A time and space complexity reduction for coevolutionary analysis of trees generated under both a Yule and Uniform model

Benjamin Drinkwater[1,*], Michael A. Charleston[*]

School of Information Technologies, University of Sydney

[a]*School of Information Technologies, J12, University of Sydney, 2006, Australia*

Abstract

The topology or shape of evolutionary trees and their unbalanced nature has been a long standing area of interest in the field of phylogenetics. Coevolutionary analysis, which considers the evolutionary relationships between a pair of phylogenetic trees, has to date not considered leveraging this unbalanced nature as a means to reduce the complexity of coevolutionary analysis. In this work we apply previous analyses of tree shapes to improve the efficiency of inferring coevolutionary events. In particular we use this prior research to derive a new data structure for inferring coevolutionary histories. Our new data structure is proven to provide a reduction in the time and space required to infer coevolutionary events. It is integrated into an existing framework for coevolutionary analysis and has been validated using both synthetic and previously published biological data sets. This proposed data structure performs twice as fast as algorithms implemented using existing data structures with no degradation in the algorithm's accuracy. As the coevolutionary data sets increase in size so too does the running time reduction provided by the newly proposed data structure. This is due to our data structure offering a logarithmic time and space complexity improvement. As a result, the proposed update to existing coevolutionary analysis algorithms outlined herein should enable the inference of larger coevolutionary systems in the future.

Keywords: Coevolution, Phylogeny, Tree Topology, NP-Hard

1. Background

Species often place selective pressures on one another driving the evolutionary process (Anderson and May, 1982). The study of these selective pressures and their impact on the overall evolutionary process is encompassed in the field of coevolution. Coevolution considers two or more species that have an evolutionary dependence on one another (Juan et al., 2008). This dependence may come in the form a mutalistic coevolutionary relationship where both species harmoniously coexist, both benefiting from one another, such as figs and their pollinator wasps (Cruaud et al., 2012). Alternatively, this evolutionary dependence may be in the form of a parasitic coevolutionary relationship, where the evolutionary changes in the host species aim to combat their invasive parasites, while the parasites evolve specific traits to exploit their host's defences, such as pocket gophers and their parasitic chewing lice (Hafner and Nadler, 1988).

Coevolutionary analysis is not only limited to these two systems, in fact coevolutionary analysis has the potential of providing insight into a number of other biological phenomena such as mimicry between species (Cuthill and Charleston, 2012), biogeography (Ronquist, 1997), host / pathogen interactions (Mu et al., 2005), predator / prey networks (Brodie et al., 2002), herbivore / plant dynamics (Fox, 1981), and inferring species trees from gene trees (Page and Charleston, 1997). Although encompassing a diverse range of

*Corresponding author

Email address: benjamin.drinkwater@sydney.edu.au (Benjamin Drinkwater)

evolutionary systems, coevolutionary analysis is often considered in terms of an independent and dependent evolutionary network, the host and parasite respectively (Charleston and Perkins, 2006).

Macro scale coevolutionary analysis considers the host (H) and parasite (P) as a pair of bifurcating phylogenetic trees with known associations (φ) between their extant taxa (Libeskind-Hadas and Charleston, 2009). Coevolutionary analysis often restricts the number of associations such that each parasite may only infect a single host species, while host species may be infected by any number of parasite species (Poulin, 2011). This restriction is the form of coevolutionary analysis considered herein. The construction of a coevolutionary system in terms of H, P and φ is often visualised as a Tanglegram as seen in Figure 1.

One common methodology for macro scale coevolutionary analysis is the technique of *cophylogeny mapping*, which aims to reconstruct the evolutionary history of the parasite with respect to the host tree (Charleston, 2003). This is achieved by mapping the parasite tree into the host tree where the initial associations (φ) are conserved (Jackson, 2005). This process requires four biological events, *codivergence*, *divergence*, *host switch* and *loss*, which represent all permitted evolutionary interactions between the host and the parasite (Ronquist, 1995).

