different but similar enough to be able to align well. Here I used the rbcL gene from an alga – Scytosiphon lomentaria. Add this to your document, and before you move on, make sure it all looks appropriate: You will know it soon if not.
12. Now save your document as a plain text. Word for mac will default to OS Western text format – this is correct. Sometimes PCs will default to ASCII text format – this will not work. Manually chose OS Western. If you use TextEdit, you can save as a rich text and change the extension manually to .txt. I always try to keep this txt file untouched when it is finished with the right sequences, so put some indication of this in the title (e.g. RAW or something). You will want to keep one copy in the .txt format and make an exact copy to convert to .fasta.
13. Make a copy of this file (on your desktop if that is where you are working), and manually change the extension from .txt to .fasta.

Alignment in MEGA

1. I will use mac beta instructions and try to mention differences in the PC v4 instructions in parentheses, but they are really similar. First, open MEGA and hit the align icon → edit/build alignment →create new → DNA sequences

(alignment → alignment explorer/CLUSTAL → create new → DNA sequences)

2. This opens the alignment editor. Now open your .fasta file with the open icon. You may need to change the drop down to all files or FASTA files to see it.
3. Now inspect your sequences to see that all are there and nothing is obviously wrong (like an empty sequence…). If it doesn't look right or refuses to open, go back and look at the .txt file to make sure it is formatted correctly.
4. Now for alignment. MEGA offers several options, and we will use CLUSTALW for this example. CLUSTAL is actually not a very good alignment algorithm, but it is a bit like Churchill's quote "It has been said that democracy is the worst form of government except all the others that have been tried"… Actually it works just find for what we are doing, but not much past this. It is really quick and reasonably accurate. If we were doing anything related to the actual protein, this would be a bad choice. The mac MEGA offers MUSCLE alignment algorithm, but it takes a while and might be overkill for our purpose. You should try it to see if your trees turn out any differently. To run CLUSTALW, hit the W icon → DNA → select all. There are many options for running an alignment algorithm. The defaults are set well for almost all applications. The IUB is nice because it accounts for that line of N's in one of the sequences. Feel free to play with these and see the results, but it is pretty hard to improve an alignment unless you have a good reason to tweak the settings. This is handier for protein sequences than for DNA. This will take a few moments and spit out an aligned block. Hit escape to get rid of the highlighting.
5. Now scroll through the full alignment and see what happened. A couple of things should pop out. First, they are very different lengths, and they all cover only the beginning region of the sequence. Also, as you scroll through notice how different the algal sequence is from all others. This is great for us, but notice how some discrete spots are highly conserved in all sequences. Since this is an important protein coding sequence, we can probably assume that these are the important active sites of the protein, while others are free to diverge.
6. Depending on what we do with this alignment, we want the sequences to be the same length, otherwise it will throw off the tree. To do this, we should trim both ends. Above the colored nucleotides are gray boxes with some consensus stars. Starting at the beginning of the alignment, use these gray boxes to select the entire area up to where Rudbeckia begins (at a C position). You can select the first box, then hold shift and hit the last box to highlight a block. When this first incomplete block is highlighted (like the

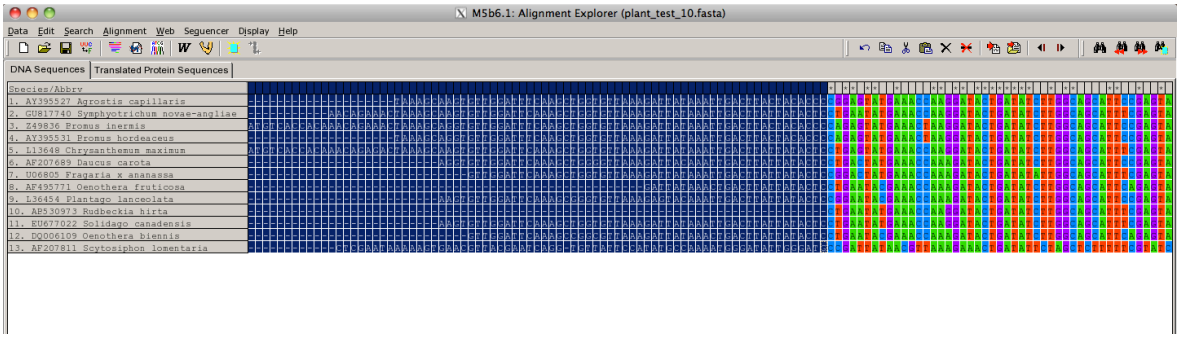

figure below), edit → delete.

7. Now do this again to the end of the blok starting at site 552 (where the N's start and near where the Solidago sequence ends. This will leave us with a 551bp set of sequences that nearly covers the 600bp barcode region.
8. Before making a tree, we should export this alignment in two different formats. First, data → export alignment → FASTA format. Name this and save it in case you want to look back at the alignment. Now do the same thing for the MEGA format, call it something, and tell it that it is NOT a protein coding sequence. Although this is a protein coding sequence, we cut it up and don't want that influencing the tree.
9. You will want to leave this alignment open until we start the tree if you are using a PC as there is a direct path to the next step. You don't need to save the session if you already exported the alignment as .fasta. Note that it will save as either .fas or .fa, but you can always change one to another and back manually with no problem. They are essentially the same format, but different programs see different extensions.

