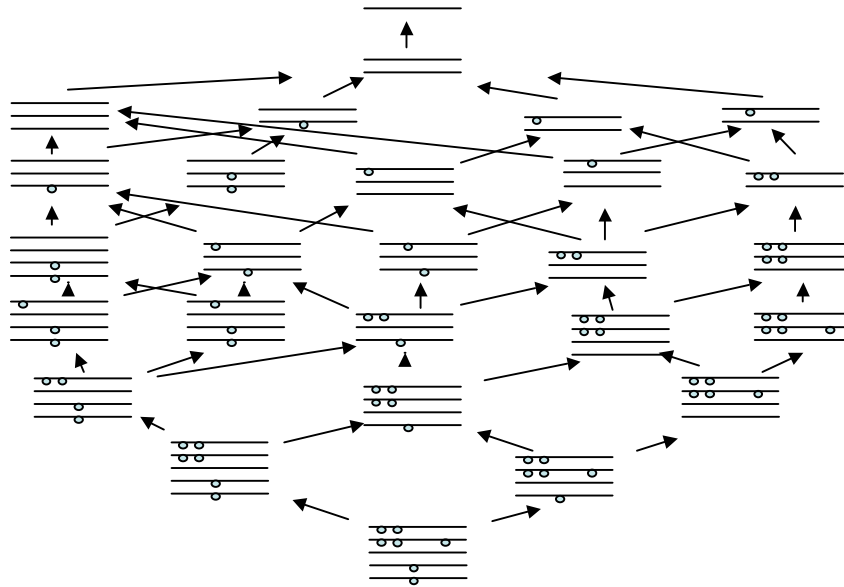


“Corner Cutting” approaches to the Ethier-Griffiths-Tavare Recursions

Almost infinite site models, Ancestor inference and basic accelerations

1.3.08

Background and Motivation. Calculating the likelihood of a set of sequences sampled from a population is important in order to estimate selection, recombination, mutation and population structure. In a simplified model of sequences called the infinite sites model (Kimura and Ohta, 1971) calculating this likelihood is possible in a basic population model thanks to a series of recursions published by Ethier, Griffiths and Tavare in the period 1987-95. In this project proposal we propose to explore a simple idea of how to extend and accelerate the fundamental recursions. In a recent paper, Lyngsø, Song and Hein (2008) accelerated likelihood calculations by a very large factor (10^6 - 10^9) by only considering ancestral states reachable in less than k recombinations. Unfortunately, an even larger factor is needed to make the algorithm practical. However, very similar ideas can be applied to the model without recombination and a DNA model (not infinite sites) of sequences. The EGT recursions have not been explored from a computational view point and could have much larger practical potential than the original papers indicate, where only toy examples are explored. If these algorithms could be applied to say 40 sequences with 50 mutations, they would be excellent analysis tools.



The configuration in the bottom of the figure has 5 sequences and there are 4 columns where there are mutations present (little balls). We assume we know the ancestral state in each position, so there is not doubt that the single ancestral sequence at the top, must be “ball free”. The last event in the history leading to the configuration at the bottom could either be a duplication creating the two identical sequences at the bottom or a mutation creating a single ball on the second sequence. All configurations that can be visited on a path from the bottom configuration to the single sequence together with which configuration could lead to which configuration. This defines a directed acyclic graph (DAG).

The probability of the data (bottom configuration above) can be calculated by the following recursion:

$$P_n(\mathbf{n}) = \frac{n-1}{n-1+\theta} \sum_{n_j > 1} \frac{n_j-1}{n-1} P_{n-1}(\mathbf{n} - \mathbf{e}_j) + \frac{\theta}{n-1+\theta} \sum_{\text{singletons}} \frac{n_i+1}{n} P_n(\mathbf{n} - \mathbf{e}_j + \mathbf{e}_i)$$

with the initial condition $P_1(1) = 1$. Θ is here the scale mutation rate $4\mu N$, μ is mutation rate per gene per generation and N is (effective) population size and the relationship of the sequence are described by the standard coalescent (Hein, Schierup and Wiuf, 2005). n is the number of alleles in the sample, \mathbf{n} is the sample, n_j is number of type j . \mathbf{e}_j is the vector with all θ except at the place indicating type j . $P_n(\mathbf{n})$ is thus decomposed into what which configurations could have created \mathbf{n} by a duplication (coalescent event forward in time) or a mutation event. A similar recursion (not on a DAG anymore) can be formulated for a proper DNA model of a sequence, but this will in its naïve version lead to sums ($i=1, \dots, n$) of 4^{ik} configurations, where k is the length of the sequence.

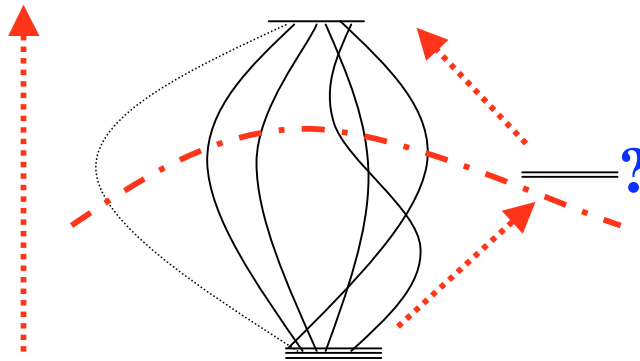
The project. In this project we propose to apply corner cutting techniques to approximate the EGT recursions for a proper DNA model. The main observation is that only a very small number of the possible configurations will contribute significantly to the likelihood of the data for mutation rates of reasonable size. The configurations necessary to consider would be the ones defined by the infinite site model (or DNA model with at most 1 event in each position) plus a few extra mutations in total. Three issues that could be addressed by this project are

1. Acceleration of the basic EGT algorithm by discarding states that can only be reached by unlikely paths. Find most and least likely paths can easily be done by dynamical programming, but cannot be done in a greedy way before filling out the complete matrix – taking the most/least likely step at each level might not define the most/least likely path.