A *codivergence* event is a concurrent divergence of both the host and parasite species. A high concentration of codivergence events is seen as a strong signal that the host and parasite have a coevolutionary dependence (Page, 2002). A *duplication*, conversely, is an independent divergence of the parasite which is not in response to a divergence of its host species. Following the parasite's divergence both new parasite species continue to infect the initial host (Charleston, 2003). A *host switch*, similar to a duplication, is independent of the evolutionary changes in its host species. Rather than both new parasite species continuing to infect a common host, as is the case for a duplication event, only one of the new parasites continues to infect the initial host, while the second parasite species establishes itself on a new independent host species (Charleston and Perkins, 2003). These three events are *divergence events* and are used to infer all possible parasite evolutionary events with respect to the host. A *loss* event, unlike a divergence event, is the case where the parasite fails to codiverge (Paterson et al., 2003). Following the host divergence only one of the two new host species remains infected by the initial parasite species.

The cophylogeny reconstruction problem aims to apply the process of cophylogeny mapping to recover a minimum cost map ($\Phi(P)$) using the four recoverable events. The recovery of such a map has been proven to be NP-Hard (Ovadia et al., 2011). An example of a minimum cost map, $\Phi(P)$, which includes all four permitted events can be seen in Figure 1.

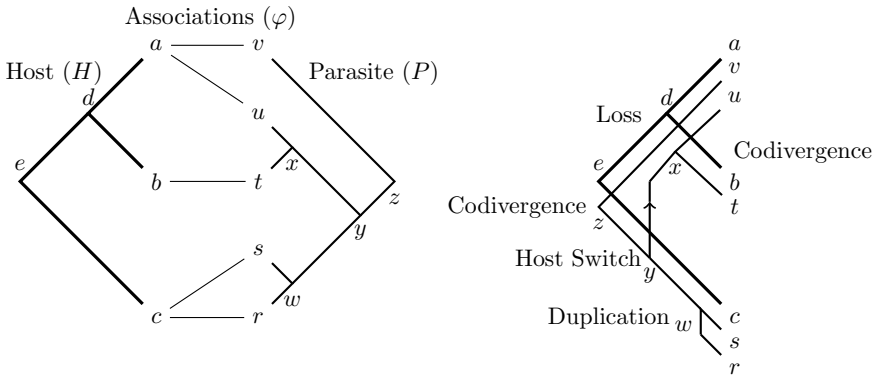

Figure 1: Tanglegram and its corresponding map. Example of a Tanglegram (H, P, φ) (left) and a minimum cost map ($\Phi(P)$) constructed from (H, P, φ) (right)

The computational intractability of the cophylogeny reconstruction problem is due to the exponential number of node orderings possible for a bifurcating tree and the exponential number of host switch events that this gives rise to (Libeskind-Hadas and Charleston, 2009). Due to the difficult nature of this problem,

significant research has been undertaken to approximate the cophylogeny reconstruction problem. Primarily, research has focused on imposing restrictions on the relative node ordering of the host or parasite tree (Conow et al., 2010; Libeskind-Hadas and Charleston, 2009; Merkle and Middendorf, 2005; Merkle et al., 2010).

One approach applied in both Tarzan (Merkle and Middendorf, 2005) and CoRe-PA (Merkle et al., 2010) is to ignore the relative ordering of the nodes in the parasite tree, allowing for a map to be recovered in quadratic time (Yodpinyanee et al., 2011). The issue with this approach, however, is that it has the potential of inducing *time inconsistent* solutions (Doyon et al., 2011b).

A time inconsistent map is a solution where the order of evolutionary events inferred from the map contradicts the order of evolutionary events evident in the parasite's phylogenetic history. Such a result is deemed to be biologically infeasible (Doyon et al., 2011b).

As a result of Tarzan and CoRe-PA's potential for recovering biologically infeasible solutions, recent research has focused primarily on traversing the exponential number of fixed node orderings using a meta-heuristic (Conow et al., 2010; Doyon et al., 2011a,b; Libeskind-Hadas and Charleston, 2009). This approach has been shown to be effective, as solving the cophylogeny reconstruction problem where the internal node ordering is fixed can be solved in polynomial time (Libeskind-Hadas and Charleston, 2009). The initial implementation, known as Node Mapping, ran in $O(n^7)$ (Conow et al., 2010), but has been subsequently replaced by Edge Mapping (Yodpinyanee et al., 2011), Slicing (Doyon et al., 2011a,b) and Improved Node Mapping (Drinkwater and Charleston, 2014a), all running in cubic time. Of this set of cubic time algorithms, Edge Mapping and Improved Node Mapping offer the best space complexity as both require only quadratic space (Yodpinyanee et al., 2011; Drinkwater and Charleston, 2014a) as opposed to Slicing which requires cubic space (Doyon et al., 2011a,b).

