Ma et al, from Gene Trees to Species Trees

AJ Milne

November 22, 2001

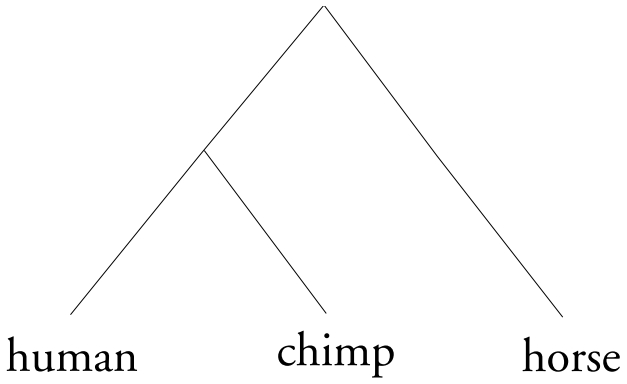

Figure 1: A species tree

0.1 introduction

The paper is Ma et al, From Gene Trees to Species Trees, SIAM Journal on Computing, Vol 30, No, pp. 729-252. [3].

The problem, briefly stated, is the reconciliation of gene trees and species trees.

I'll be concentrating on the NP-hard proofs in the paper, as there is a great deal of other material. Another aspect of this paper readers might find interesting is the development and testing of a heuristic method for finding reconciled species trees, which Ma et al claim has outperformed the current method in Page's GeneTree.

0.2 Problem explanation – descendance of genes and of species

A species tree is a representation of similarity of a number of species. See Figure 1.

A gene tree is a similar construction, and represents the relative similarity within a group of homologolous genes. An example appears in Figure 2.

Gene trees are derived as follows: given a set of sequences like this, it can be determined which is more similar to which others, though not trivially, relatively simply, with several well-known algorithms. These algorithms essentially just try to find what's the minimum number of changes you have to make to get from one gene to another. The algorithms are looking for

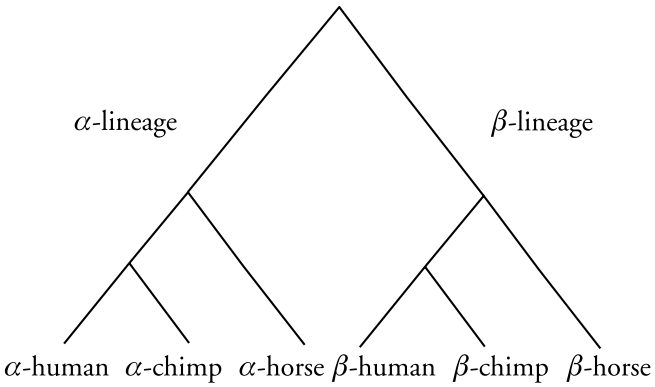

Figure 2: A gene tree

the tree that represents a branching tree of genes representing these shortest paths, so we call the resulting tree the most parsimonious one. So these are parsimonious gene tree deriving algorithms.

Typically, when genes are sequenced, and the phylogenetic relationship between the host species is known, a pattern like that in the given hemoglobin gene tree appears, in which the phylogeny of the species involved is traced by the similarity of the genes (with, in this example, a duplicate set adding only slightly to the complexity).

Why this is isn't difficult to grasp. Genes within two relatively similar species are more similar to one another than they are to a third gene of the same family within a more distantly related species most probably because those species descended from one another in the order their similarities suggest. The two more similar species are more similar precisely because they've had a more recent common ancestor, and thus less time and fewer opportunities for their homologolous seqences to diverge from one another, so there are fewer divergences.

However, this is not always the case. Not uncommon sequences of gene duplication and loss events can transpire to confuse this picture. A schematic example is shown in figure 3.

In this example, the actual phylogenetic relationship between a horse, a chimp, and a human is confused, due to the effective 'reemergence' at the species level of two gene duplication events – the splitting of the β line from α, and the subsequent splitting of γ from β. The resulting sequence data would suggest that chimps and horses shared a more recent common ancestor than do humans and either of these groups.

Were this the only data we had contributing to our knowledge of the

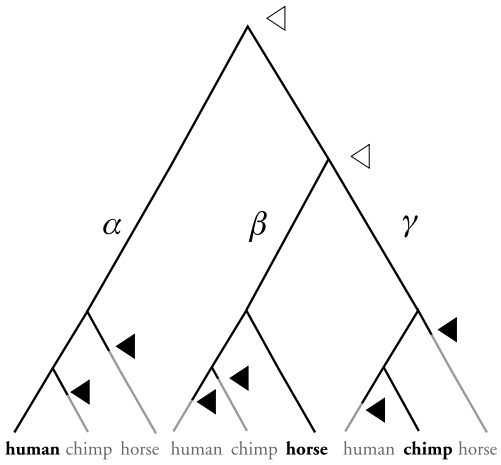

Figure 3: A gene tree obscured by duplication and gene loss events

phylogeny of these three species, we would wind up with a rather confusing picture.

