**Phylogenetic Inference,**

**Construction of Phylogenetic Trees**

**Basics Steps**

1. Construct a data matrix (e.g., DNA sequences, characters, etc)
2. Identify trees that are more compatible with the data than other trees.
3. Conduct statistical analyses to determine how confident you should be in the phylogenetic conclusions

**1. Constructing a Data Matrix**

1. Choosing Taxa
2. Choosing Characters
3. Assembling Data Matrix (Includes Sequence Alignment)

**1.1 Taxa**

In most cases one cannot possibly include all living forms. So, often the taxa depend on what question you want to ask. If you want to ask if aquatic pinnipeds (e.g., seals, sea lions, walruses) form a monophyletic group relative to all other mammals, then you don’t need to include every living cat species. However, you also can’t just include two seals and answer this question.

You also need to choose an **outgroup**. Outgroups serve as a point of comparison to the ingroup taxa, allowing us to root the tree and determine direction of character change (e.g., did the ancestors of the aquatic pinnipeds live near water). The best outgroups are reasonably closely related to the ingroup. You need to be able to compare characters easily. For example, with DNA and protein sequences, you need to be able to align the sequences from the outgroup to the ingroup.

Some guidelines for choosing outgroup taxa:

1. Should include multiple taxa that are closely related to your ingroup
2. You should be certain that at least some of your outgroup taxa are clearly outside the ingroup.
3. You must be able to find homologous characters across ingroup and outgroup taxa (see below for more information on characters and homology).

**1.2 Characters**

A character is a feature of an organism that can be described and usually varies between species. A character state is a form of a variable feature (e.g., A,G,C,T for a DNA sequence position).

* It is important that characters be homologous. Why?

***Molecular vs. Morphological***

If you are planning on doing phylogenetic analysis, it is most likely that you will be using molecular sequence data rather than morphological data. The benefit of using morphological data is that it can allow for inclusion of fossil data. Fossil data can be useful for reconstructing ancient relationships. We can also often date fossils, making it possible to more accurately place dates on trees. So, rather than saying, this split happened earlier than this split, we can say, this split occurred approximately 50 million years ago.

***Types of Molecular Data***

We tend to use nucleotide sequence data and protein sequence data. Reconstruction of phylogenies using amino acid sequence alignments rather than nucleotide sequence alignments is often used when trying to reconstruct ancient relationships.

With the advent of genomics, people are developing new methods that can incorporate gene presence/absence across a genome and gene order.

***What Genes? How Many Genes?***

The history of descent of a gene may not be identical to the history of descent of a species. This is particularly true in bacteria, for example, where lateral gene transfer leads to the transmission of genes between populations and species. This is also true of other species where lateral gene transfer is not as common. Processes such as incomplete lineage sorting, introgression and gene duplication can all cause gene trees to not match species trees (see handout from previous day). This leads to genealogical discordance, where different genes sampled from the same set of taxa lead to different trees. This is the main reason that sequencing more genes can be better. How many genes is enough? There is no clear answer to this. Look at the best standards in your field given the taxa that you are working with and the questions that you want to address.

* What are some advantages and disadvantages of using morphological data? What are some advantages and disadvantages of using molecular data?
* Researchers often talk about how quickly or slowly a gene evolves when trying to decide what gene to use in an analysis? Why does this matter? When would one want a fast evolving gene? When would one want a slowly evolving gene?
* A researcher decides to use four mitochondrial genes in a molecular phylogenetic analysis? Each of the genes separately leads to the construction of the same tree topology? Why is this expected? Was four genes better than one?

**1.3 Assembling a Data Matrix**

For morphological characters, this means looking at specimens and scoring the data. You can have characters that are discrete (e.g, 0 = no wings, 1 = wings) or continuous (e.g., 1 toe, 2 toes or 3 toes on front foot). If there is intraspecific variation, proportions or averages can be used.

For sequence data, getting sequences typically involves one or more of the following:

* Sequencing a portion of a specific gene for the taxa of interest using traditional Sanger sequencing based methods.
* Generating a sample of sequences via next generation sequencing
* Retrieving sequences from NCBI or other databases.

Once you have the sequences, they must be **aligned.**

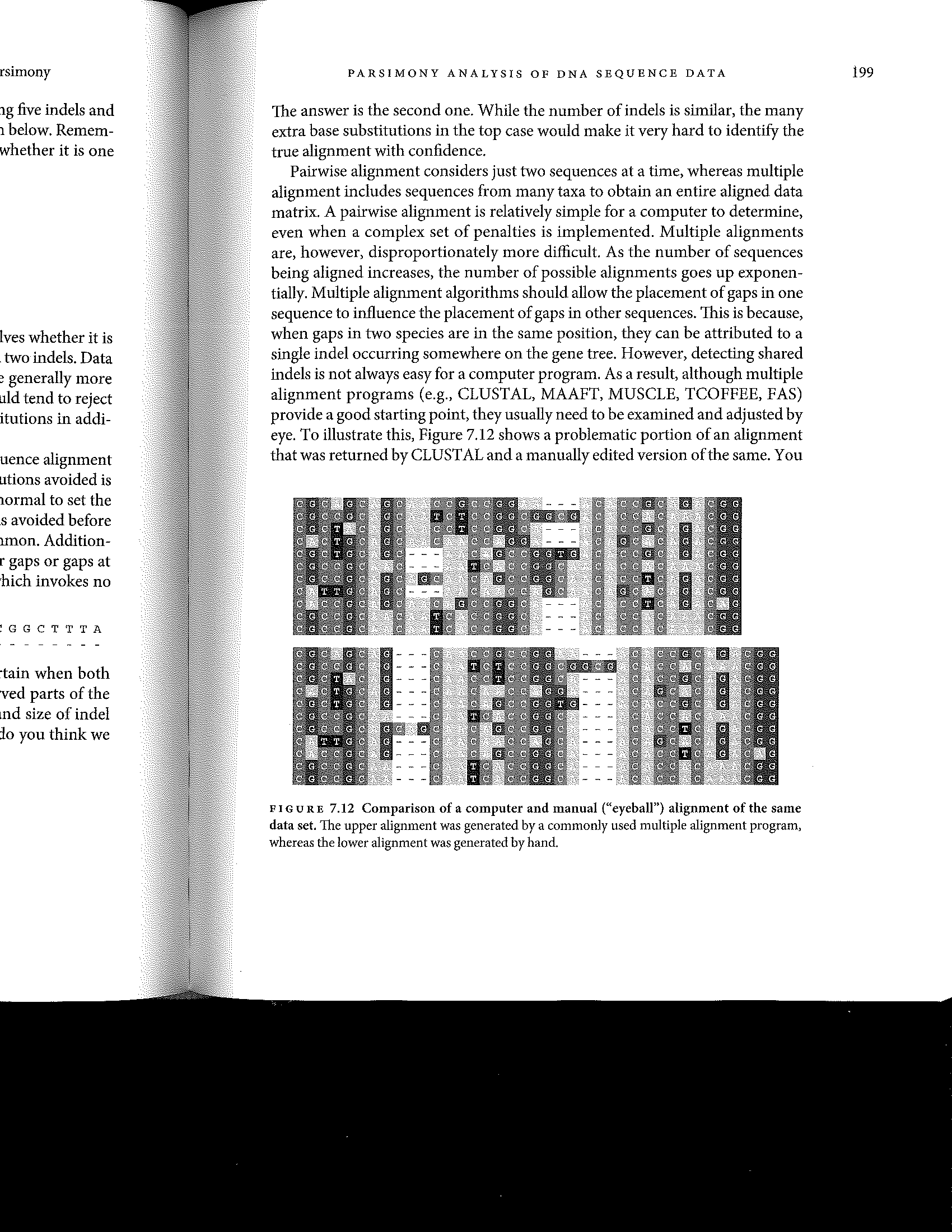
***Sequence Alignment***

Whether you sequence all of your own samples or download sequences from the internet, not all of your sequences will be the same length. Thus, the 1st base pair in one sequence is often not homologous to the 1st base pair in another sequence. Furthermore, because of insertions and deletions, the sequences will also differ in length. The goal of sequence alignment is to align sequences such that the nucleotide or amino acid position in one sequence is matched with the homologous position in other sequences. If the entire sequence of interest is highly conserved, then this is fairly straightforward. However, in most sequences, there are at least some regions that are evolving rapidly enough that this is quite challenging. In particular, some sequences have had base pairs added in some places, and some sequences have had deletions in some places, it can be nearly impossible to confidently align your sequences. In general, here is what you do:

1. Use a computational algorithm (e.g., Clustal W, Muscle) to align your sequences. For nucleotide sequences not associated with protein coding genes (e.g., ribosomal RNAs), you will align the nucleotide sequence. For nucleotide sequences associated with protein coding genes, you ultimately want the nucleotide sequence aligned, but you can benefit from using the translated amino acid sequence in the process. This can be facilitated with tools such as TranslatorX.