These algorithms are able to run in cubic time by leveraging certain topological properties of bifurcating trees. This is evident when considering each algorithm's computational complexity increase when the host phylogeny is a directed acyclic graph rather than a bifurcating tree. For this problem instance Node Mapping runs in $O(n^7)$ (Conow et al., 2010) while Edge Mapping and Slicing currently cannot solve this more difficult problem (Doyon et al., 2011a,b; Yodpinyanee et al., 2011). This problem is even more difficult if the parasite phylogeny is a directed acyclic graph, with no published algorithm to date able to solve this problem in polynomial time even if the internal node ordering is known (Libeskind-Hadas and Charleston, 2009).

While a number of specialised characteristics for bifurcating trees have been applied that reduce the computational complexity of cophylogeny mapping, such as leveraging constant time queries for recovering the most recent common ancestor (Doyon et al., 2011a), and utilising that the number of edges in a tree is $O(n)$ (Yodpinyanee et al., 2011), as far as we know no prior research has considered tree topology as a means to decrease the computational complexity of the cophylogeny reconstruction problem.

Evolutionary trees have long been identified as often having an unbalanced structure (Gregory, 2008). Modelling this unbalanced nature dates back at least to Yule's proposed model in 1924. While no synthetic model can capture the diversity of evolutionary behaviour, it has been shown that the topology of trees produced by the evolutionary process converges between the topology of trees produced by the *Yule* and *Uniform* models (Aldous, 2001; Mooers and Heard, 1997).

The Yule model, also known as the equal-rates-Markov model (Cavalli-Sforza and Edwards, 1967; Harding, 1971), is a synthetic bifurcating tree generation model which applies a continuous-time pure birth process starting with a single node (Aldous, 1996). Under this model the length of each branch in the phylogenetic tree has no bearing on the likelihood of speciation (Harding, 1971), where each leaf exists for a random period of time before speciating (Aldous, 1996). This process continues until the tree has n leaves where the model assumes that all speciation events are bifurcating and two species may not interbreed (Harding, 1971). This model can be extended further to consider extinction events, by modifying the rate of divergence (Aldous, 2001). The Yule model is often applied to simulate evolutionary systems as this model generates trees with a topology which represents the most balanced phylogenetic trees (Aldous, 2001).

The Uniform model, also known as proportional-to-distinguishable arrangements (PDA), is a synthetic tree generation model which produces trees through uniform sampling of all possible tree topologies (Rosen, 1978). This is not an explicit model of the evolutionary process, nor does this model grow trees (Mooers and Heard, 1997). Rather, the trees generated by the Uniform model are constructed by sampling uniformly from trees with n leaves (Aldous, 2001). Although this model does not simulate the evolutionary process

directly, this model is often used as it appears to provide a bound for the most unbalanced phylogenetic trees (Aldous, 1993, 1991). As a result, Yule and Uniform trees provide a topological bound for expected evolutionary data (Aldous, 2001)

This research proposes a new data structure for cophylogeny analysis using Improved Node Mapping. This data structure is shown to recover an optimal map, $\Phi(P)$, in subquadratic space, where the topologies of the host and parasite phylogenies are bounded by the topology of trees produced under the Yule and Uniform models. This topological boundary is in the sense that the evolutionary trees' height and degree of balance are bounded by these synthetic models. This result is then used to prove a subcubic time complexity bound for Improved Node Mapping. In both cases the time and space complexity bound reductions presented herein are shown to be in terms of a logarithmic factor.

2. Methodology

The most space efficient algorithms for solving the cophylogeny reconstruction problem where the internal node ordering is fixed require quadratic storage. The space complexity bound is due to the size of the dynamic programming tables which store linear subsolutions, where each subsolution consists of up to a linear number of mapping sites. A subsolution in this context is a list of possible mapping sites for the parasite node in question.

In this paper we prove that the traditional dynamic programming table used for this problem can be replaced to allow for a space complexity reduction. In particular, this work proves that when the host and parasite phylogenies' structure, in the sense of balance and height, is bound between the topology of trees produced by the Yule and Uniform model (the majority of evolutionary data) (Aldous, 2001), that the proposed method provides a logarithmic factor reduction in the required space to solve the cophylogeny reconstruction problem optimally. This result, in turn, provides a logarithmic reduction in the time complexity as well.

