How to infer phylogenies.

Breakin' it down — the rationale, the implementations, and the methods of molecular phylogenetic inference — a very brief overview.
A typical molecular phylogenetics protocol

1) Collect homologous sequences (lecture 2);
2) Multiple sequence alignment (lecture 3);*
3) Phylogeny estimation (today’s lecture &);*
4) Test the reliability of phylogenetic estimate(s);
5) Interpretation and application of phylogenies.

* Ideally, these two steps would be combined.
A multiple sequence alignment is a hypothesis of evolutionary history. Homology is way more than just anatomy, physiology, and genes. It extends to the smallest heritable unit of life, individual DNA base pairs. Even after homologous molecules are identified, it’s necessary to establish the correspondence between individual sequence positions. Therefore, make sure you have prepared a good multiple sequence alignment (MSA)!

All molecular sequence phylogenetic inference programs’ first and most critical assumption is the validity of your input alignments — meaningful results absolutely depend on their quality. Devoting considerable time and energy toward developing the best possible MSA is invaluable. Assure that known enzymatic, regulatory, and structural elements all align. Be sure that it makes biological sense. Use all available information and understanding to insure that your alignment is as good as it can be.
For today

1) After your MSA what?

2) How big is your dataset - implementation
   a) Exact routines: tiny - exhaustive; small - branch and bound.
   b) Everything else - heuristic or sampling technique.

3) Philosophy - which methods, models, or combination of methods and models should I use? Which loci and how many should I use? Should I analyze multiple loci separately or concatenated?
   a) Quick and dirty - algorithmic approaches - neighbor joining.
   b) Optimality criteria based:
      i) Least squares and minimum evolution distances;
      ii) Maximum parsimony;
      iii) Maximum likelihood; and . . .
      iv) Bayesian inference.

4) How confident are we of the results?
Phylogenetic methodology can be thought of in different ways.

- Algorithmic versus “optimality criteria” based . . .
- These are, respectively, those methods like neighbor joining that simply give one answer without knowing how good or bad it is, versus methods that try to find the “best” tree and offer scores to evaluate them.
- Within the optimality based methods we need to consider implementation versus method.
- By implementation I mean do we attempt to look at all of the trees and evaluate each one, or do we use some trick to do the same thing, or do we use a heuristic to only sample ‘promising’ trees, or do we use a semi-random sampling technique to get a feel for the ‘landscape?’
Implementations

Remember the problem: The number of possible unrooted, bifurcating trees for a dataset rises as a factorial of the number of sequences, thus, where $T$ is the number of individual sequences:

<table>
<thead>
<tr>
<th>Number of sequences</th>
<th>Number of possible bifurcating, unrooted trees</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>3</td>
</tr>
<tr>
<td>5</td>
<td>15</td>
</tr>
<tr>
<td>6</td>
<td>105</td>
</tr>
<tr>
<td>7</td>
<td>945</td>
</tr>
<tr>
<td>8</td>
<td>10,395</td>
</tr>
<tr>
<td>9</td>
<td>135,135</td>
</tr>
<tr>
<td>10</td>
<td>2,027,025</td>
</tr>
<tr>
<td>20</td>
<td>$&gt;2 \times 10^{20}$</td>
</tr>
<tr>
<td>50</td>
<td>$&gt;3 \times 10^{74}$ ($&gt;\Sigma$ of all the atoms in the universe!)</td>
</tr>
<tr>
<td>10 million</td>
<td>$&gt;5 \times 10^{68,667,340}$</td>
</tr>
</tbody>
</table>

$$B(T) = \prod_{i=3}^{T} (2i - 5) = \frac{(2T-5)!}{2^{T-3}(T-3)!}$$

Monday, February 15, 2010
Therefore . . .

1) Exact Methods — Exhaustive versus Branch-and-Bound

a) Exhaustive:

Add every sequence, stepwise, at every possible position. This is incredibly cpu intensive and is intractable for most datasets, only being practical for up to a dozen sequences or so.

b) Branch-and-Bound:

Start with the evaluation of one route, 'all the way out,' but only evaluate other routes with better scores than the starting route, and update that starting score as the search proceeds. This eliminates much of the search time, yet guarantees an optimal tree. However, even it is only practical for up to about forty sequences or so.
Exhaustive Searching

Every single route is taken, all the way out, and the best one wins.
Branch and Bound Searching

Only routes better than what have already been seen are explored.
So, normally exact implementations can’t be used. We need to use some sort of heuristics or sampling technique.

* Heuristic: Only take and evaluate the route of ‘lowest resistance.’ Analogous to ‘hill-climbing’ energetics algorithms used in molecular dynamics computations. Problem: getting ‘trapped’ on local rather than global maxima.