* Why can the amino acid translation aid in nucleotide sequence alignment? Will be more useful for highly conserved genes or variable genes?

1. Visually inspect the alignment. The dashes (see above) represent **gaps**, “placeholders” in the alignment to show hypothesized insertions and deletions (i.e., indels). In some cases, manual alignment can help refine areas. In cases where the alignment does not look clear cut (you can see these pretty easily as a mess of bases here and bases there), then you want to explore a range of **gap-opening penalities** (e.g., the cost of putting a new gap between characters in a sequence.



1. Exclude areas that cannot be aligned, regardless of what gap-opening penalty you use, from further analysis. To do this, if you using Mr. Bayes or Garli for example (see below), you can put a block at the end of your nexus file (the files you use for these programs) that says exclude characters XXX-XXX (e.g., characters 725-777). Then, when the analysis is running, it just ignores the bases at positions 725-777 of your alignment. This is critical. Be conservative.

Note, in any character matrix, you may have missing or ambiguous data. For example, a few bases in a sequence may be noisy or it may not be known if a particular extinct taxa had kidneys. The analyses can typically handle this. These missing data are often indicated with a question mark.

**2. Identifying Trees that are Most Compatible with the Data**

The most commonly used methods for phylogenetic inference are parsimony, distance methods (e.g., neighbor-joining), maximum likelihood and Bayesian analysis. Maximum likelihood and Bayesian methods are the customary methods used in most publications today. Distance methods are great if you need a fast but not precise method (e.g., you need to get a feel for the data when selecting additional taxa). We’ll briefly discuss parsimony and an original method (Hennigian) as these methods provide fundamental insights into the problem at hand.

**2.1 Hennigian Inference – The First Method of Phylogenetic Inference**

Henningian Inference, while no longer used, provides a good starting point for thinking about the challenges of phylogenetic inference. There are two critical assumptions:

1. There is a strictly treelike phylogenetic history
2. There is no **homoplasy.** Homoplasy is when two species of different ancestry share the same state because of convergent evolution (e.g., wings of bats and wings of birds). If there is no homoplasy, each character evolved from an ancestral state to its current state only once without subsequent reversal. If there is no homoplasy, characters cannot contradict each other in terms of the relationships that they support.

From a molecular perspective, this means there are no back mutations and no independent forward mutations. This is clearly not biologically accurate. Again, it’s a starting point. So, let’s look at this data on carnivores (data from *Tree Thinking*)*:*

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
| Outgroup | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| cat | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 |
| hyena | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 |
| civet | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| dog | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| racoon | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| bear | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 |
| otter | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| seal | 1 | 0 | 1 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 1 | 1 |
| walrus | 1 | 0 | 1 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 1 | 1 |
| sea lion | 1 | 0 | 1 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 |

A 0 represents that a discrete trait is absent and a 1 represents that a discrete trait is present. For example, for character 7, 0 means they have non-retractable claws and 1 means they have retractable claws.

We assume that the state of the outgroup is the ancestral state.

Now, we identify sets of taxa that share a derived character state and infer that they must form a clade. Character 4, for example, supports cat + hyena as a clade.

The main problem with this method is that HOMOPLASY HAPPENS, and, **there is no way to proceed if you find that characters contradict each other** because of homoplasy.

**2.2 Parsimony**

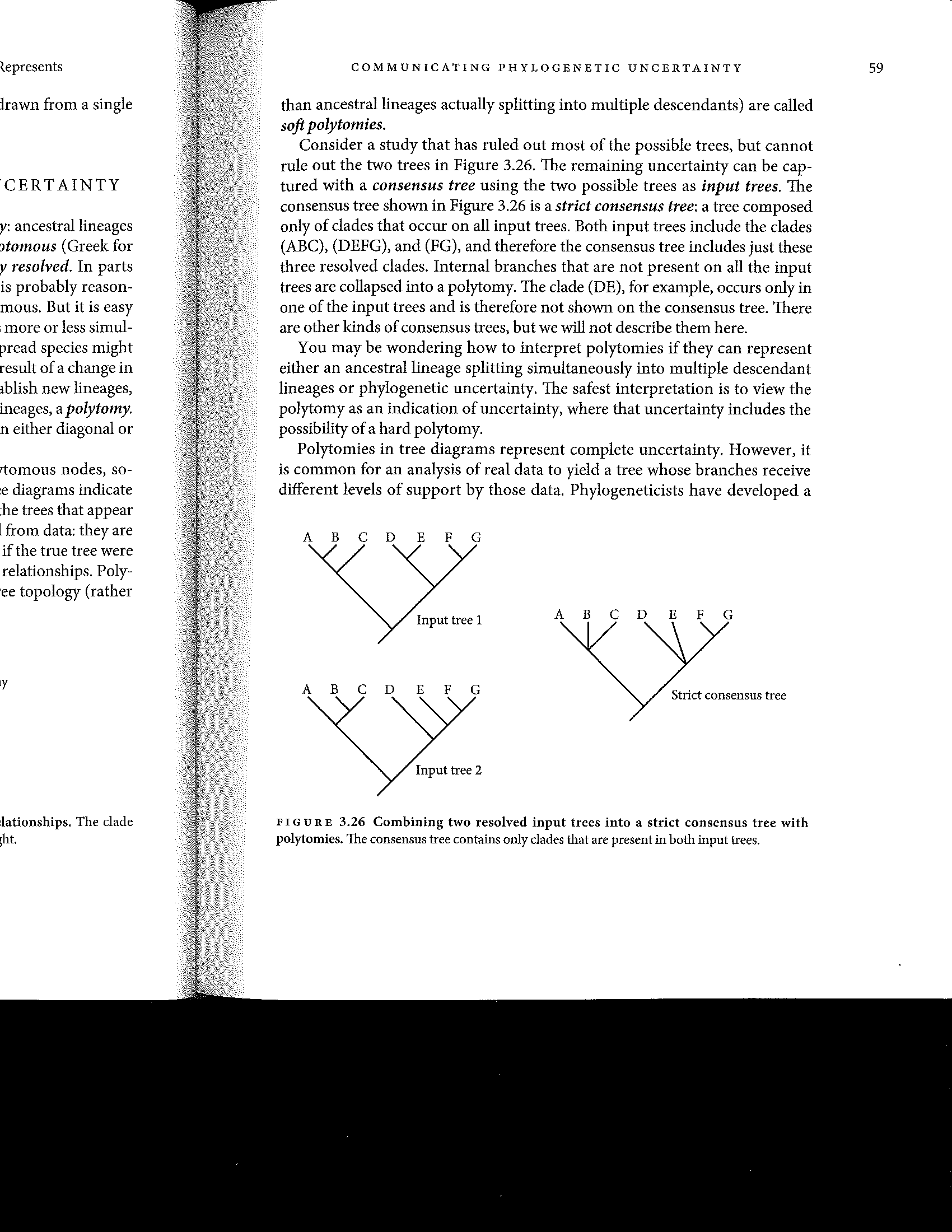
So, what do we do next? What we need is an **optimality criterion.** An optimality criterion is a metric used to decide, given the data, which trees are better and which trees are worse. **Maximum parsimony** is one optimality criterion.

The basic tenet of parsimony is that homoplasy (convergent evolution) happens, but that the optimal tree is the one that minimizes it. This means that the tree topology that requires the fewest character state changes is the most optimal tree.

There are three steps:

1. For a given tree, we consider each character in turn and determine the minimum number of character state changes (steps) that are required to account for the distribution of the character states of the taxa.
2. We sum up the number of changes for all characters. This is the tree length.
3. We repeat this for alternative trees. The tree with the smallest tree length is the most parsimonius tree.

***Proceed to Simple Parsimony Problem.***

Note, two trees may have the same tree length. In that case, they are equally parsimonius. It would then be common to present the **consensus tree** of all most parsimonius trees. In strict consensus tree (see below), all the branches are collapsed in which the two trees disagree. In a **majority rule consensus tree**, you would retain branching events found in at least 50% of the trees and collapse the others. The branches can then be labeled with the % of trees that share that branching event.