The presented method may, at a later stage, be extended to Edge Mapping, however, this was not considered herein, as Edge Mapping also requires quadratic storage for preprocessing (Yodpinyanee et al., 2011), which would negate the space efficiency gains in terms of a worst case space complexity bound. This is not the case for the Improved Node Mapping algorithm, however, due to its preprocessing requirement of only linear time and space (Drinkwater and Charleston, 2014a).

2.1. Proposed data structure to improve the space complexity for the Improved Node Mapping algorithm

Improved Node Mapping in its current form incrementally maps each parasite node into the host phylogeny from the leaves up to the root. Under this approach, the first pass of the algorithm maps the parasite leaves to their respective host leaves based on the input association set, (φ). Following this step, the internal parasite nodes are incrementally mapped into the host, if and only if, their children have already been mapped.

This approach therefore, lends itself to being implemented where all the nodes from each level of the parasite tree are processed before progressing to the next level (Conow et al., 2010). A node's level is the height of the node in the tree, that is, the length of the longest path between the current node and its descendents. Therefore, the leaves of the parasite tree are at level 0.

It has previously been shown that the subsolutions constructed for the leaves of the parasite tree, subsolutions at level 0, only require $O(1)$ space (Drinkwater and Charleston, 2014a). As the algorithm continues to map parasite nodes into the host tree, the required space incrementally increases for each level of the parasite tree. The amount of space increases until it will require $O(n)$ space for the set of mapping sites for each subsolution (Drinkwater and Charleston, 2014a). The proposed data structure takes advantage of the rate at which the number of mapping sites increases, to allow for a subquadratic space solution for the cophylogeny reconstruction problem where the internal node ordering is fixed.

The proposed data structure replaces the currently used two dimensional matrix for storing each subsolution with an array of lists. The array itself is $O(n)$ in size, where each element represents a single parasite node. The lists stored in each array element consists of a list of unique mapping locations for the

parasite node (p_i) in the host, which is bounded by $O(n)$, the number of internal nodes in the host tree. The worst case reconstruction using this data structure offers the same worst case space complexity bound as the dynamic programming matrix used by both Edge Mapping and Improved Node Mapping.

The unique aspect of this data structure compared to a two dimensional array is that each list stored in the array is not fixed to the size of the host tree ($2n-1$). This follows from the differences between the representation of a graph as either an adjacency matrix or an adjacency list. For sparse graphs an adjacency list has the potential to offer a significant reduction in the space complexity, while the worst case complexity bound in both cases is the same. Similarly, this work proves that phylogenetic trees which are bounded by the topologies produced by the Yule or Uniform models have sufficiently many subsolutions which require less than $O(n)$ mapping sites, such that the overall storage requirement is subquadratic.

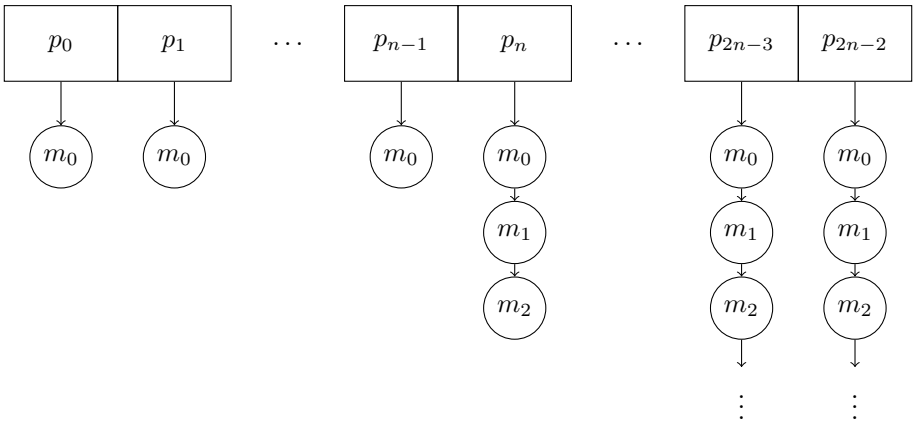

Figure 2: The proposed data structure. This data structure is designed to minimise the space complexity of the Improved Node Mapping algorithm, by replacing the dynamic programming matrix previously used. Applying the proposed data structure requires no additional changes to the algorithm first proposed by Drinkwater and Charleston, 2014a to achieve the logarithmic time and space complexity reduction

