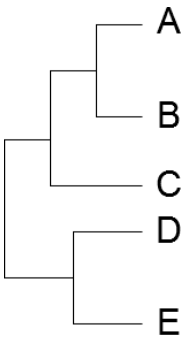
Tree Intro

Ben Liebeskind

PROPERTIES OF TREES

Trees are acyclic graphs that represent the relatedness or non-independence between samples. They have two components: branches and nodes (respectively called edges and vertices by CS folks). The tips of the tree are special types of nodes that only connect to one branch and don't have any child nodes. All the interior nodes of the tree have child nodes (usually two, if the tree is bifurcating), and are connected to three branches. The root of the tree is also a special kind of node that I'll discuss below.

Here's a tree (1):



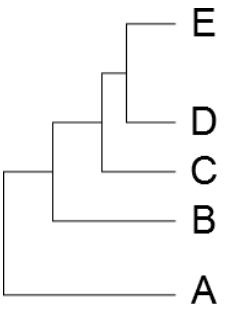
What does this tree tell us? It tells us the relative relatedness between A, B, C, D and E.
What doesn't this tree tell us? Here's a list of some common misperceptions, some of which occur even amongst accomplished evolutionary biologists:

1.) This tree does not tell us that A – E are "related." I could have put five completely randomized sequences in a tree estimation program and come up with this tree. On the other hand, all organisms, and perhaps all proteins too, are related if you go back far enough. So saying that A-E are related is either trivial or wrong, depending on how you look at it.

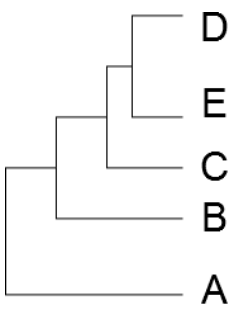
2.) It does not tell us that D and E are "ancient" or "basal" or "ancestors to A-C." This is a very common mistake that occurs for two reasons. One is that people believe that some organisms (or sequences) are "living fossils" that are exactly like the ancestors. This is not true. All extant sequences and organisms have been evolving for the exact same amount of time and none, with the exception of overlapping generations, are the ancestors of any others. True, some species have changed more than others, but you can't assume this going in.

The second reason is trickier. People often assume that all trees tell the direction of *time*, but in fact most do not. You actually have to use extraneous information to make a tree reflect time. To see why this is, let me re-root the tree.

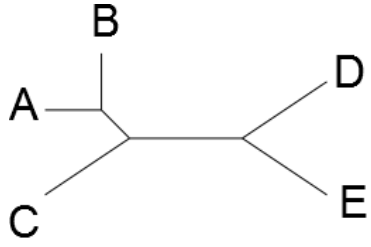
(2) Ta-da! Same tree



(3) This is also the same tree

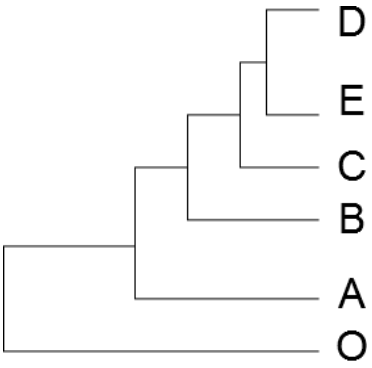


(4) So is this – this one is unrooted



The only difference between tree1 and tree 2 is that I put 2's root on the old "A" branch. The only difference between the 2 and 3 is that I switched E and D. In fact, I can rotate around any node, and the tree contains the exact same information. I can also re-root on any node if I want. Just imagine pinching any of the branches on tree 4. This only changes the information if I'm interpreting the tree as telling time.

If I want to figure out the branching order of A-D, the typical way is to use an outgroup: this is another taxon or sequence that I *already know* is less related to all the tips A-D than they are to one another. Then, I may see something like this:



This tells me that the proper branching order is like the third tree above. There's no indicator on the tree that "O" is an outgroup – it has be pointed out by the author, so never assume that a given tree has been properly rooted. Also remember that this rooting relies entirely on the assumption that "O" is a true outgroup, and those are sometimes hard to identify *a priori*.