However, species contain many genes, and many detectable homologolous sequences within. The result is, typically, when attempting to deduce species phylogeny from sequences, researchers have several different data sources to consider.

So this is the gene to species tree problem, in informal terms: given n gene trees G_i, find a species tree S which minimalizes the number of gene duplication and loss events (the overall mutation cost) we must introduce to reconcile the trees. It is generally assumed this tree will probably most closely represent the phylogenies of the species involved.

Ma et al state this in more formal terms in their paper, and define a number of metrics for assessing the cost of duplication and loss events associated with reconciling gene and species trees. The duplication cost will be discussed in more detail in the discussion of the proof.

0.3 Solutions to now

There have been several papers on the problem in the past five years. Page and Charleston, from the University of Glasgow, published a work in 1997 that serves as an excellent introduction to the problem and to one family of heuristic solutions. [4].

Page also wrote GeneTree, a freeware package that's something of a

standard in the field now, which used the approaches this paper describes.

The framing of the problem in the 1997 paper, and in GeneTree is that you can attach a cost to each event you interpolate into a gene tree to reconcile it with a common species tree, and the trick is to find a solution that minimizes total cost, summed over all the genes you're using.

The methods described in the 1997 paper worked with a toolbox of heuristic searches, that used a kind of hill-climbing strategy, starting from various trees and attempting to wiggle them in various ways to climb adjacent hills – or to minimize the gene tree interpolation cost. It was also understood that, if the 'fitness landscape' for solutions (where peaks of fitness represent local minima in terms of interpolation costs) were generally rugged, such strategies would get stuck on local peaks, when higher but remote global peaks, separated from the starting point by a valley, would be missed.

The 1997 paper critiques a 1996 study by Guigo et al as evidence for this possibility – the 1996 study, the Page paper claims, got stuck on some local maxima, and they were later able to find better ones.

As of 1997, various approaches working from different starting points were used. Two basic methods for wiggling the trees were used – NNI (nearest neighbour interchange), and cut and paste, or subtree pruning and cutting. And that's about where we were when the Ma paper came out this year.

As of the writing of Ma et al, no polynomial time algorithm was known to solve this problem. While I have found no formal statements that anyone believed at this point the problem was NP-complete, it seems likely it was suspected.

0.4 Ma et al, and the NP-proofs

As stated in the introduction, Ma et al provide a proof that the gene to species tree reconciliation problem is NP-hard. They actually provide proofs for several cases of the problem, and call them

Optimal Species Tree I (OST I) – in which there are n gene trees to be considered, and we consider only minimizing the duplication cost (the cost of interpolating gene duplication events), and

Optimal Species Tree II (OST II) – in which there are n gene trees to be considered, and we consider what they call the 'mutation cost', which is summed from the gene duplications and the gene losses, and

Optimal Species Tree III (OST III) – in which there is just one gene tree,

and they attempt to minimize just the duplication cost in formulating the species tree.

All of these problems are converted to decision problems, to which known NP-hard decision problems are reduced, to prove their NP-completeness.

I shall work through the proof for OST I in detail, and just describe the other two proofs briefly, as the proofs (particularly for OST I and II) are quite lengthy.

0.4.1 OST I NP-hard

The Ma et al proof that OST I is NP-hard is also the foundation from which they prove that OST is NP-hard.

The proof relies on a somewhat elaborate construction of a species tree for each node, and a species tree for each edge, for a graph instance on which INDEPENDENT-SET would be run.

Line tree and label construction

Ma et al start by constructing a large set of labels, and a set of $n+1$ line trees carrying these labels.

They start from an instance $G = (V, E)$ where $V = \{v_1, v_2, \ldots v_n\}$. They then let $N = 5n^3$.

For each v_i, they then introduce N labels $l_{ip}, 1 \leq p \leq N$, and line tree $T_i = L[l_{i1}, l_{i2}, \ldots, l_{iN}]$.

They then Introduce an extra N labels $l_{0p}, 1 \leq p \leq N$, and line tree $T_i = L[l_{01}, l_{02}, \ldots, l_{0N}]$.

This gives them, lease note, $(n+1)N$ labels, and $n+1$ line trees.

Vertex tree constructions (gene trees)

Next, they construct n gene trees, one for each vertex in the graph. They construct them as shown in the attached figure.

Note that each of these is highly similar, on its left, to that of each of its neighbours.

Edge tree constructions

Next, they construct $2|E|$ vertex trees, two for each vertex. These are constructed as show in figure [error].

Note that the trees for G_{ij} and G_{ji} are identical on the left, and different on the right.