Figure 2 is a visualisation of the proposed data structure where the subsolutions $(p_0, \ldots, p_{2n-2})$ have been added to the array in increasing order based on their height in the tree, where $(p_0, \ldots, p_{n-1})$ are the subsolutions for the leaves of the parasite tree while p_{2n-2} stores the root's subsolution. What is important to note is that for the first n $(0, ..., n-1)$ subsolutions only one mapping site is stored; the corresponding leaf node in the host tree for which the parasite, p_i, is associated. For the remaining $(n-1)$ internal nodes there are up to $O(n)$ possible mapping sites. Initially, however, the subsolutions will have fewer than $O(n)$ mapping sites, such as the mapping sites generated for the parasite nodes at level one, which will only have three possible mapping sites, as discussed in the following section. This is represented by the subsolution stored at index n, (m_0, m_1, m_2). The number of mapping sites will strictly increase (See Lemma 1) as the index increases until the number of required sites is bound by $O(n)$; the case where each subsolution may be mapped to all nodes in the host tree.

3. Proof of a lower time and space complexity bound

3.1. Required storage to recover an optimal map

The recovery of a map where internal node ordering is fixed has previously required the storage of linear subsolutions, where each subsolution consists of up to a linear number of mapping sites (Yodpinyanee et al., 2011). Although previous research has identified that mapping algorithms may not require linear mapping sites to be considered (Drinkwater and Charleston, 2014a), no prior analysis has proven any bound less than $O(n)$.

To prove that the required number of mapping sites is less than $O(n)$ for trees produced under the Yule and Uniform models, we first identify the number of mapping sites required at each level of a bifurcating tree, which allows for a map to be incrementally reconstructed. To establish this bound we consider the number of mapping sites for the leaves of the parasite phylogeny. These mapping sites represent the direct mapping formed by the associations between the parasite and host phylogenies, which is always one by our initial assumption. These are the first set of subsolutions reconstructed as the Node Mapping algorithm reconstructs an optimal map from the leaves to the root (Libeskind-Hadas and Charleston, 2009).

The next set of subsolutions to recover are for the parasite's internal nodes at level one. These are the internal nodes where both children have previously been mapped. These nodes have three possible mapping sites, which include the two minimum cost host switch events and either a codivergence or a duplication event (Drinkwater and Charleston, 2014a). Only one of these two events needs to be considered, as the optimal codivergence and duplication events both map to the same host node, the most recent common ancestor (Doyon et al., 2011a), and therefore only the most cost efficient solution needs to be stored to recover an optimal map.

The trend identified for level one can be used to compute the storage requirement for the nodes at level two. In this case, however, there may be potentially more than one optimal mapping site for the left and right child of the parasite node in question. We therefore must consider each mapping site for the left child and compare it to all other mapping sites for the right child. Each mapping site pair may have up to three possible mapping sites, giving rise to the recurrence relation, a_i; the maximum number of mapping sites required for the subsolution of a parasite node at level i.

$$\begin{aligned} a_0 &= 1 \\ a_1 &= 3 \\ a_i &= 3 \times (a_{i-1})^2 \text{ for all } i \geq 1 \end{aligned} \tag{1}$$

It is important to note that the recursive function in Equation (1) is the representation of the worst case, as it assumes that the children have an equivalent worst case cost and that each mapping site pair gives rise to a unique set of mappings. This can occur in the case where the host and parasite phylogenies are congruent balanced binary trees. As a tree becomes more unbalanced, however, there will be an increased number of cases where the left and right child will not have an equal number of mapping sites, as there are fewer cases where both children reside at the previous level. This results in a reduced number of mapping sites and a lower rate of growth for the function a_i. Therefore, Equation (1) represents the worst case growth for the number of mapping sites for all bifurcating trees.