The **good thing about parsimony** is that it makes biological sense for many traits. The likelihood of a complex trait arising independently twice is rare, so we minimize this likelihood as a best estimate of what occurred. Of course, if the rate of evolution is high, then traits will be shared due to homoplasy. For example, in a DNA sequence, one taxon may have an A at a particular position that was before a T. Another taxon may have an A there that was once a C. These arose independently. However, we expect that with enough characters, the most parsimonius tree can still accurately reflect the **true tree.**

One bad thing about parsimony is that the higher the rate of evolution, the more likely the shortest tree is not a reflection of the true tree. Another is that parsimony is susceptible to **long branch attraction.** If the true tree has some very long branches because those taxa have evolved at a rapid rate (remember, branch length reflects number of changes from the ancestral node), then the shortest tree tends to be the one that has the long branches near one another. Third, parsimony requires you to select a **weighting scheme,** which means you treat all characters equally, or say that some should influence the tree choice more than others. There is no formal method for identifying the best weighting scheme. Fourth, **parsimony analyses can be really computationally intensive.**

For morphological characters, parsimony methods are still commonly used because other methods are poorly developed. For molecular characters, most reviewers would not allow a paper with a tree based only on parsimony.

**2.3 Before We Continue Further…**

Before we talk about the other methods, we need to reflect on two issues, the necessity to efficiently **search through tree space**, and how one **models molecular evolution.**

***Searching Tree Space***

In the simple parsimony problem that we did above, we only had three ingroup taxa and one outgroup, for a tree of 4 taxa total. There were only 3 possible trees. However, the number of possible trees increases very rapidly.

|  |  |
| --- | --- |
| Number of ingroup taxa | Number of possible trees |
| 2 | 1 |
| 3 | 3 |
| 4 | 15 |
| 5 | 105 |
| 10 | 34,459,425 |
| 50 | ~27.5 x 1074 |

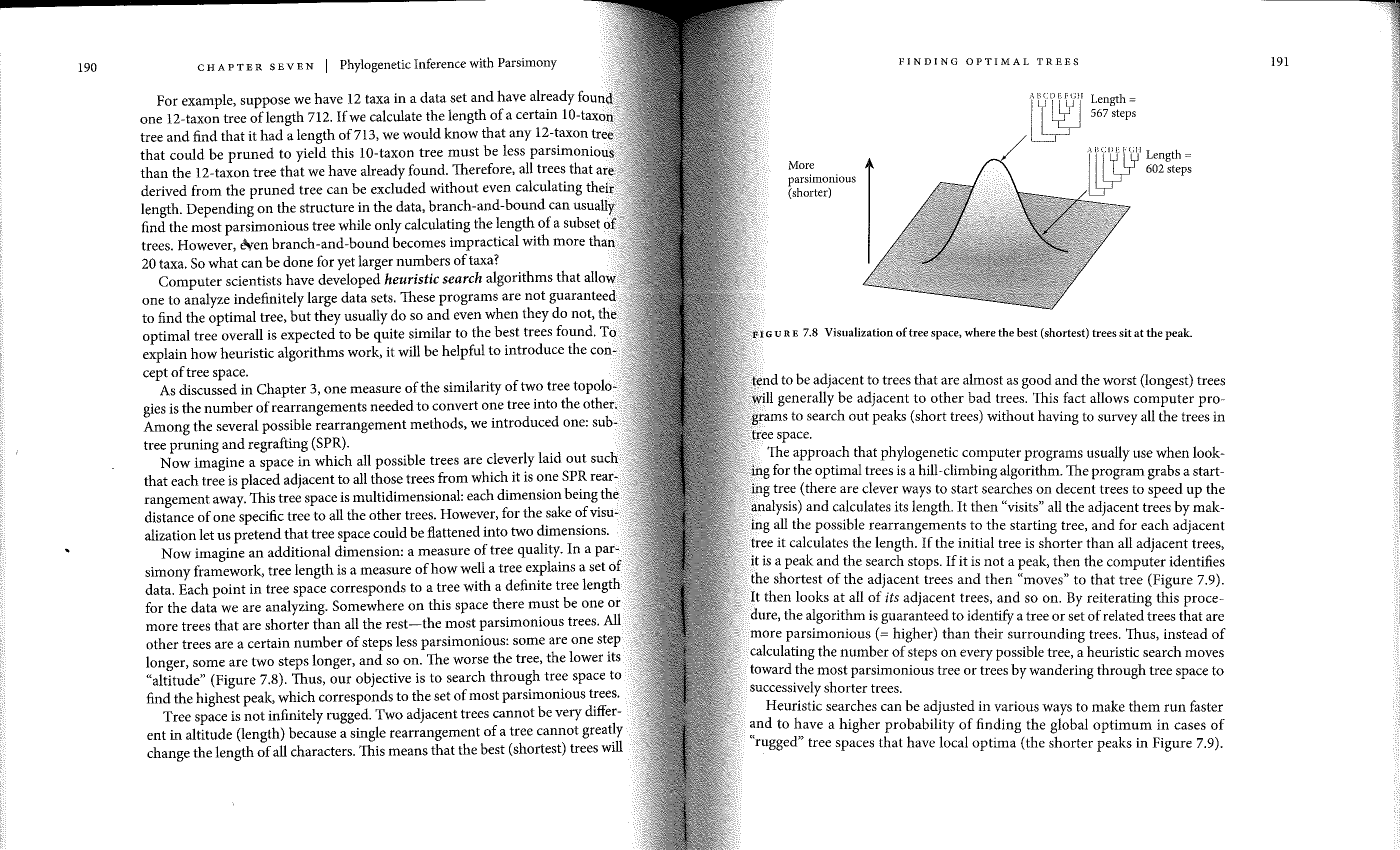
For 52 ingroup taxa, there are more trees than there are estimated numbers of electrons in the entire universe.

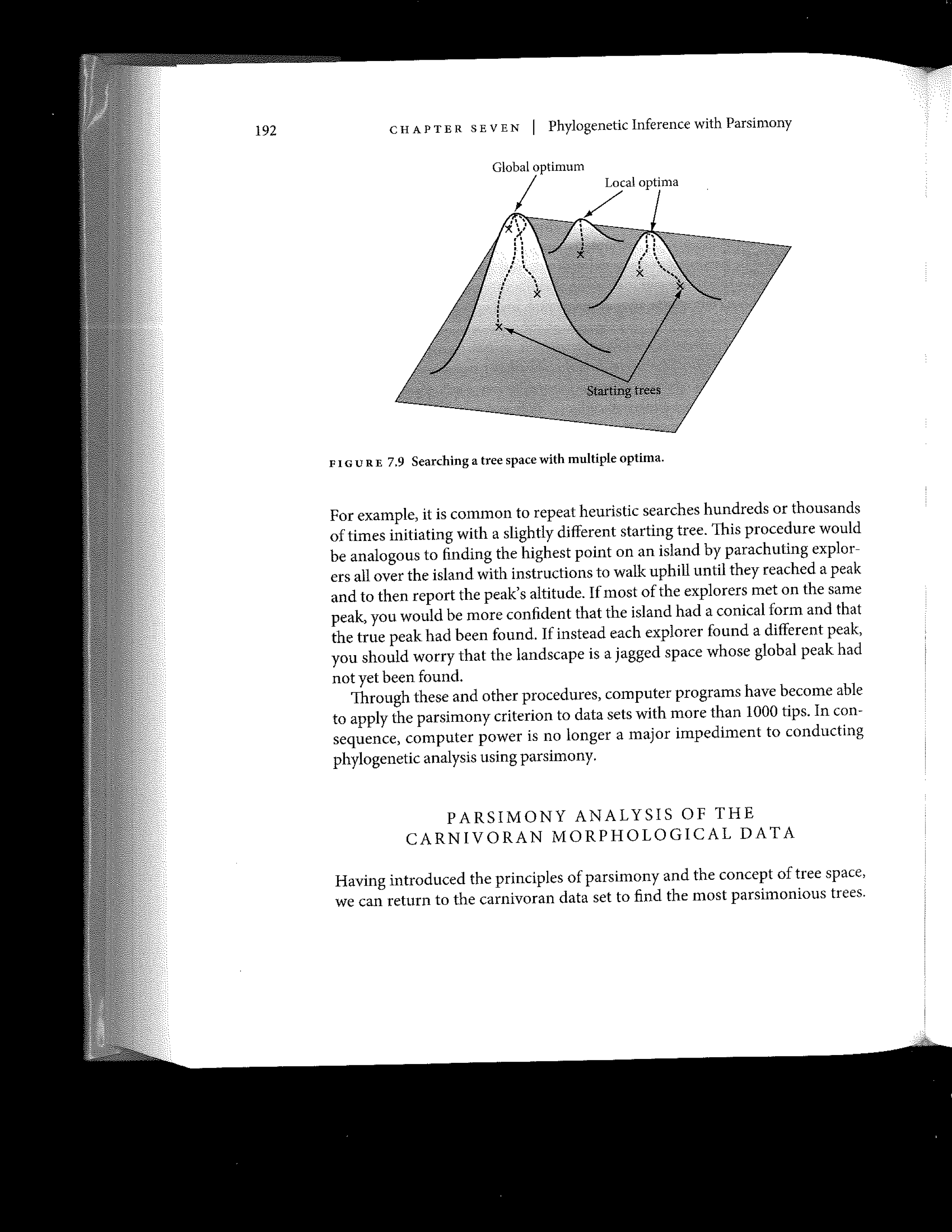
Using a modern computer, it is possible to calculate the length of every tree for up to about 10 taxa. For up to about 20 taxa, it is possible to use a method called **branch and bound.** The idea behind branch and bound is that adding a taxa to a tree can only possibly increase tree length (it can’t get shorter). Thus, if a pruned tree is already longer than the best full tree found so far, you can avoid searching any tree with that pruned set.