* Therefore, tree rearrangement algorithms are used in the programs and should always be taken advantage of.

* And, start at multiple points; i.e. take advantage of random input order options and repeat the analyses many times, at least ten.
The “blind parachutist” analogy helps:

* Imagine a squadron of severely myopic parachutists who all lose their glasses on their jump. Their collective goal is to find the highest peak in the area — they all have altimeters and portable two-way radios.

* They can’t see the surrounding peaks, but they can tell that they are walking uphill versus downhill, so they climb up. Once they reach the top of their respective peak they read their altimeter and communicate with one-another to collectively identify the highest peak. Ergo, send down lots of parachutists to cover lots of territory, i.e. repeat the analysis as many times as practical and, since the results are order dependent, use several different starting orders.
Heuristic Searching

Tree rearrangement ‘moves’ – there are many – quartet “puzzling” is another.
And what's this about sampling?
Markov Chain Monte Carlo techniques

- The posterior probabilities of trees are complex joint probabilities that cannot be calculated analytically.
- Instead, the posterior probabilities of trees are approximated with Markov Chain Monte Carlo techniques that sample trees from their posterior probability distributions.

- The posterior probability distribution of trees is represented by the surface of the field, with its points visited in proportion to their elevations. The higher its elevation, the more frequently a point will be visited, therefore, the greater its posterior probability, and the more frequently a tree will be sampled.
Philosophy – which methods, models, or combination of methods and models should I use? Which loci and how many should I use? Should I analyze multiple loci separately or concatenated?

Let’s delay methods and models for just a minute to briefly mention these other general concerns.

You should sequence as many loci as your research budget and time will allow! A whole genome? Sure! But realistically more than one is always nice, and for population parameter estimation five is great.

Concatenation versus individual analysis is tricky. Maybe do both. It’s especially dangerous to use one consistent model across a combined dataset when that model is not appropriate for every loci. Some programs allow the model to vary along the length of the dataset.
OK, what about those models – Jukes/Cantor to GTR+$\gamma$+pinvar?

- All are time homogeneous – the substitution process remains constant across the tree, and...
- Time reversible – Probability of starting with i and ending with j equals the probability of starting with j and ending with i.
- Furthermore, evolutionary models are a nested family, with each being a subset of the other.
- Don’t worry about memorizing the particulars of each, just realize that each adds more parameters than its simpler subcase.
- And variable rates along the sequence are modeled with $\gamma$ and pinvar.
- Figure from Swofford et al., 1996.
- And also see [http://workshop.molecularevolution.org/resources/models/](http://workshop.molecularevolution.org/resources/models/)
So, how the hell do you know which model to use?

- Complexity versus simplicity:
- Use complex models to accommodate the complexities of evolution; and . . .
- Use simple models to avoid over-parameterizations and increased variances.
- Therefore, use the simplest model that works with your data.
Overparameterizing a model

Fortunately there are some very nice resources available that attempt to pick the best model for you based on robust statistical methodology. These use likelihood ratio tests, and Akaike and Bayesian information criterion to pick the best DNA or protein model for your dataset:

* [http://darwin.uvigo.es/software/jmodeltest.html](http://darwin.uvigo.es/software/jmodeltest.html)
* [http://darwin.uvigo.es/software/modeltest_server.html](http://darwin.uvigo.es/software/modeltest_server.html)
* [http://darwin.uvigo.es/software/protest.html](http://darwin.uvigo.es/software/protest.html)
* [http://darwin.uvigo.es/software/protest_server.html](http://darwin.uvigo.es/software/protest_server.html)

\[
y = -330 + 134x - 15.5x^2 + 0.816x^3
- 0.0225x^4 + 0.000335x^5
- 0.00000255x^6 + 0.00000000777x^7
\]

\[
(r^2 = 1.000)
\]
What about rate heterogeneity among sites?

- Proportion of invariable sites (pinvar). Plus...
- Rate variation among sites according to the gamma ($\gamma$) distribution.
- Usually modeled as a Hidden Markov Chain process with a discrete number of categories.
In short . . .

- Remember what the model is trying to do – recover ‘unseen’ substitutions – so that we can infer the amount of change, to estimate distances between sequences and/or rates of character change at a particular position. So . . .

- If change is sufficiently rare, i.e. if the sequences aren’t too diverged for the analysis at hand, none of the models will suffer from systematic error as a result of deviance from their implicit or explicit assumptions.