2. Relaxation of the assumption that no more than 1 mutation can happen in the same position. This is possible as DNA sequences indeed are finite. I.e. we propose a method that would allow a few. Therefore the name: "almost infinite". In contrast to the Lyngsø, Song and Hein (2008) paper the number of recombinations is now zero and cannot be used as criteria for discarding states.

3. In the original Ethier and Griffiths (1987) paper the ancestral states were known, but this was relaxed in the Griffiths and Tavaré (1995) paper where all possible ancestors were considered. Often the sequence from an outgroup can be found. Such a sequence will not provide the ancestral states perfectly, but could provide a strong prior around a single ancestral sequence, where only ancestors with 1-2 mutations relative to the outgroup would have significant probability. The graph could be given one sink (root) by having a last step with special evolution leading from possible ancestors to the out group with a special Θ_o (o for outgroup).

It is possible to add configurations to a growing graph and calculate the probability of the data if they have arisen by histories represented by paths within that graph. Nodes can be added singly or in groups dependent on computational expediency. A potential new node must be added after an evaluation of how much it is likely to add to the probability of the data. This involves how many paths are leading to and from it and what is their average probability.



In a dynamic expansion of the configuration space approach it must be evaluated how much a configuration in question (blue question mark) will contribute to the probability of the data. This will be the probability of the paths leading to the data multiplied by the probability carried by paths leading to the ancestor configurations.

The proposed algorithm can best be described as a dynamic expansion of the configuration space based on current information. It does not provide a guarantee at present to calculate the quantity in question, but it is most likely an extremely good approximation. Possibly combinatorial arguments could provide proper bounds. The "in"-components are easy to evaluate as they are always part of the growing graph, but the "out"-components must be evaluated by guesses or lower/upper bounds. The efficiency of the proposed approach hinges on how uneven the contribution is to the total probability of the data. The more uneven, the more efficient the algorithm will be. This is unknown at present, but the combined effect of the combinatorial factor and the path probabilities justifies the expectation of an efficient algorithm. To exemplify, states reached by choosing only mutations (or only coalescences) will have a low path probability, but also a small combinatorial factor as there few events on the paths that could be permuted.

Cycles are possible in the resulting graph. For instance $A \rightarrow C \rightarrow A$. They will add to the complexity of the programming that has to be done, as nodes in a cycle will have to be evaluated simultaneously and a small linear equation system will have to be solved. Without cycles each node can be evaluated if the precursor nodes have been evaluated. However, a strong case can be made for ignoring cycles. If we want explore histories with 5 (or 100) extra mutations relative to the infinite site model, then cycles will only occur if more than two are at the same position in a time period when no other events changes the configuration.

Plan

1. Read the basic literature necessary to understand the fundamental algorithm.
2. Implement the basic fundamental algorithm as described above.
3. Implement upward-downward version of the algorithm and analyze which configurations contribute how much to the total probability. Normally we only do an upward pass, but in this case a downward pass would also be needed.
4. Implement an ancestral state-doubling algorithm, where a new set of ancestral states is incorporated into the graph after an evaluation of their likely contribution.
5. Tabulate how large data set the resulting algorithm can analyze as function of Θ , sample size and number of segregating sites.
6. Investigate the number of multiple events at the same position as function of key parameters. How much does the size of ancestral states grow?

Comments i. This project needs good programming skills, understanding of probability and algorithms. Knowledge of population genetics would be motivating, but is not necessary.

References

- Ethier and Griffiths (1987) "The infinitely many sites model as a measure valued diffusion" *Ann. Prob.* 15.2:515-545.
 Griffiths, R.C. (1987). "Counting genealogical trees" *J. Math. Biol.* 25, 422-432.
 Griffiths, R.C. (1989). "Genealogical-tree probabilities in the infinitely-many-sites model". *J. Math. Biol.* 27, 667-680.
 Griffiths, R.C. and Tavaré, S. (1995). "Unrooted genealogical tree probabilities in the infinitely-many-sites model". *Mathematical Biosciences* 127, 77-98.
 Griffiths, R.C. (2001). "Ancestral inference from gene trees" In: Donnelly, P. and Foley, R. (Eds.), *Genes, Fossils, and Behaviour: an Integrated Approach to Human Evolution*, IOS Press, Amsterdam, pp.137-172.
 Hein, Schierup and Wiuf (2005) "Gene Genealogies, Variation and Evolution" OUP chapter 2.
 Kimura and Ohta (1971) "Theoretical Aspects of Population Genetics" Princeton
 Lyngsø, Song and Hein (2008) "Accurate calculations of likelihoods in the coalescent with recombination using parsimony" *Recomb 2008* Singapore