Make A Tree

It is worth saying a few things about trees before making them. Most algorithms for trees work in similar ways and generally come up with similar results. Neighbor Joining takes the two closest matching sequences and groups them, then adds each successively less similar sequence and groups based on dissimilarity. Parsimony trees follow the same principle as parsimony based statistics, and looks for the shortest path between two sequences and uses that path as a way to group sequences. Likelihood tests the probability that an initial tree topology is supported by the each column of aligned nucleotides, and so forth. They are good for different reasons, and for our example we will use NJ as it is the quickest and works well for nucleotide sequences that are pretty similar – as these are. All of these models basically work on the same 4 assumptions, and these are worth mentioning if not just for comedic purposes:

- Nucleotides are equally and randomly distributed within a sequence
- All nucleotides are equally likely to exhibit point mutation (or independent from all other nuc. sites)
- Genes change at constant rates within an organism
- The same gene in different organisms changes at uniform rates

Clearly these assumptions are all violated, especially for protein coding sequences for many different reasons, but somehow the methods are reasonably robust to violations, and more complex methods are only slightly better and only in some cases. For now, just know that we are violating all major assumptions, but for phylogeny, that is ok. These methods also assume that each nucleotide difference is the result of a single mutation (e.g., A→T), when in reality some have undergone multiple hits at a given site (e.g., A→G→T when all we can see is A→T). For this reason, NJ is especially bad for very divergent sequences, but this is not a problem for us. Another important thing to mention is the molecular evolution model choice. When we make a tree, we will have to choose between lots of them. Most are too

complex for our purpose, but two are most common and most reasonable for us: Jukes-Cantor model is a 1-parameter model, meaning that there is an equal likelihood that a given nucleotide will mutate into the other three nucleotides. From a biochemical perspective, this is a goofy assumption because transitions (purine→purine or pyrimadine→pyrimadine) are far more likely than transversions. For that reason, we often want to go with the Kimura 2-parameter model that assigns two different weighted penalties for the two different mutations. In reality, very little is often gained, but the K2P just feels better. There are many other models offering many different weighted parameters but improvements are very small compared to the computation involved. For our purposes, these two are best but feel free to play with them when it comes up.

1. If using mac, hit the phylogeny icon and chose neighbor joining tree. You might be asked if you want to use the currently open data, and if you have saved the right data, say yes. If not, chose your .meg file. For PC users, after you have exported your alignment as both .fas and .meg, close the alignment window. You will be asked if you want to open the data, chose yes. This opens a funky alignment data file. Then (in the main MEGA window) choose phylogeny→construct phylogeny→Neighbor Joining.
2. Now a window opens with several options for constructing a tree. Under the model option, there are the choices mentioned above, we will use K2P, but feel free to play with these to see what happens and if you can get a better tree. Under Gaps/missing data, we will use Complete deletion, but this one might also tweak the outcome. There is also a Test of Phylogeny option. With no test, we will get the first tree it comes up with. This might be fine, but if there are places on the tree that are tough decisions, you will generally want to go ahead and do some statistics. For this example, choose Bootstrap and 1000 replications. This is the most common way to do things, but there are other options. Bootstrapping is essentially choosing some sequences at random, making a tree and deciding if that tree (with only partial data) shares the same topology (tree shape). This happens 1000 times and the most common tree topology is called the consensus tree. When making a NJ tree, boot strapping is worth the small time it takes. Other methods take more time. When these options are settled, hit Compute. This shouldn't take much time, but other methods will take WAY more time, so be prepared for that.
3. When the tree pops up, there are a few things to notice. First, the bootstrap values will likely already be displayed on the internal nodes. If not, display them by hitting the hammer icon – this will lead to several different options for stuff to display. You want bootstrap values for the first look. The BS value is sort of like a confidence level – a statistic that tells you how likely each clade (group of sequences) is to actually contain exactly those sequences and not the other ones around it. Similar to statistical methods, you really only want to rely on those clades that are supported at >95. Especially for this tree, the low values you see will almost certainly improve if we added more sequences related to the loner tips of the branches. For

instance, if we had other things related to Plantago and Daucus, those would likely form good solid clades and the values would increase (notice that the tree cannot figure out where to put Plantago because it is just branched out of a funny place. For us, these don't really matter, but when we actually do a tree for publication, we will want to address all of these BS values to make the most well supported tree we can. Also notice that the outgroup (Scytosiphon) is clearly a good outgroup and is doing its job well. Published trees usually cut this out of the picture but mention what was used. Also, there are two tree options, an original and a consensus tree. They are explained pretty much as they are named. The consensus tree often has a better topology and maybe higher values, but not always.

4. Next, we can tweak the order of the tree with the icons on the left side. A tree is like a mobile and can easily be rearranged for clarity while retaining its identical topology. This one can be rearranged to look almost exactly like the original hand-drawn tree from these same plants. If we had more sequences to sort out the unsure tips, it would look identical.
5. With the hammer icon, you can remove BS values and add branch lengths for our analysis.
6. Notice the icons above the tree with different tree shapes. This tree displayed has uneven branch tips, and therefore is a scaled tree. We can make an unscaled tree with the tips in a nice column, but we lose the scale to see divergence. We probably want to stick with a scaled tree.
7. One of the best features of MEGA is the caption option that will give you a nice publishable summary of what you just did with citations for the methods used.
8. This tree can be saved as a .mts session file for work later, or exported as a .tif (PC version) or a .pdf (mac - .tif is a bug and is being worked out). Also, some journals want an eps file (this can be converted to a nice .pdf).