- The moral of the story: exclude those most highly diverged sequence regions from your analyses. It is in these areas of high homoplasy that all assumptions are most violated. Homoplasy is similarity of organs, other bodily structures, or, in this case, individual base pairs of DNA, between different species, not due to common ancestral origin and development, but rather due to independent evolutionary change: parallelism, reversal, or convergence. Thus, homoplasy is a mistaken hypothesis of homology that confounds phylogenetic analyses. Regions of high homoplasy correspond to areas where your alignment is least sure, the history is completely saturated. Therefore, get rid of ‘em.
And now methods!

* When you don’t have time and you just want a ‘quick and dirty’ answer as a preliminary step to further analyses (but not something to publish), then . . .

* Algorithmic methods are the way to go. And the ‘tried and true’ one is still . . .

* Neighbor joining (NJ), and it’s ultra fast!

* You still need to decide on a model, and then calculate all pairwise distances. Then an additive tree is assumed, but different branches are allowed to have different rates (i.e. it’s cluster analysis, but not UPGMA [Unweighted Pair-Group Method with Arithmetic Means]). The NJ algorithm is a special case of star decomposition that finds neighbors sequentially that may minimize the total length of the tree. Many programs are available.
And the rest of the time . . .

* Use an optimality criteria – but which one?
  * Efficiency
    * How fast is a method?
  * Power
    * How much data does the method need to produce a reasonable result?
  * Robustness
    * Will minor violations of the method’s assumptions result in poor estimates of phylogeny?
  * Falsifiability
    * Will the method tell us when its assumptions are violated, (i.e. that we should not be using the method at all)?
In turn, we’ll look at . . .

1) The major other distance based, but optimality criteria driven methods – least squares fit and minimum evolution – where we use an evolutionary model to estimate the distances between all sequences;

2) Maximum parsimony, where we use the characters themselves, not the evolutionary distances between the sequences, minimizing the number of evolutionary events along each branch – the implicit model is minimum change;

3) And maximum likelihood and Bayesian inference, in which we use all the characters and an explicit evolutionary model to effectively combine the best features of both parsimony and distance based methods.
Least squares fit or minimum evolution to an additive tree . . .

* Are pretty much the same . . .

* Given pairwise distance estimates, out of possible trees, find the tree(s) and branch lengths that best explain these estimates, that is that minimizes the difference between the observed pairwise distances and path-length distances calculated on a tree (i.e., the sum of branch lengths between a pair of external nodes) using the least squares method (similar to finding the best line that fits a scattered point dataset) or merely by minimizing those paths lengths. Variations on . . .

\[ E = \sum_{i=1}^{T-1} \sum_{j=i+1}^{T} wij |d_{ij} - p_{ij}|^\alpha \]

* If you need to root a distance tree, use only one, most closely related, outgroup to decrease the amount of implicit, systematic error in the method’s assumptions.
Historically these ideas arose in the analysis of complex morphological features. It’s intuitive to assume these were only invented once. Therefore, all species in which the feature is found, a synapomorphy, should share a common ancestor with that feature. This means the evolutionary history requiring the minimum amount of reinvention is preferred. Mathematically, this is equivalent to the minimum amount of invention. This philosophy is the core of phylogenetic "cladistics."

Therefore, the optimality criterion is to minimize the parsimony score, and hence the implied amount of evolutionary change, i.e. the number of events (steps, perhaps weighted steps) required by a tree, to explain the variation in the data.

We assume character independence; therefore, we can calculate the length required by each character, and then sum over all characters to get the total tree length.

Although proponents pontificate its lack of models, it has a very real implicit model – evolution proceeds by the least amount of change, i.e. the slowest rate – and this is also its greatest weakness.
The method of inferring the most parsimonious tree is purely computational. It is relatively easy to find out how good (or bad) a given tree is, but it is not so simple to find the best tree. In this case the best tree is that tree with the shortest overall branches, i.e. the smallest sum of all of its branch lengths. This is called the shortest “tree length.”

- “Brute force” would enumerate every possible ancestral state reconstruction, and count the total cost required for each of these reconstructions, then sum over all characters, picking the smallest route, but this is computationally pretty intense, so many tricks . . .

- Enable us to reduce the complexity of the operation; e.g. dynamic programming can be used to calculate the shortest tree length without enumerating every path, just like in alignment.

- And in generalized parsimony the user specifies the cost of each type of change, usually upweighting changes between a purine (A or G) and a pyrimidine (C or T) (“transversions”), and downweighting changes between two purines or between two pyrimidines (“transitions”).
Here's an example for just one data column, using equal weighting and a transversion bias of 4 to 1.
But, there's a danger, parsimony runs a very real risk of...