One other point: The length of the branches may or may not be meaningful in a phylogeny. If all the tips are aligned, as above, then there are two possibilities: the tree may represent real time, and the tips are aligned because all the samples are from extant specimens, or the branch lengths may have no meaning and the tips are aligned for convenience. The vertical lines never have any meaning.

TREE INFERENCE

This is probably what you care about. There are several ways to infer trees, and whole subfields devoted to nothing other than figuring out what methods work and under what conditions. I will only discuss methods for inferring trees from molecular sequence data. Also, because there is enough information to support a whole scientific field, I will mostly confine myself to saying what works and what doesn't for common phylogenetic datasets. But that doesn't mean you shouldn't look some of this stuff up yourself.

First Things:

You will be dealing with many from-the-can softwares below, all of which require and spit out weird formats you may not be familiar with. To deal with this, you *must* use a plain text editor. If you're on Mac, try TextWrangler. If you're on PC, try Notepad++. Microsoft Word or other document editors will not work because they will add all kinds of weird formatting symbols in the background and will drive you nuts. Also, it's handy to be able to easily transfer between formats. The software ReadSeq will help (http://www.ebi.ac.uk/Tools/sfc/readseq/). Basically, there are a few main formats to know: Fasta, Nexus, Stockholm, Phylip, and Newick. I won't describe them here – they're not hard to understand and you'll learn soon enough.

I've tried to make everything below easily doable with online tools and simple GUI software. However, programming knowledge is a big plus in this field and will help immensely with getting your text files cleaned up and organized.

Finding Sequences:

This is the necessary first step. Because it will define what your analysis tells you, it's worth putting some thought into it. Here are some questions you may want to ask with a gene-family phylogeny (I'm ignoring species trees and high-throughput analyses where you want to analyze >10 different gene families):

1.) How old is the gene family (i.e. orthologs *and* paralogs)?
2.) How is my gene family related to other gene families?
3.) What is the history of gene duplication events, and which species have the closest orthologs?
4.) When did interesting changes occur in the sequence?

Often, you'll want to answer all four of these questions. The best thing to do, in my opinion, is to cast a wide net to start with. That way, you get your gene family of interest and any other closely related families – you can always prune later. A good way to do this is to catch everything that shares a PFAM domain with your sequence of interest. So take your protein and search for its model in the PFAM database (http://pfam.xfam.org/) using the tab "sequence search." Then go to the tab on the left called "Curation and Model" and, at the bottom, download the model as a text file (not in Word!).

Next, find all the proteins that match that model at some fairly stringent significance threshold (say 10^{-10}) using the HMMER website (http://hmmer.janelia.org/) – the algorithm you're looking for is hmmsearch; just paste your PFAM model right in the window. Now, you may not want *every* matching protein in Uniprot, so think about paring down your search space with one Uniprots "representative sets." These use similarity threshold to weed out "redundant" sequences so you can get a nice wide sampling of sequence space without downloading thousands of proteins. The smallest representative set is called Rp15. Try that to start with, but you may want to weed out even further (do you really need rat *and* mouse sequences?). You'll have to program this or do it by hand, sorry.

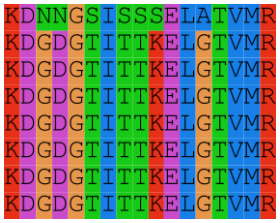
There are other routes to go here. You could use one of any number of databases that group sequences into "orthologous groups." A few of these are OMA, PhylomeDB, InParanoid, and OrthoMCL. If you use these, I strongly suggest making sure that your goals for your phylogeny are in line with the assumptions these databases make. For instance, sequences outside the orthologous group may still have detectable homology.