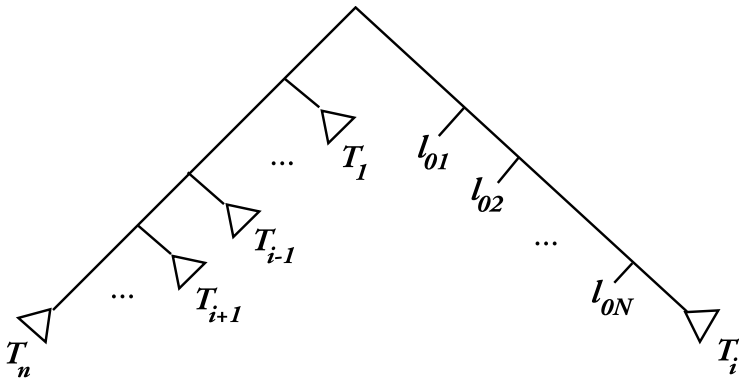

Figure 4: The vertex species tree construction

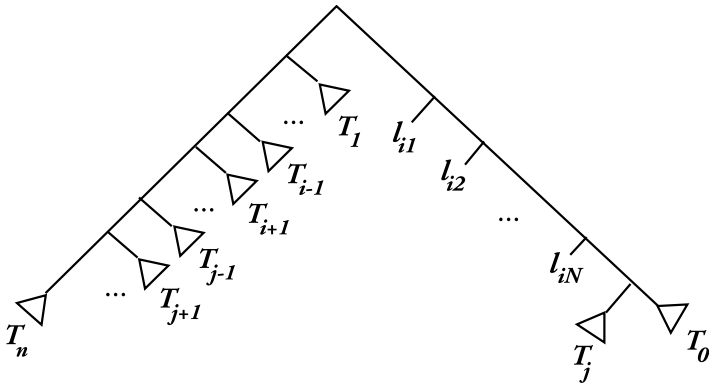

Figure 5: The edge species tree construction

They state without proof at this point that you can do this transformation in polynomial time. I won't belabor the point. It sure looks like you can. I've done it with pen and paper for small graphs, representing each of the line trees with ellipses and triangles as do they, and it's not difficult.

Lemma 4.2 from Ma et al

They then state the following Lemma: The graph G contains an independent set of size d if and only if there is a species tree S for all the gene trees G_{ij} and G_i constructed above with the duplication cost $c < (|E| + N - d + \frac{1}{2})N$.

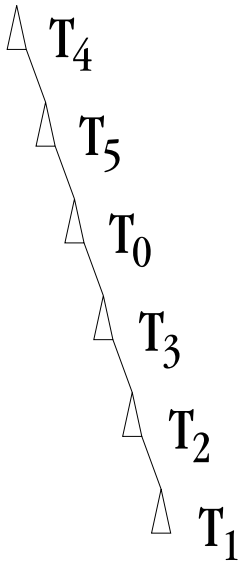

Figure 6: The species tree, for a five node graph

0.4.2 Species tree construction

To prove this lemma, they construct a species tree as follows:

(1) $S = L[l_{n1}, \ldots, l_{nN}, \ldots, l_{(d+1)1}, \ldots, l_{(d+1)N}, l_{01}, \ldots, l_{0N}, l_{d1}, \ldots, l_{dN}, \ldots, l_{11}, \ldots, l_{1N}]$

The following figure shows what this would look like for a five node graph.

Now Ma et al propose, let's assume G contains an independent set K of size d. They assume $V(K) = \{v_1, v_2, \ldots, v_d\}$.

They then assert, quite without explaining it, that for all $i \leq d$,

(2) $t_{dup}(G_i, S) = n - 1$

while for all $i > d$,

(3) $t_{dup}(G_i, S) = N + n - 1$

It's easy enough to see why this would be. For all $i \leq d$, there are no T-trees that contradict the strict order imposed by the construction in

which the T_0 tree is in the ancestral chain above all the T-trees from 1 to d. However, in all the trees $i > d$, this is the case, and you must propose N additional substitutions to corerect for it.

Next, they assess the costs of reconciling the edge graphs. It's clear that for all these graphs, if there's an independent set of size d, and all of v_1 to v_d are in it, for every pair G_ij and G_ji of these graphs, the duplication costs works out higher than N (one of the two will require such a transform), but less than N + 2n.

Then, they just add up the all the duplication costs, and get their result, thus proving the if half of the lemma.

Proof of Lemma 4.2 (only if)

The only if half of the Lemma 4.2 if and only if condition is proved through negation.

I won't work through this proof in great detail here, as it's largely parallel to the previous one, and uses a demonstration that if the set does not contain an independent set of size d, the positioning of the subtrees requires that c must be $(|E| + n - d + 1)N$, which contradicts the Lemma.