But, what if you have more taxa than this? You need a **heuristic search algorithm.**

These methods are not guaranteed to find the optimal tree, but they do a good enough job that the optimal tree is expected to be quite similar to the best tree(s) found.

To think about this, we should think about **tree space.** Across the landscape are placed all trees, with trees that are similar being closer together. So, for example, you swap the relationship of two close branches around and place those trees next to each other in tree space. In this landscape, there are hills and valleys. At the top of the hills are the most optimal trees. For example, the tree of the shortest length is at the top of the highest peak.



****

Heuristic searches start in some place in tree space, where they use a metric to assess that tree (for parsimony, this would be tree length). They then swap branches and assess a tree nearby in tree space. If that tree is better, then you can envision climbing up a hill. You then search trees nearby and see if you get farther up the hill. If you don’t then you continue in a different direction.

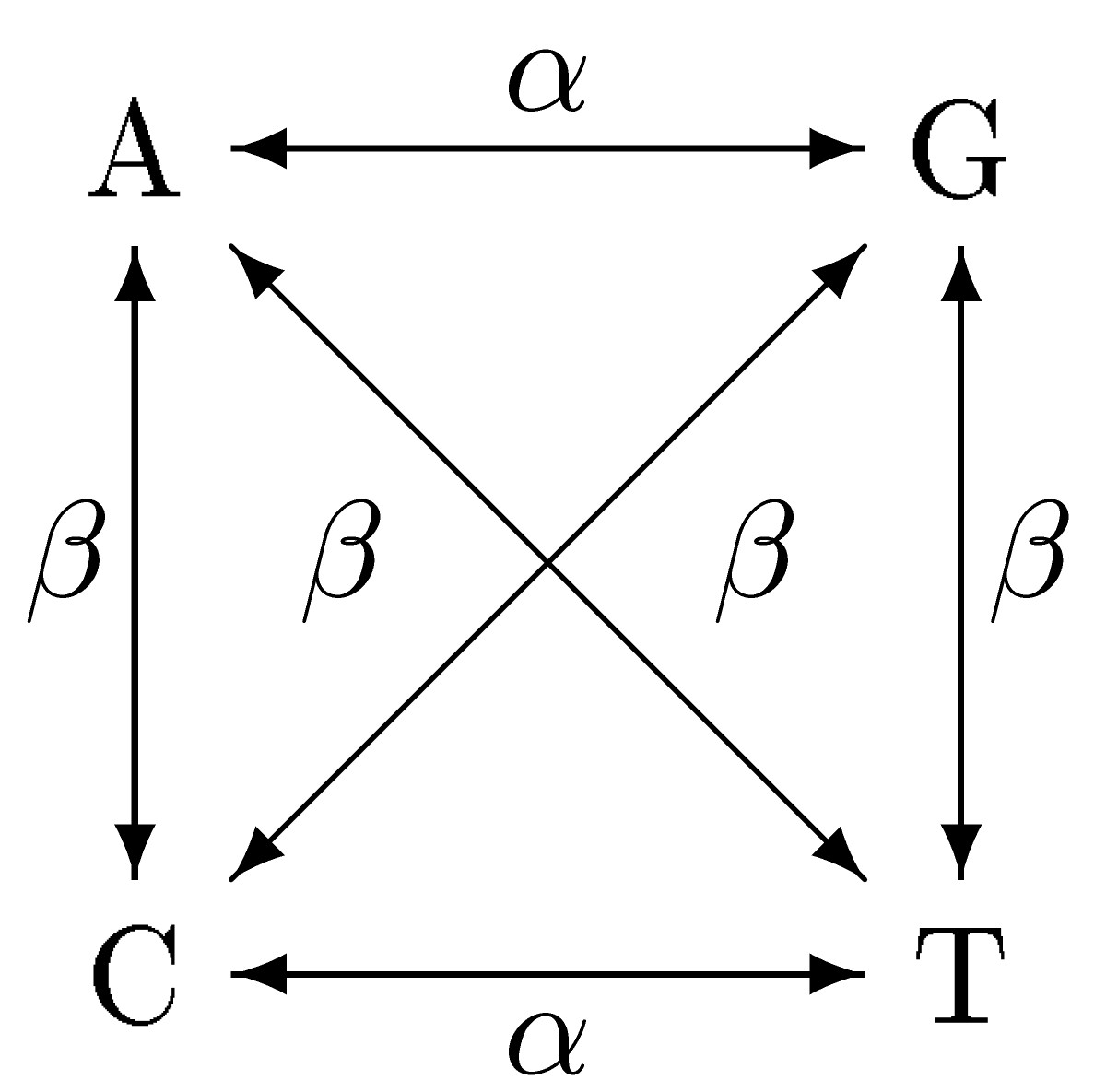
Notice, you could easily climb to the top of a nearby hill, but that may not be the tallest hill in tree space. Thus, it is common to repeat heuristic searches hundreds or thousands of times, initiating with a slightly different starting tree.

***Modeling Molecular Evolution***

As already stated, the number of observed differences is not an accurate reflection of how much evolutionary change has occurred because many changes cannot be observed (i.e., if an A turned to a T and then back to an A). So, we want to determine the proportion of sites that have likely changed given the changes that we can see. We use proportions rather than absolute number of changes, because we want to be able to compare across trees generated using different sequence lengths and sequences of different genes/proteins.

We will focus on modeling molecular evolution of DNA sequences. Similar models have been created for other forms of molecular data.

The **Jukes Cantor model** is the simplest model of DNA evolution. It assumes that the probabilities of any nucleotide changing to any other nucleotide are equal. So, A changes to C with the same probability T changes to A, etc.. We focus on the instantaneous rate of change of nucelotides. In the matrix below, alpha represents the instantaneous rate of change for all transitions and beta is the rate of change for transversions. In the Jukes Cantor model, alpha equals beta:



The **Kimura two parameter model** says that alpha and beta are not equal. This can be important. We see more transitions than transversions in coding data because transversions are more likely to not be silent; they change the protein sequence.

The **HKY Model**

For the HKY Model there is a different rate for transitions and transversions, and we take into account that the base frequencies are not equal. A changing to G is probably really uncommon in a sequence in which Gs are very rare.

The **General Time Reversible (GTR) Model**

For time reversible models, the model does not care which character state was the ancestor and which is derived. The GTR model is the most general time reversible model possible. So, if we look at the below rate matrix table:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | | Substitution Base | | | |
| A | C | G | T |
| Original  Base | A | - | a | b | c |
| C |  | - | d | e |
| G |  |  | - | f |
| T |  |  |  | - |

All six rates are different. For GTR, each rate is actually the product of the substitution rate between bases and the frequency of the base (pie, below) that you would be changing to…



Note that the rows each total 1, because every base has to be something.

