STAT 771: Project

Ralph Møller Trane

2018-11-02

Introduction

Set-up

Given a number of different species, is it possible to say anything about how these are related? This is the question phylogenetic trees aim at answering. In general, aligned DNA sequences from a number of taxa (species) are obtained. Based on these, a model is specified and fitted to the data. This results in a way to calculate "the most likely tree, given the data".

For this project, I will consider rooted trees. There is exactly one *root* in the tree – namely the one node with no parent node. A node is said to be a *child* of a different node (its *parent* node) if it is further 'down' the tree. Each edge will have a *branch length.* This represents the time it took for the species in the child node to evolve from the species in the parent node. A node in the tree is called a *leaf* if it does not have any children. A *clade* is a node and all its descendants. Note that this is related to the taxa, not the structure. This will also be refered to as a *subclade* of the parent node. Here, all internal nodes (i.e. nodes that are not leaves) will have exactly two subclades. I.e. each tree will have exactly N leaves, one for each taxon.

Background

The posterior probability distribution on a set of phylogenetic trees is a well-defined mathematical object, if we are given a likelihood model, prior distribution, and data. However, it is difficult to use in practice.

Using MCMC methods, we can create samples of trees, and base inference on these posterior samples by estimating the posterior distribution of phylogenetic trees. This is typically done with simple sample relative frequencies (SRF) – i.e. the probability of a tree is the relative frequency said tree shows up in our MCMC sample.

This approach has two main disadvantages, both of which arise due to the very large sample space. First of all, since we use MCMC to sample trees from the estimated posterior distribution, and since the sample space is very large, we need to allow the MCMC to run for a very large number of generations to get a good, representative sample. Second, there'll be many trees that simply are not sampled, even though many of these will have non-zero probabilities. Some of these might even be as likely (or more) as many trees observed in the sample. However, the estimated probability of these trees would still be 0. At first, this might seem like a minor inconvenience – after all, the error of the estimated probabilities is rather small – but some methods are sensitive to this. For example, in Bayesian concordance analysis (BCA), the first step is to calculate separately the posterior probabilities of trees for each of many genes (Ane et al. 2007). If a tree is very probable in several distributions based on single genes, but is unsampled in the distribution of another given gene, the simple estimate of zero for the posterior probability of the tree for the given gene can bias the results in the second stage of the BCA analysis.

One way to correct the second problem mentioned would be to come up with a different way to estimate the posterior probabilities of trees that do can be applied to trees that do not appear in samples, but might be relatively probable.

Conditional Clade Distributions

To do so, Larget (2013) proposes to use Conditional Clade Distributions (CCD). The probability that a tree T is the correct tree is the probability that the true tree contains all the clades $C_1, \dots, C_n$ that make up the tree T. So, in other words, $P(T) = P(C_1 \cap \dots \cap C_n)$. If the clades were all independent, this would simplify to the product of the clade probabilities $P(C_i)$. This is clearly not the case, but it can often be assumed that subclades are conditionally independent given the parent node. Under this assumption, using rules of conditional probability and conditional independence, one can fairly easily show that

$$P(T) = \prod_{C \in \{\text{all clades of T}\}, |C|>1} P(L(C,T) \cap R(C,T)|C).$$

So whenever subclades are (at least approximately) independent given the parent clade, full tree probabilities can be calculated using the CCD. This means we can now estimate tree probabilities of trees not sampled!

Bootstrap vs. MCMC

This solution to the second mentioned problem provides an alternative way of actually estimating tree probabilities all together. The next question then is if this can be utilized to estimate the tree probabilities without relying on MCMC samples, but some faster method.

Much research has been done in how to use bootstrapping methods to gain insights into phylogenetic trees. In general, using bootstrap samples to estimate tree probabilities does not work super well. One reason for this is that resampling among observed aligned DNA sequences is not approximately sampling from the sampling distribution (Holmes, 2003). However, now that the target is not the posterior distribution of trees, but rather clades, maybe bootstrap samples could provide enough information to yield good results. This could potentially speed up the estimation of the posterior tree distribution by a significant amount. This is of great interest these days, as researchers are looking to include more and more taxa in their analyses.

Our goal

The goal for this project is to look into when it is appropriate to estimate tree probabilities using CCDs obtained from a bootstrap approach. To do so, data from different trees will be simulated, and using two different approaches, CCDs will be obtained.

The first approach is using the `MrBayes` software (https://github.com/NBISweden/MrBayes). This generates MCMC samples of trees from the posterior tree distribution. Based on the sampled trees, the CCDs will be estimated using the `ccdprobs` software (https://github.com/larget/ccdprobs).

The second approach will use the `Bistro` software (https://github.com/larget/ccdprobs/tree/master/Bistro) to generate bootstrap samples. Again, CCDs will be estimated using the `ccdprobs` software.

Simulating data

Before moving on to how data from a tree is simulated, some thought had to be put into how to specify a tree in Julia. Typically, a tree is stored as a long text string. An example of such is specified below.

```
"(1:0.0556631,2:0.0693414,((3:0.0739067,((4:0.039114,5:0.0372726):0.0130065,6:0.057055):0.00748735)
:0.0154953,(((((7:0.00397016,8:0.000611082):0.0220259,9:0.0203416):0.0245329,10:0.0530554):0.0427686,
11:0.0850486):0.00315267,12:0.0918821):0.0877109):0.0095466)"
```

Such a string specifies a complete tree with taxa and branch lengths. `(` specifies a new subclade, `,` separates children of said subclade, the numbers not preceded by colon specifies the species in the given leaf, and numbers preceded by a colon specifies the length of the edge from the parent node to the given node. `)` specifies the end of a subclade.

To be able to work with this more interactively in Julia, I use the following four mutable structures:

- `tree`: contains two objects
  - `nodes`: an `OrderedDict` specifying the nodes of the tree
  - `edges`: an array specifying edges by parent and child node that each edge connects, and the branch length
- `root`: a special node with no parent. Contains three objects
  - `children`: the nodes that are the children of this node
  - `values`: the species in this clade (for the root, it will always be all taxa)
  - `aligned_DNA`: the string of aligned DNA in this node.
- `node`: contains four objects
  - `parent`: specifies the parent node
  - `children`, `values`, `aligned_DNA`: all as for `root`

- `leaf`: as `node`, but without the `children` object.

After creating a `tree` object from a string as the one above, I can fairly easily simulate data at the leaves. This is done in the following way:

- define the DNA sequence at the root, and a transition matrix Q
    - when working with DNA sequences, $Q \in \mathbb{R}^{4\times 4}$ with q_{ij} being the transition probability of an c_i changing to c_j for $(c_1, c_2, c_3, c_4) = (A, C, G, T)$.
    - the probability of observing a change over an edge of length t is $P(t) = e^{Qt}$.
- loop over all nodes, and for each create a new sequence based on the DNA sequence in the parent node, and the length of the edge connecting the node with the parent node.

At each step, the `aligned_DNA` slot of the `node` object is filled. (For full implementation: https://github.com/rmtrane/ccdprobs_771/tree/master/scripts)

To-do list

- create function to write `tree` structures to string format, so they can be used with `Bistro` and `MrBayes`
- think about what transition matrix/tree topology to use for simulations
- simulate and analyze data
    - resulting in two sets of CCDs
- think about what to do with results of simulated data
    - compare tree probabilities
    - compare CCDs
    - which features of trees could be compared between runs?