Alignment:

There are lots of multiple alignment algorithms (aligning more than two sequences at once). Strangely, they almost all involve inferring a tree first – phylogenetics is circular!! We prefer to say that it's iterative, which applies to most experimental methods. There are some algorithms that co-estimate alignments and trees, but I they're pretty new and I won't cover them here. If you don't plan on getting into fancy alignment methods, the basic story is this: use Mafft, not Clustal. Many people use Clustal because it's available in lots of GUI programs, but if command line is a limitation for you, Mafft's online interface is really just as easy (http://mafft.cbrc.jp/alignment/server/). Check out the different algorithm options that Mafft has below the paste-in area. I'd say use L-ins-i or E-ins-i, which are slower but more accurate.

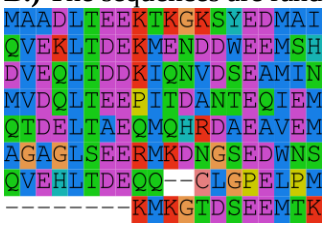
Once you've aligned your sequences, how do you tell it's a good alignment? Sadly, this is not so easy, but there are two situations where you *know* you're in trouble:

A.) All the sequences are basically identical.



In this situation, there isn't any information about which sequences are closer to any others, except for the first sequence, which is clearly more distant.

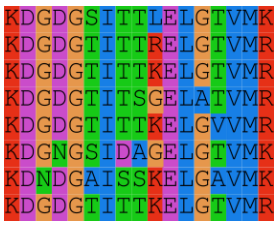
B.) The sequences are randomized, and there are lots of gaps.



In this situation, it is very difficult to infer the correct history because you have to infer many changes for each branch.

It's unlikely you'll see either situation so clearly, but if you're getting close to it, think about other ways to analyze your data. For instance, if you're using amino acids, but you have lots of identical positions, use nucleotides. If you do use nucleotides, you should think about forcing the codons to align by referencing the amino acid sequences. The alignment program Prank will do this for you (http://www.ebi.ac.uk/goldman-srv/prank/prank/). You can also use pal2nal (http://www.bork.embl.de/pal2nal/). This is absolutely necessary for inferring positive selection from codon models, which I won't cover here.

C.) This is what a nice alignment looks like. Some sites are more variable than others, and there's a nice mix.



Most likely, you'll have regions of the alignment that look like A, B, and C. In fact, I took all three of these pictures are from the same alignment. If that's true, unless only a tiny region looks like C, go ahead and try to make a phylogeny, but you might want to think about trimming ugly regions like B.

There are also some methods for getting rid of "poorly aligned" columns. There's no way to know for sure and in advance what a poorly aligned column is, but these methods are still useful. I suggesting looking at the programs Trimal, GUIDANCE, and BMGE, and seeing what they do. To make these pictures, I used an alignment editor called SeaView, which you can use to manually remove columns that look ugly. Whatever you do, make sure you keep track of it and report it.

Inferring a Tree (at last!):

There are lots and lots of papers that test different methods for inferring trees, but I'll just focus on the parts that are essential for someone that just wants a tree they can trust. There are three types of tree inference algorithms that I'll discuss: distance methods, maximum likelihood (ML), and Bayesian inference.

Distance methods, such as neighbor-joining, are the Clustal of tree inference algorithms: they're widely available, fast, and often problematic. If you want to get into the reasons why, there's plenty of literature, but the basic fact is that you just shouldn't trust the tree programs that come in GUI packages. This is especially true now that fast ML methods are easily available to the non-expert. Essentially what these programs are doing is inferring all-by-all distances using some method or other, and then finding a tree that best matches these distances. The problem is that all-by-all distances are "flat" clustering, and trees are hierarchical clustering, so no tree will perfectly match the distance matrix and the method you use to find the best approximation turns out to strongly affect that answer you get.