These are just a few of the example of models of evolution for DNA sequence data. A quick overview of common models can be found at:

http://www.molecularevolution.org/resources/models/nucleotide

**Invariant Sites**

If you look at some sequences, you will notice sites that don’t vary at all. This may be because the sequences are closely related, or it may be that this position is constrained not to change. In your model, you can include the proportion of invariant sites. In a paper, when someone describes the model that they used, they will put + INV or + I after the model name to show that this was included in the model.

**Rate Variation**

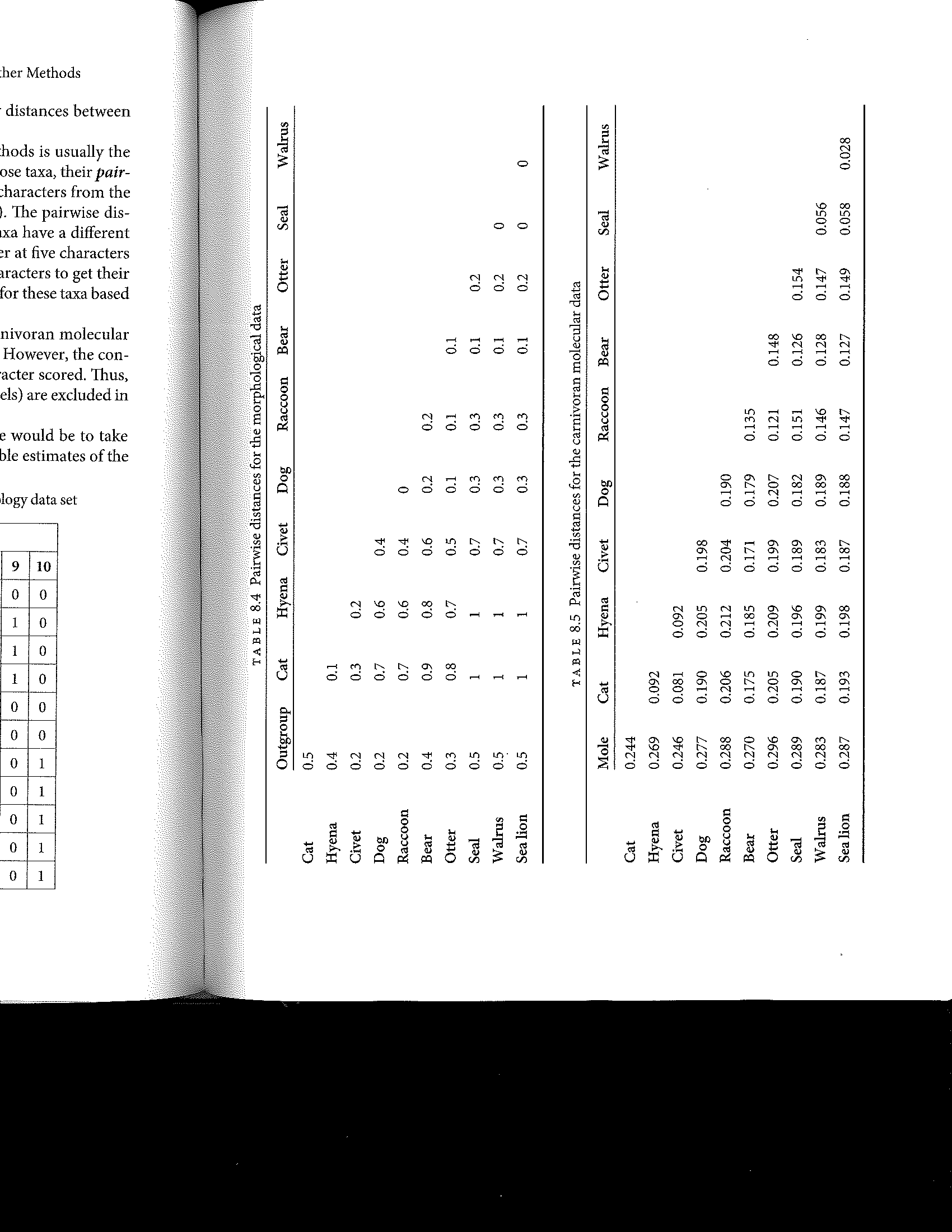
The models above assume that the rate is the same for all positions within the sequence, but some sites might be more constrained than others and show less variation. One could independently estimate the rate for each site, but this is not computationally practical. So, instead, you assume a rate distribution across sites, which is typically modeled with the gamma distribution. The shape parameter of the distribution is referred to as alpha.

GTR +I +G in a paper means that the authors used the general time reversible model, incorporating the proportion of invariant sites and a gamma rate distribution.

**Selection of the appropriate model of evolution** is based on comparing the likelihood (see below) of different models given the data. The simplest model that has no more complex models with significantly higher likelihood is chosen. There are several computer programs (e.g., jModelTest, MrModelTest) where you input a character matrix and it will run through the analyses to help you select the appropriate model. See Appendix A for an output from MrModelTest and the interpretation. See Appendix B for more of how models are statistically compared.

Just as with nucleotide sequences, not all changes occur at the same rate for protein sequences either. For protein sequences, some amino acid substitutions are more likely to occur than others. So, like with nucleotide sequences, for maximum likelihood and Bayesian methods, you will need to provide a model of evolution. Try Prottest to select the model.

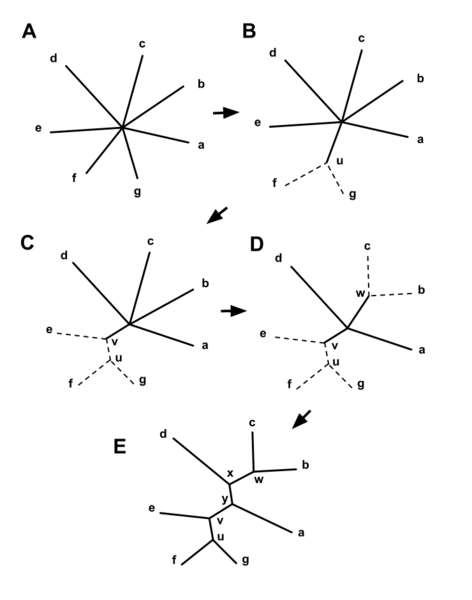
**2.4 Distance Methods**

With consideration of molecular evolution, we can now return to methods of tree construction. The evolutionary distance between two taxa is the sum of all the branch lengths on the path between those taxa. The idea behind distance methods is that if we knew the true evolutionary distances between all pairs of taxa then we would be able to construct the one true tree that corresponded to those distances.

The starting point for distance methods is to compute the pairwise distances for all pairs of taxa, leading to the creation of a distance matrix:

However, these **pairwise distances** are not an accurate reflection of the **evolutionary distances** because they do not account for unobservable changes. For example, if an A changed to a C then to a T, we might see this as only one DNA substitution when two occurred. So, we correct each pairwise distance based on expected changes under a chosen model of evolution (see above) to give the estimated evolutionary distance. These are only an estimate. **Deviations of estimates of molecular evolution from the true distances can have significant impacts on the accuracy of distance based phylogenetic reconstruction**.

**Neighbor-joining** conducts a series of calculations on the distance matrix to get the tree that fits the data. Very roughly, one starts with a completely unresolved tree. The two taxa with the shortest pairwise distance are then joined together in a clade. Distances are then calculated to the node where these are joined. The process is then repeated again and again.



*(Figure from Wikipedia.)*

Neighbor-joining has good and bad points:

* Neighbor-joining is extremely fast.
* Neighbor-joining only gives you one tree without any indication of your confidence for that tree. You can’t determine if one tree is significantly better than another.
* Neighbor-joining is strongly influenced my inaccuracies in the model of evolution chosen.

Use neighbor-joining when you need a quick but not precies estimate of the tree. It’s great for initial looks at the data.

There are other distance-based methods, including **Minimum Evolution** and **UPGMA**.

**2.5 Maximum Likelihood**

In phylogenetics, the goal of Maximum Likelihood (ML) is to search for the tree that has the highest probability of giving rise to the data under the specified model of evolution.

***The Basic Principle***

Let’s say you have a pile of coins in which 50% of the coins are fair (they land 50% of the time heads, 50% tails) and 50% of the coins are biased (they land 75% heads). You grab a coin out of the pile. You have two hypotheses:

1. The coin is fair
2. The coin is biased.

You flip the coin 8 times and get 7 heads and one tail. What is the likelihood of 7 heads and 1 tail under each of your two scenarios?