0.5 Remaining NP-hard proofs

Ma et al prove OST II NP-hard through a reduction from the cyclic ordering problem, a problem proved NP-complete in 1977. [2]. The proof

OST III is proved through a reduction from OST I, the proof of which is demonstrated above.

0.6 The heuristic method

Ma et al's proposed heuristic method has to do with a new metric they have devised for assessing the cost of reconciliation – the symmetric duplication cost, which can be calculated rapidly between gene trees. They report significantly improved search times with their approach, which, in contrast to earlier approach, does not start from a random gene tree (as the starting point for the gene tree) in the input set, but a gene tree selected for having the minimum symmetric duplication cost (done between this tree and each other tree in the input set, then summed).

0.7 Supplemental – NP-completeness of INDEPENDENT-SET through CLIQUE

I thought it appropriate, since the proof of NP-completeness of OST I relies on that of INDEPENDENT SET, to provide a brief proof of the NP-completeness of INDEPENDENT SET.

0.7.1 NP completeness of CLIQUE from 3-CNF-SAT

This is adapted from Cormen et al, 1004-5 [1].

The CLIQUE problem is: given an undirected graph $G = (V, E)$, a clique is a subset $V' \in V$ of vertices, each pair of which is connected by an edge in E. In other words, a clique is a complete subgraph of G.

The decision version of the CLIQUE problem is, given G and constant k, does G contain a clique of size k?

The proof for the NP-completeness of CLIQUE itself is given in Cormen et al, 1004, and is as follows:

(a) $CLIQUE \in NP$ because given the graph $G = (V, E)$, using set $V' \subseteq V$ as the certificate, we can check in polynomial time for each pair $u, v \in V'$, whether the edge (u, v) belongs to E.

(b) So if we can prove $3 - CNF - SAT \leq CLIQUE$, CLIQUE is NP-hard. Cormen et al demonstrate the following reduction:

Let $f = C_1 \wedge C_2 \wedge C_3 \wedge \ldots \wedge C_k$ be a boolean formula in 3-DNF with k clauses. For $r = 1, 2]ldotsk$ each clause C_r has exactly three distinct literals $l_{r1}, l_{r2}, andl_{r3}$.

Construct a graph G such that G has a clique of size k if and only if f is satisfiable.

We do this as follows - (a) for each clause $C_r = (l_{r1}l_{r2}l_{r3})$ in f we place a tripple of vertices v_{r1}, v_{r2}, v_{r3} into V. (b)We put an edge between two vertices v_{ir} and v_{js} if both of the following hold: v_{ir} and v_{sj} are in different triples, that is, $r \neq s$, and their corresponding literals are consistent, that is, l_{ir} is not a negation of l_{sj}.

Proof that this transformation is a reduction is: if f has a satisfying assignment, then each clause C_r contains at least on literal l_{ir} that is assigned 1, and each such literal corresponds to a vertex v_{ir}. Picking one such literal from each clause yields a set V' of k vertices. We claim that V' is a clique. For any two vertices $v_{ir}, s_{ej} \in V'$, where $r \neq s$, both corresponding literals are mapped to 1 by the given satisfying assignment, so they can't be complements. thus by the construction of G, the edge v_{ir}, v_{sj} is in E.

If G has a clique V' of size k, no edges in G connect vertices in the same triple, and so V' contains exactly one vertex per triple. We can assign 1 to each literal lir such that $v_{ir} \in V'$ without fear of assigning 1 to both a literal and its complement, since G contains no edges between inconsistent literals. Each clause is satisifed, so f is satisfied.

0.7.2 NP completeness of independent set through CLIQUE

Independent set $\in$ NP because given the graph $G = (V, E)$, using set $V' \subseteq V$ as the certificate, we can check in polynomial time for each pair $u, v \in V'$, whether the edge (u, v) belongs to E.

Given graph $G = (V, E)$, create its complement $H(V, E')$, following the following rule: for each vertex pair (u, v) in G, if edge (u, v) is in E, edge (u, v) is not in E', and for each edge (u, v) not in E, edge (u, v) is in E' If and only if G has a clique of size k, H has an independent set of size k, since the same k vertices in the cliquie in G that had a connection now have no connection, by the definition of H.

Bibliography

[1] Cormen, Thomas et al, Introduction to Algorithms. 2nd ed. MIT Press, 2001.

[2] Galil, Z. and N Megido. Cyclic Ordering is NP-complete. Theor. Comp. Sci., 5:179-182, 1977.

[3] Ma, Bin, et al. From Gene Trees to Species Trees. SIAM J. Comput., Vol 30, No 3, pp 729-752.

[4] Page, Roderic, and Michael A. Charleston. Reconciled trees and incongruent gene and species trees. DIMACS Seris in Discrete Mathematics and Theoretical Computer Science, 1997.

0.8 Web Resources

See also http://taxonomy.zoology.gla.ac.uk/rod/genetree/genetree.html re GeneTree.