Confusing identical characters due to multiple changes (reinvention) with those that are the same due to conservation. This problem is particularly evident with long tree branches. Given 'long' enough branches, this 'long-branch-attraction' problem, often referred to as being in the 'Felsenstein Zone,' guarantees that standard parsimony will always infer the wrong tree.
Accuracy of methods with simulated data on a moderate Felsenstein zone four-taxa tree

\[ \alpha = 1.0, \ p_{\text{inv}} = 0.5 \]
Minimize this parsimony problem by:

1) Avoiding very diverged sequences, those which have the longest branches; by . . .

2) Avoiding the most variable sequence positions, which will have the greatest chance of multiple substitutions; and by . . .

3) Including more than one representative of a distant group, which subdivides the long branch into shorter branches (do not do this with distance methods — it can compound problems).

* Is there an alternative then? Yes, there are probabilistic statistical techniques available that minimize all these problems — maximum likelihood and Bayesian inference!
Maximum likelihood and Bayesian inference yield a statistical ranking of trees; they . . .

Attempt to trace the evolution of each sequence position through an assumed phylogenetic tree, subject to the expected amount of change implied by the tree branch lengths — it chooses those trees that maximize the probability of observing the data. The optimality criteria is evolutionary probability.

In effect, they bridge distance and parsimony methods, and can provide the ‘best’ answers to most problems since they combine the strongest points of the other methods. The mathematics are quite complicated. However, given ‘enough’ data, an appropriate model of evolution, and long enough run time they will always find the correct tree, even in those cases where parsimony will positively fail. Another advantage is they can estimate the most probable evolutionary models to use.
The likelihood of a tree is the probability of the data given a tree and a model.

\[
\text{Pr}(\text{data} | \text{hypothesis}) \propto \text{Prob(data | hypothesis)}
\]

Likelihood(tree, model) = \(k \times \text{Prob(}\text{observed sequences | tree, model})\) [not the \(\text{Prob(tree | data, model)}\)]

This is known as the Likelihood of the tree. One method of reconstructing the evolutionary history is then to find the tree that has the Maximum Likelihood.
Looking at just two characters for four taxa

\[ \mathcal{L}(T) = \Pr\{D \mid T, Q\} \]

The probability of the data, \( \Pr\{D \mid T, Q\} \) can be efficiently calculated given a phylogenetic tree (T), and a probabilistic model of molecular evolution (Q).

In statistical phylogenetics, branch lengths are traditionally unconstrained.

* Felsenstein's likelihood (1981)

\[ \mathcal{L}(T) = \Pr\{D \mid T, Q\} \]

The probability of the data, \( \Pr\{D \mid T, Q\} \) can be efficiently calculated given a phylogenetic tree (T), and a probabilistic model of molecular evolution (Q).

In statistical phylogenetics, branch lengths are traditionally unconstrained.
How’s it done? Let’s look at just one data column.

* Computing the likelihood of a single tree

1  j  N

(1) C...GGACA...C...GTTTA...C
(2) C...AGACA...C...CTCTA...C
(3) C...GGATA...A...GTTAA...C
(4) C...GGATA...G...CCTAG...C
Computing the likelihood of a single tree

Likelihood at site $j = i.e:$

$$L = L_1 L_2 \ldots L_N = \prod_{j=1}^{N} L_j$$

$$\ln L = \ln L_1 + \ln L_2 + \ldots + \ln L_N = \sum_{j=1}^{N} \ln L_j$$

*But Felsenstein’s (1981) pruning algorithm avoids much of the computation.*
Finding the maximum-likelihood tree (in principle)

- Evaluate the likelihood of each possible tree (defined by the heuristic search strategy being used) for a given collection of taxa.
- Choose the tree topology which maximizes the likelihood over all possible trees, given a model.
- And, of course, branch lengths matter!

The "oscilloscope" analogy — imagine a grand device where every parameter of a tree (the topology, and the branch lengths) can be tweaked until the maximum of this multidimensional landscape can be found.
Drawback to maximum likelihood

- It is incredibly computationally intensive, especially if you use it to estimate models and trees at the same time, though there are lots of tricks to speed things up. A major one is estimate the best model separately, then fix the model. However, as computers become faster and faster, and heuristic programs get better and better, it has become the method of choice among most phylogeneticists...

- With the exception of Bayesian techniques (though these still scare some people)!
Bayesian techniques

What we really want is the probability of the tree given the data. We can compute that from the likelihood using Bayes Theorem:

\[
P(\text{tree} | \text{data}) = \frac{\text{Likelihood} \times \text{Prior Probability}}{\text{Normalization constant}}
\]

This is known as the Posterior probability of the tree. Another method of reconstructing the evolutionary history is then to find the tree that has the Maximum Posterior probability.