To get the likelihood you multiply the probability of each toss given the hypothesis.

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| outcome | H | H | H | H | H | H | H | T |  |
| fair | 0.5 | 0.5 | 0.5 | 0.5 | 0.5 | 0.5 | 0.5 | 0.5 | **.004** |
| biased | 0.75 | 0.75 | 0.75 | 0.75 | 0.75 | 0.75 | 0.75 | 0.25 | .**03** |

The likelihood of 7 heads and 1 tail given a biased coin is 0.03, which is 7.5 times more likely than under a scenario of a fair coin. The likelihood ratio, .03/.004, is usually presented as the log likelihood, or ln(0.03/.004), which is 2.01. We consider a log likelihood greater than 2 significant, so the data support the conclusion that the coin is biased.

***For Phylogenetics***

* You can think of each possible tree as a hypothesis.
* The model of evolution is analogous to specifying above that a fair coin lands 50% heads and 50% tails.
* Both topology and branch lengths are assessed in giving a tree a likelihood score.
* The likelihood of a tree is the product of the likelihood of each site’s state given the model of evolution.
* For each tree topology, you find the substitution rate parameter values and branch lengths that optimize the likelihood of getting the observed data (the sequences).
* The best tree topology is the one with the maximum likelihood. The tree with the highest likelihood is the best estimate of the true tree.

The key steps are:

1. Start with a random tree with defined branch lengths and topology.
2. Calculate the likelihood of each site by summing over all possible histories for that site given the model of evolution:



1. Multiply the likelihood of each site for all taxa. (e.g., for all bases in the character matrix)
2. Iteratively change the branch lengths and other parameters on that topology and recalculate the likelihood until the likelihood is maximized.
3. Move in tree space to another tree and repeat.

An important note to prevent confusion, we are searching for the tree with the highest likelihood. We often work with the log likelihood in order to avoid dealing with the computational problems associated with keeping track of very small numbers. So, we end of looking for the tree with the LEAST NEGATIVE LOG LIKELIHOOD.

**2.6 Bayesian Inference**

***The Basic Principle***

Bayesian inference is based on posterior probabilities, which are estimated based on some model after learning something about the data.

So, let us return to our bag of coins. This time, 90% of the coins are fair and 10% are biased (they land heads 75% of the time).

You draw a coin out of the bag. With no other data, your best estimate that the coin is biased is 0.10. However you now flip the coin 10 times. You get:

HHTHHTHHHH

We will call this result X.

You can now incorporate this information into your estimate of the coin being biased. Let’s look at the probability of the result, X, given that the coin is fair versus given that the coin is biased. To do this, we multiply the probabilities of each coin toss under each model (biased, fair).

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | H | H | T | H | H | T | H | H | H | H | P = |
| P[X|Fair] | 0.5 | 0.5 | 0.5 | 0.5 | 0.5 | 0.5 | 0.5 | 0.5 | 0.5 | 0.5 | 0.00098 |
| P[X|Biased] | 0.75 | 0.75 | 0.25 | 0.75 | 0.75 | 0.25 | 0.75 | 0.75 | 0.75 | 0.75 | 0.162 |

The posterior probability that the coin is biased given the data (X) is:

P[Biased|X] =

P[X|Biased] x P[Biased]/P[X] =

P[X|Biased] x P[Biased] / (P[X|Biased] x P[Biased])+ (P[X|Fair] x P[Fair]) =

0.162 x 0.1 / 0.162 x 0.1 + 0.00098 x 0.9 = 0.95

We have greatly increased our estimate of the probability that the coin is biased by incorporating the data on coin tosses.

***For Phylogenetics***

For the coins, our models were a model of what happens when a coin is fair and what happens when a coin is biased. For phylogenetics, our model is a tree topology, branches lengths and parameters (substititions rates, proportion of invariant sites) for the model of evolution.

Our data is our sequence data. So…

**P[Tree|Data] = P[Data|Tree] x P[Tree] / P[Data]**

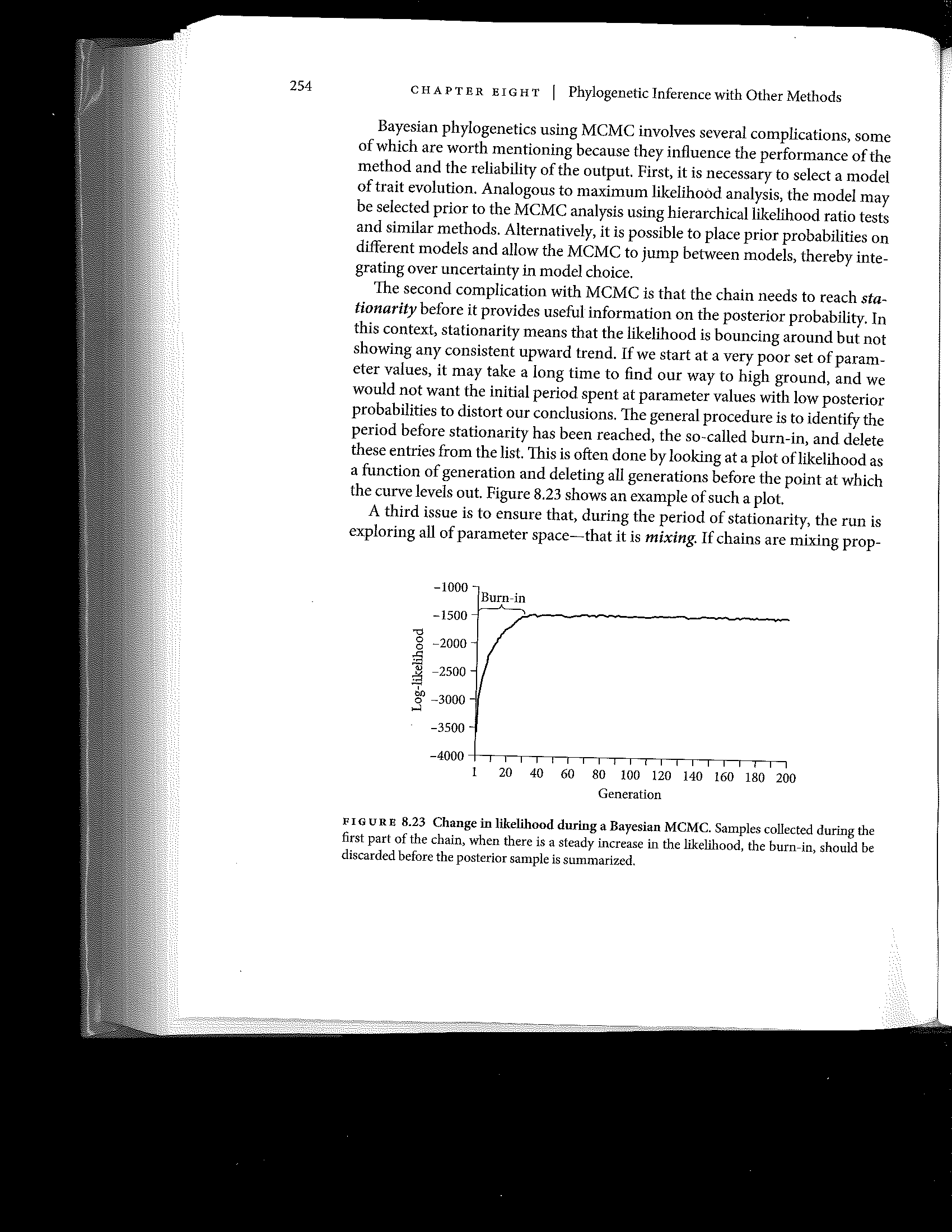
* Calculating the probability of the data given the tree, **P[Data|Tree],** the probability of the data given the tree, entails determing the likelihood of the tree, similarly as is done for maximum likelihood. Rather than determine the values of the parameters (e.g., the rates of substitution), we integrate over the prior probability for all parameters.
* **P[Tree],** the prior probability of a given tree, is often 1/number of possible trees. This is called a flat prior.
* **P[Data],** the posterior probability of the data,is calculated by comparing a new tree and set of parameters to an old tree. The key steps are:
* Start at a randomly selected tree, base substitution parameters that are consistent with your model of evolution, and amount of rate variation across sites.
* Create a new tree that is slightly different (i.e., a tree with one branch length changed or one branch moved). Assess the likelihood of the new tree relative to the previous.
* A random number between 0 and 1 is calculated. If the probability of the new tree given the old tree is greater than the random number, this tree and its associated parameters are kept. If it is not, you ignore this tree and go with the one you had before. If the new tree is much better than the old tree, it is likely to get maintained in this chain of searching. If they are pretty similar, you may save it even if it is a little less likely.
* Each iteration of the above comparison is called a generation. The general trend will be to pick more likely trees each generation. The trees and parameters are stored about every 100 generations into a big file.
* Over time, the likelihoods of the trees each generation will start to stabilize. This can be seen in the output for Mr. Bayes below, showing the likelihoods of sampled trees in the first 1000 generations (top, trees sampled every 100 generations) and in the 9000-10000th generations (bottom).