Lemma 1. *The storage requirement for each subsolution, where i is the current level in the parasite phylogeny for which the subsolution is associated, is $O\left(3^{(2^i-1)}\right)$ for all $i \geq 1$.*

Proof.

$$\begin{aligned} a_i &= 3 \times (a_{i-1})^2 \\ a_i &= 3^3 \times (a_{i-2})^2 \\ a_i &= \dots \\ a_i &= 3^{2^i-1} a_0 \\ a_i &= 3^{2^i-1} \end{aligned} \tag{2}$$

Lemma 1 provides a closed form function for a_i and can therefore be used to provide a tighter bound for the required storage, $f(i)$, to solve the cophylogeny reconstruction problem. When combined with the existing worst case bound for the number of mapping sites required to solve this problem optimally, $O(n)$, (Drinkwater and Charleston, 2014a) we get:

$$f(i) = \begin{cases} 1 & \text{if } i = 0 \\ \min(3^{(2^i-1)}, n) & \text{if } i \geq 1 \end{cases} \tag{3}$$

This result is of more interest if we can explicitly identify all values of i where $3^{(2^{(i)}-1)} < n$. If this value is in terms of n then there is the possibility of reconstructing an optimal map in subquadratic space. We show that there is such a strict set of values where $3^{(2^{(i)}-1)} < n$, defined as:

Lemma 2. *$3^{(2^i-1)} \leq n$ for all $i \leq \log_2(\log_3 n + 1)$*

Proof.

$$\begin{aligned} 3^{(2^i-1)} &\leq n \\ 3^{(2^i)} &\leq 3n \\ 2^i &\leq \log_3 n + 1 \\ i &\leq \log_2(\log_3 n + 1) \end{aligned} \tag{4}$$

Using Lemma 2 we redefine $f(i)$ as.

$$f(i) = \begin{cases} 1 & \text{if } i = 0 \\ 3^{(2^i+1)} & \text{if } 0 < i \leq \log_2(\log_3 n + 1) \\ n & \text{if } i > \log_2(\log_3 n + 1) \end{cases} \tag{5}$$

To derive the total storage requirement for the proposed data structure requires that the number of nodes at each level is known along with the height of the tree. If we assume a function, $g(i)$, exists which can calculate the number of nodes at each level and that h is the height of the tree, then we can define the storage requirement for solving the cophylogeny reconstruction problem using the proposed data structure as a geometric series (Equation (6)).

$$\sum_{i=0}^{h} (f(i) \times g(i)) \tag{6}$$

The height of all trees is bound between $\log n$ and $(n-1)$. For this proof we will show that if in the case where the height is $(n-1)$ and the function $f(i)$ as defined in Equation (5) is applied, both of which will over count the number of required mapping sites, that the space required to solve the cophylogeny reconstruction problem is subquadratic in size.

3.2. Defining the Shape of Yule and Uniform Trees

To derive the number of nodes at each level for trees produced under a Yule and Uniform model requires the introduction of a random variable L and the function $C(x)$. L_i is the number of nodes at a level i in a tree, where L_0 is the number of leaves in the tree, n. $C(j)$ is the number of cherries in a tree with j leaves, where a cherry is a pair of leaf nodes which are adjacent to a common ancestor node (McKenzie and Steel, 2000).

As trees produced under the Yule and Uniform models have previously been shown to append levels within their respective models at a constant rate (McKenzie and Steel, 2000), it is possible to construct a recurrence relation for the number of nodes expected at each level within a constructed tree in terms of L and the function $C(x)$, as seen in Equation (7).

$$\begin{aligned} L_1 &= C(L_0) \\ L_2 &= C(L_0 - L_1) \\ L_3 &= C(L_0 - L_1 - L_2) \\ L_i &= C(L_0 - \sum_{j=1}^{i-1} L_j) \end{aligned} \tag{7}$$

Equation (7) is of particular value as the initial number of cherries for trees constructed under a Yule or Uniform model has been proven by McKenzie and Steel to be $\frac{n}{3}$ and $\frac{n}{4}$ respectively (McKenzie and Steel, 2000). Therefore, using this prior result, we prove the following two Theorems (see Proof for Theorem 1 and Proof for Theorem 2).

Theorem 1. *The expected number of internal nodes at each level of a tree generated under a Yule process is* $\frac{2^{(i-1)}n}{3^i}$*, where* n *is the number of leaves in the tree and* i *is the level for all* $i \geq 1$*.*

Theorem 2. *The expected number of internal nodes at each level of a tree generated under a Uniform process is* $\frac{3^{(i-1)}n}{4^i}$*, where* n *is the number of leaves in the tree and* i *is the level for all* $i \geq 1$*.*

Theorems 1 and 2 provide a closed-form function for $g(i)$ from Equation (6). Combining these two results with the storage requirement function, $f(i)$, provides a space complexity bound for trees produced under the expected Yule and Uniform models.