ML methods use a model of evolution to calculate a probability of the alignment given a certain tree topology. The space of possible topologies is then searched to find the topology that maximizes this likelihood. This can be computationally intensive, but computers have caught up to these types of calculations, if you're just inferring trees for one gene family at a time, there's no good reason not to use ML. I suggest using Garli (http://molecularevolution.org/software/phylogenetics/garli/garli_create_job) which was developed at UT, or PhyML (http://www.atgc-montpellier.fr/phyml/). Make sure you check out which alignment format they need, and convert if needed using ReadSeq.

You can also use Bayesian statistics to find a tree. This is similar to ML, but instead of maximizing the likelihood, Bayesian methods search tree space using either MCMC or Gibbs sampling and return a posterior distribution of tree topologies and model parameter values, which you can then summarize in various ways. I suggest using Mr. Bayes (http://mrbayes.sourceforge.net/) or PhyloBayes (http://megasun.bch.umontreal.ca/People/lartillot/www/index.htm). Bayesian methods are usually too computationally expensive to use webservers, so you'll have to download them. If you go this route, think about using AWTY to check for convergence in your sampling runs (http://king2.scs.fsu.edu/CEBProjects/awty/awty_start.php).

Assessing Tree Support

As I mentioned above, I could take completely randomized sequences, align them, and make a tree, and I could probably even find some way to squeeze a biological interpretation out of it. It's really not data until you can show that your tree is statistically well supported. Unfortunately, it's not really possible to put a P-value on a tree because the null distributions are not well understood. So you have two recourses.

If you're doing ML, you should definitely do multiple runs because the heuristic algorithms can get stuck in local optima. On top of that, you should also do bootstrap. This method samples columns from your alignment (with replacement) to make multiple pseudo-alignments that you will then find ML trees on. Usually people do 100-1000 bootstraps (I usually do 100 because I do big trees with protein data, which take longer to infer). You will then need to summarize your 100 bootstrap trees into a consensus tree with support values for each edge - these represent the number of bootstrap replicate trees that contain that edge. The Garli and PhyML websites should be able to do this.

The nice thing about Bayesian trees is they come with a built-in source of tree support: the frequency of an edge or other model parameter in the posterior distribution. As I said, this comes with a time trade-off. Mr. Bayes and PhyloBayes have their own ways of doing this which are beyond my scope here. You'll just have to check them out for yourselves, but I can tell you that their out-of-the-box summary methods are pretty good, so don't worry too much.

Visualizing Trees:

You presumably want to publish your tree, but the programs above will split out trees in plain text - probably either Newick or Nexus format. For instance, the tree (1) above would be represented: `((A,B),C),(D,E))`. You need a tree-viewing software to convert this into a nice looking tree. I recommend FigTree. You can re-root the tree, convert between square and circular views, color branches, and do all kinds of other nice things with FigTree (http://tree.bio.ed.ac.uk/software/figtree/).You'll then want to export to pdf, and do final editing in a nice vector-graphic editor, like Illustrator or Inkscape.

Learning More:

As I've said above, phylogeny inference is a whole field, so there's lots to learn. Here are some more resources:

Joe Felsenstein is a founder of likelihood-based phylogenetic methods. I recommend any paper by him. You can also get his book, but it's really more for the specialist (http://www.amazon.com/Inferring-Phylogenies-Joseph-Felsenstein/dp/0878931775).

David Hillis was one of my thesis advisors, and did a lot of the early work testing different methods. He also made empirical phylogenies using phage, together with Jim Bull. Check out those early papers for a good time. David also has a book - some of it is pretty outdated but it's worth looking at (http://www.sinauer.com/molecular-systematics.html).

Here are some other books worth looking at:
http://www.amazon.com/The-Phylogenetic-Handbook-Practical-Hypothesis/dp/0521730716
http://www.sinauer.com/phylogenetic-trees-made-easy-a-how-to-manual.html

I also can't resist a plug for the molecular evolution workshop at Woods Hole, which I've both taken and TA'd for: https://molevol.mbl.edu/wiki/index.php/Main_Page. You can check out the lectures on the course wiki for free.