Each row, above lists the generation number and the likelihood of 8 sampled trees. Mr. Bayes simultaneously does 2 runs of 4 chains each. The two runs are completely independent of each other. Within a run, the 4 chains are swapping information between each other to maximize improvement. This swapping of information is based on the Metropolis-Coupled Monte Carlo Markov Chain (MCMCMC) method.

With Bayesian analyses, you repeat this for millions of generations. Your “best tree” is the one that shows up most often in run.

Originally, the stored trees are terrible, because you started at a random topology that was not informed by the data at all. As you continue, the trees, branch lengths and parameters become more likely (you get a higher likelihood score). Eventually, you hope to stabilize such that each generation the stored topology and parameter values are very similar to the last because you are converging on the most likely topology. The early period before stabilization is called the burn-in.



If we ignore the trees collected during Burn-in, we have a set of trees and parameters with very similar likelihoods. We can then ask what proportion of those trees had a particular clade. Let’s say that of the 7500 trees whose information was stored after burn-in, 90% had taxa A and B form a monophyletic clade. The posterior probability of this clade is 0.9, and is marked on the branch to that clade. This is the support value.

Typically, a tree is drawn that illustrates all the clade with posterior probabilities of great than 0.5. This is called the **Bayesian majority-rule consensus tree** or the Bayesian tree.

***How Maximum Likelihood and Bayesian Analyses Compare***

**While Maximum Likelihood seeks the tree that maximizes the probability of observing the data given the tree and the model of evolution, Bayesian analysis seeks the tree that maximizes the probability of the tree given the data and the model of evolution.**

Bayesian Analysis is searching for a best set of trees. Bayesian analysis will also consider the same tree again if it gets to it. Maximum likelihood is searching for the most likely tree and will only consider each topology once.

Both use likelihood to assess quality.

**3. Assessing Confidence**

One great thing about Bayesian analysis is that you simultaneously search for the best topology and collect information to obtain confidence limits for each clade. In the case of Bayesian inference, these confidence limits are posterior probabilities. This cannot be done during a maximum likelihood or parsimony analysis, where each search leads to a single tree (or maybe a few trees of equal likelihood or that are equally parsimonious). To generate support values for the best tree found under these criteria, you need to be able to compare this tree to other trees.

**3.1 Nonparametric Bootstrap Support**

Non-parametric bootstrapping, often just called bootstrapping, is the most common way to assess confidence for a particular clade in a given tree. Bootstrapping assesses the likelihood of recovering a given clade again if we were able to assess a different set of characters. We simulate other possible character sets, by resampling our data with replacement.



Imagine that the true tree has a clade X. If X is the true relationship between its members, then we expect to see an excess of characters that support its relationship. So, if we create a new matrix of sequences, taking only some of the characters at random, we should still get this clade.

To do this, we create 100s or 1000s of bootstrap data sets, each with the same number of characters as our original data set. We then repeat the original analysis for each dataset. If 750 of 1000 of the analyses have clade X, then our bootstrap score is 75%.

One challenge of bootstrap analyses is that if your original analysis to find the best tree took 10 hours to run, then if you want to do 500 bootstrap replicates, it would take 10 x 500 hours to run all of them under the same conditions. Many likelihood and parsimony analyses can take days even without bootstrapping. There are many ways in which one can relax some of the search criteria to speed up the searches and still get bootstrap values that you are be confident in. Read manuals that are available for packages such as Garli (used for maximum likelihood searches) carefully before you delve into thorough bootstrap analysis.

**Overview of Bootstrapping Steps:**

1. Create set of > 100 new data matrices by randomly sampling characters with replacement. Each data matrix had the same taxa and number of characters but a slightly different set of characters.
2. Analyze each matrix and get a tree
3. Make a majority-rule consensus tree from set of trees. The frequency of a clade amongst the replicates indicates support for the clade



**An Important Final Note**

So, how do you know if you have inferred a correct tree (or clade within a tree)? This is a challenging question, but there are several types of evidence:

1) The BEST evidence would be consilience, meaning that analyses using different types of data (multiple genes, morphological data, etc.), yield similar topologies

2) GOOD evidence comes from strong statistical support (i.e., bootstrap values, posterior probabilities)

3) WEAKER but still informative evidence is that different phylogenetic methods (i.e., parsimony, Bayesian, Maximum Likelihood) yield similar trees.

**Appendix A**

**Log Likelihood Ratio Tests for Model Comparison**

****

**Appendix B**

**An output from MrModelTest.**

Other Programs to test models of evolution include JModelTest and ModelTest. The outputs tend to look similar across these different tools.