So Bayesian methods are also a probabilistic, but they use a Monte Carlo sampling rationale to sample amongst thousands of possible trees, summing the information afterwards to provide posterior probabilities of which tree nodes are most supported.
This is necessary because the math gets even nastier!

* Posterior Probability of a Tree:

\[ P(\tau_i \mid X) = \frac{P(X \mid \tau_i)P(\tau_i)}{\sum_{j=1}^{B(t)} P(X \mid \tau_j)P(\tau_j)} \]

* \( P(\tau_i \mid X) = \) posterior probability of tree i (\( \tau_i \)) given data X
* \( P(X \mid \tau_i) = \) likelihood for tree i *
* \( P(\tau_i) = \) prior probability of tree i
* Denominator = unconditional probability of data X; calculated across all \( B(t) \) trees
* * Calculated under same evolutionary models as before
And it uses integrated likelihoods!

\[ P(X|\tau_i) = \int_{v_i, \theta} P(X|\tau_i, v_i, \theta) p(v_i, \theta) dv_i d\theta \]

* \( P(X|\tau_i) = \) likelihood for tree \( i \), obtained by integrating over all combinations of branch lengths (\( v_i \)) and substitution model parameters (\( \theta \))

* \( P(X|\tau_i, v_i, \theta) = \) probability of data \( X \) given tree \( i \), branch lengths \( v_i \) and model parameters \( \theta \)

* \( p(v_i), p(\theta) = \) prior probability densities of \( v_i \) and \( \theta \)

* \( dv_i, d\theta = \) infinitesimal intervals of \( v_i \) and \( \theta \)
The space of all possible trees can be visualized as a hilly landscape. Nearby points in this landscape represent similar trees, and the height of the landscape is the probability of the tree at that point.

This space can be sampled in a Bayesian analysis with MCMC (Metropolis Coupled Monte Carlo).
The posterior probability distribution of trees is the area under the surface of the field.

- The output of a Bayesian evolutionary analysis is a probability distribution of trees and parameter values.
- For phylogenetics the tree topology and branch lengths are the objects of interest. The substitution parameters and tree prior parameters are a nuisance that we average over using MCMC and then ignore.
Putting it all together

* Start with random tree and arbitrary initial values for branch lengths and model parameters (or not).

* Each generation consists of one of these (chosen at random):
  * Propose a new tree (e.g. Larget-Simon move) and either accept or reject the move.
  * Propose (and either accept or reject) a new model parameter value.

* Every $k$ generations, save tree topology, branch lengths and all model parameters (i.e. sample the chain)

* After $n$ generations, summarize sample using histograms, means, credible intervals, etc.

* The success of Bayesian techniques largely depends on . . .

* How long you let your MCMC chains run!

* But also on models and the priors for each parameter of the model.
And what about confidence?

* Bayesian inference comes with built-in node support values. These are confidence limits in the Bayesian context.
* Everybody else needs an external assessment method. The most common is the non-parametric bootstrap.

Original data set

 pseudo rep 1

 pseudo rep n

Monday, February 15, 2010
The bootstrap

* Bootstrap resampling provides a loose confidence limit on the ‘cohesiveness’ of groups separated by given tree branches. It achieves this by randomly selecting character columns from the original dataset, with replacement, up to the same number of characters as the original dataset, creating a set number of ‘pseudoreplicate’ datasets. You then perform whichever inference algorithm you want to use on each of the bootstrapped datasets, usually at least 100, better yet a 1000. A consensus tree is calculated from all the different pseudoreplicate trees, and support values are printed for each node (i.e. How often is a particular node found among all the bootstrapped trees?).

* A ‘rule of thumb’ often used is, those nodes with bootstrap values of more than ~60% are probably better than 60% accurate and those nodes with bootstrap values worse than ~40% are way worse than 40%. This is why one seldom sees bootstrap values published less than 50%.
Acknowledgements

Without these folk I would not be able to teach molecular phylogenetics!

They have provided ideas, graphics, and entire presentations for my use over the years. In no particular order:

- Gary Olsen, University of Illinois;
- Michael Miyamoto, University of Florida;
- Jim Wilgenbusch, Florida State University;
- David Swofford, Duke University;
- Paul Lewis, University of Connecticut;
- Joe Felsenstein, University of Washington;
- Alexei Drummond, University of Auckland.
And don’t forget . . .

Next time we have Peter Beerli from Florida State University to discuss the estimation of population dynamics parameters using molecular phylogenetic ‘style’ techniques.