---------------------------------------------------------------

\* \*

\* HIERARCHICAL LIKELIHOOD RATIO TESTS (hLRTs) \*

\* \*

---------------------------------------------------------------

Equal base frequencies

Null model = JC -lnL0 = 29114.3945

Alternative model = F81 -lnL1 = 29098.6777

2(lnL1-lnL0) = 31.4336 df = 3

P-value = <0.000001

Ti=Tv

Null model = F81 -lnL0 = 29098.6777

Alternative model = HKY -lnL1 = 28177.1426

2(lnL1-lnL0) = 1843.0703 df = 1

P-value = <0.000001

Unequal Tv and unequal Ti

Null model = HKY -lnL0 = 28177.1426

Alternative model = GTR -lnL1 = 27857.4219

2(lnL1-lnL0) = 639.4414 df = 4

P-value = <0.000001

**Equal rates among sites**

**Null model = GTR -lnL0 = 27857.4219**

**Alternative model = GTR+G -lnL1 = 24688.9902**

**2(lnL1-lnL0) = 6336.8633 df = 1**

**Using mixed chi-square distribution**

**P-value = <0.000001**

No Invariable sites

Null model = GTR+G -lnL0 = 24688.9902

Alternative model = GTR+I+G -lnL1 = 24647.7559

2(lnL1-lnL0) = 82.4688 df = 1

Using mixed chi-square distribution

P-value = <0.000001

\*\* Hierarchical Likelihood Ratio Tests (using hLRT2) \*\*

Equal base frequencies

Null model = SYM+I+G -lnL0 = 24676.8496

Alternative model = GTR+I+G -lnL1 = 24647.7559

2(lnL1-lnL0) = 58.1875 df = 3

P-value = <0.000001

Unequal Tv and unequal Ti

Null model = HKY+I+G -lnL0 = 24674.4551

Alternative model = GTR+I+G -lnL1 = 24647.7559

2(lnL1-lnL0) = 53.3984 df = 4

P-value = <0.000001

Equal rates among sites

Null model = GTR+I -lnL0 = 26228.3164

Alternative model = GTR+I+G -lnL1 = 24647.7559

2(lnL1-lnL0) = 3161.1211 df = 1

Using mixed chi-square distribution

P-value = <0.000001

No Invariable sites

Null model = GTR+G -lnL0 = 24688.9902

Alternative model = GTR+I+G -lnL1 = 24647.7559

2(lnL1-lnL0) = 82.4688 df = 1

Using mixed chi-square distribution

P-value = <0.000001

\*\* Hierarchical Likelihood Ratio Tests (using hLRT3) \*\*

Equal rates among sites

Null model = JC -lnL0 = 29114.3945

Alternative model = JC+G -lnL1 = 25795.5879

2(lnL1-lnL0) = 6637.6133 df = 1

Using mixed chi-square distribution

P-value = <0.000001

No Invariable sites

Null model = JC+G -lnL0 = 25795.5879

Alternative model = JC+I+G -lnL1 = 25760.9375

2(lnL1-lnL0) = 69.3008 df = 1

Using mixed chi-square distribution

P-value = <0.000001

Ti=Tv

Null model = JC+I+G -lnL0 = 25760.9375

Alternative model = K80+I+G -lnL1 = 24747.8809

2(lnL1-lnL0) = 2026.1133 df = 1

P-value = <0.000001

Unequal Tv and unequal Ti

Null model = K80+I+G -lnL0 = 24747.8809

Alternative model = SYM+I+G -lnL1 = 24676.8496

2(lnL1-lnL0) = 142.0625 df = 4

P-value = <0.000001

Equal base frequencies

Null model = SYM+I+G -lnL0 = 24676.8496

Alternative model = GTR+I+G -lnL1 = 24647.7559

2(lnL1-lnL0) = 58.1875 df = 3

P-value = <0.000001

\*\* Hierarchical Likelihood Ratio Tests (using hLRT4) \*\*

Equal rates among sites

Null model = GTR+I -lnL0 = 26228.3164

Alternative model = GTR+I+G -lnL1 = 24647.7559

2(lnL1-lnL0) = 3161.1211 df = 1

Using mixed chi-square distribution

P-value = <0.000001

No Invariable sites

Null model = GTR+G -lnL0 = 24688.9902

Alternative model = GTR+I+G -lnL1 = 24647.7559

2(lnL1-lnL0) = 82.4688 df = 1

Using mixed chi-square distribution

P-value = <0.000001

Unequal Tv and unequal Ti

Null model = HKY+I+G -lnL0 = 24674.4551

Alternative model = GTR+I+G -lnL1 = 24647.7559

2(lnL1-lnL0) = 53.3984 df = 4

P-value = <0.000001

Equal base frequencies

Null model = SYM+I+G -lnL0 = 24676.8496

Alternative model = GTR+I+G -lnL1 = 24647.7559

2(lnL1-lnL0) = 58.1875 df = 3

P-value = <0.000001

--

ATTENTION: The choice based on hLRT can be sensitive for the specific

hierarchy used. If selected models differ, User need to

make the choice!

**Model selected by hLRT (default): GTR+I+G**

Model selected by hLRT2: GTR+I+G

Model selected by hLRT3: GTR+I+G

Model selected by hLRT4: GTR+I+G

--

Model selected: GTR+I+G

-lnL = 24647.7559

K = 10

Base frequencies:

freqA = 0.2177

freqC = 0.2831

freqG = 0.2951

freqT = 0.2042

Substitution model:

Rate matrix

R(a) [A-C] = 0.5280

R(b) [A-G] = 3.6538

R(c) [A-T] = 0.6881

R(d) [C-G] = 0.7091

R(e) [C-T] = 2.6572

R(f) [G-T] = 1.0000

Among-site rate variation

Proportion of invariable sites (I) = 0.2428

Variable sites (G)

Gamma distribution shape parameter = 0.8469

--

PAUP\* Commands Block: If you want to implement the previous estimates as likelihod settings in PAUP\*, attach the next block of commands after the data in your PAUP file:

[!

Likelihood settings from best-fit model (GTR+I+G) selected by hLRT in MrModeltest 2.2

]

BEGIN PAUP;

Lset Base=(0.2177 0.2831 0.2951) Nst=6 Rmat=(0.5280 3.6538 0.6881 0.7091 2.6572) Rates=gamma Shape=0.8469 Pinvar=0.2428;

END;

--

MrBayes Commands Block: If you want to implement a "best" model in MrBayes, attach the next block of commands after the data in your NEXUS file:

(NOTE: In a Bayesian analysis, the Markov chain is integrating over the uncertainty in parameter values. Thus, you usually do NOT want to use the parameter values estimated by the commands in MrModeltest or Modeltest. You rather want to specify the general "form" of the model (such as nst=1 etc.)

[!

MrBayes settings for the best-fit model (GTR+I+G) selected by hLRT in MrModeltest 2.2

]

BEGIN MRBAYES;

Prset statefreqpr=dirichlet(1,1,1,1);

Lset nst=6 rates=invgamma;

END;

-

---------------------------------------------------------------

\* \*

\* AKAIKE INFORMATION CRITERION (AIC) \*

\* \*

---------------------------------------------------------------

Model selected: GTR+I+G

-lnL = 24647.7559

K = 10

AIC = 49315.5117

Base frequencies:

freqA = 0.2177

freqC = 0.2831

freqG = 0.2951

freqT = 0.2042

Substitution model:

Rate matrix

R(a) [A-C] = 0.5280

R(b) [A-G] = 3.6538

R(c) [A-T] = 0.6881

R(d) [C-G] = 0.7091

R(e) [C-T] = 2.6572

R(f) [G-T] = 1.0000

Among-site rate variation

Proportion of invariable sites (I) = 0.2428

Variable sites (G)

Gamma distribution shape parameter = 0.8469

--

PAUP\* Commands Block: If you want to implement the previous estimates as likelihod settings in PAUP\*, attach the next block of commands after the data in your PAUP file:

[!

Likelihood settings from best-fit model (GTR+I+G) selected by AIC in MrModeltest 2.2

]

BEGIN PAUP;

Lset Base=(0.2177 0.2831 0.2951) Nst=6 Rmat=(0.5280 3.6538 0.6881 0.7091 2.6572) Rates=gamma Shape=0.8469 Pinvar=0.2428;

END;

--

MrBayes Commands Block: If you want to implement a "best" model in MrBayes, attach the next block of commands after the data in your NEXUS file:

(NOTE: In a Bayesian analysis, the Markov chain is integrating over the uncertainty in parameter values. Thus, you usually do NOT want to use the parameter values estimated by the commands in MrModeltest or Modeltest. You rather want to specify the general "form" of the model (such as nst=1 etc.)

[!

MrBayes settings for the best-fit model (GTR+I+G) selected by AIC in MrModeltest 2.2

]

BEGIN MRBAYES;

Prset statefreqpr=dirichlet(1,1,1,1);

Lset nst=6 rates=invgamma;

END;

--

\*\* MODEL SELECTION UNCERTAINTY : Akaike Weights \*\*

Model -lnL K AIC delta Weight CumWeight

----------------------------------------------------------------------------------------------------------------------------------------

**GTR+I+G 24647.7559** 10 49315.5117 0.0000 1.0000 1.0000

HKY+I+G 24674.4551 6 49360.9102 45.3984 1.39e-10 1.0000

SYM+I+G 24676.8496 7 49367.6992 52.1875 4.65e-12 1.0000

GTR+G 24688.9902 9 49395.9805 80.4688 3.36e-18 1.0000

SYM+G 24702.9102 6 49417.8203 102.3086 6.08e-23 1.0000

HKY+G 24733.9199 5 49477.8398 162.3281 5.64e-36 1.0000

K80+I+G 24747.8809 3 49501.7617 186.2500 3.60e-41 1.0000

K80+G 24790.9062 2 49585.8125 270.3008 0.00e+00 1.0000

F81+I+G 25674.9766 5 51359.9531 2044.4414 0.00e+00 1.0000

F81+G 25721.4043 4 51450.8086 2135.2969 0.00e+00 1.0000

JC+I+G 25760.9375 2 51525.8750 2210.3633 0.00e+00 1.0000

JC+G 25795.5879 1 51593.1758 2277.6641 0.00e+00 1.0000

GTR+I 26228.3164 9 52474.6328 3159.1211 0.00e+00 1.0000

SYM+I 26298.4258 6 52608.8516 3293.3398 0.00e+00 1.0000

HKY+I 26384.8066 5 52779.6133 3464.1016 0.00e+00 1.0000

K80+I 26460.1289 2 52924.2578 3608.7461 0.00e+00 1.0000

F81+I 27368.7422 4 54745.4844 5429.9727 0.00e+00 1.0000

JC+I 27392.6816 1 54787.3633 5471.8516 0.00e+00 1.0000

GTR 27857.4219 8 55730.8438 6415.3320 0.00e+00 1.0000

SYM 27938.2656 5 55886.5312 6571.0195 0.00e+00 1.0000

HKY 28177.1426 4 56362.2852 7046.7734 0.00e+00 1.0000

K80 28211.5098 1 56425.0195 7109.5078 0.00e+00 1.0000

F81 29098.6777 3 58203.3555 8887.8438 0.00e+00 1.0000

JC 29114.3945 0 58228.7891 8913.2773 0.00e+00 1.0000

-------------------------------------------------------------------------------------------------

-lnL: negative log likelihood

K: number of estimated (free) parameters

AIC: Akaike Information Criterion

delta: Akaike difference

weight: Akaike weight

cumWeight: cumulative Akaike weight