3.3. Achieving a new lower bound for space complexity

From the prior results we are able to define a pair of geometric series for the space required to solve the cophylogeny reconstruction problem for trees constructed under either the Yule or Uniform model. These definitions provide a worst case bound for the space requirements in both cases.

Definition 1. *The number of mapping sites required to recover an optimal map,* $\Phi(P)$*, for a tree constructed under an expected Yule process is* $O\left(n \sum_{i=0}^{n-1} \frac{2^{(i-1)}f(i)}{3^i}\right)$*, where* $f(i)$ *defines the total number of nodes which need to be stored at each level.*

Definition 2. *The number of mapping sites required to recover an optimal map,* $\Phi(P)$*, for a tree constructed under an expected Uniform process is* $O\left(n \sum_{i=0}^{n-1} \frac{3^{(i-1)}f(i)}{4^i}\right)$*, where* $f(i)$ *defines the total number of nodes which need to be stored at each level.*

As a closed-form solution exists for $f(i)$ (see Equation (5)), the geometric series from Definitions 1 and 2 can be expanded to provide a worst case space complexity result for both Yule ($\Upsilon(n)$) and Uniform ($\Psi(n)$) trees.

$$\Upsilon(n) = O\left(n + n \sum_{i=1}^{\lfloor \log_2 (\log_3 n+1)\rfloor} \frac{2^{(i-1)}3^{(2^i-1)}}{3^i} + n \sum_{i=\lfloor \log_2 (\log_3 n+1)\rfloor+1}^{n-1} \frac{2^{(i-1)}n}{3^i}\right) \tag{8}$$

$$\Psi(n) = O\left(n + n \sum_{i=1}^{\lfloor \log_2 (\log_3 n+1)\rfloor} \frac{3^{(i-1)}3^{(2^i-1)}}{4^i} + n \sum_{i=\lfloor \log_2 (\log_3 n+1)\rfloor+1}^{n-1} \frac{3^{(i-1)}n}{4^i}\right) \tag{9}$$

By simplifying Equations (8) and (9) a new lower bound for the space complexity required for the cophylogeny reconstruction problem, where the internal node ordering is fixed, may be established. The proofs for both reductions can be seen in the Appendix (Proof for Theorem 3 and Proof for Theorem 4), which give rise to the following Theorems.

Theorem 3. *The required space for solving the cophylogeny reconstruction problem where the internal node ordering is fixed for trees constructed under the expected Yule process is bounded by* $O\left(\frac{n^2}{(\log n)^{\log 3-1}}\right)$

Theorem 4. *The required space for solving the cophylogeny reconstruction problem where the internal node ordering is fixed for trees constructed under the expected Uniform process is bounded by* $O\left(\frac{n^2}{(\log n)^{2-\log 3}}\right)$

Theorems 3 and 4 prove that the proposed data structure provides the first subquadratic space algorithm which solves the cophylogeny reconstruction problem optimally when the internal node ordering is fixed for any tree topology. While the proposed algorithm offers no worst case improvement for all trees, this result shows that the proposed update to the underlying data structure of the Improved Node Mapping algorithm provides a logarithmic reduction to the required space for the two models which bound the topology of evolutionary data (Aldous, 2001).

3.4. A new running time complexity bound

Theorems 3 and 4 provide a new worst case bound for space required to solve the cophylogeny reconstruction problem, where the internal node ordering is fixed. In this section we apply these two results to provide a reduction to the cubic time complexity bound faced by Improved Node Mapping.

A subquadratic space requirement is achieved for trees produced by the Yule and Uniform models by reducing the number of mapping sites stored for each subsolution. This result asserts that the total number of mapping sites which are stored for each subsolution is sublinear in size. From Theorems 3 and 4 we can derive the average number of mapping sites stored for each subsolution for trees produced by both the expected Yule ($\psi(n)$) and Uniform ($\upsilon(n)$) models as:

$$\psi(n) = \frac{n}{(\log n)^{\log 3 - 1}} \tag{10}$$

$$\upsilon(n) = \frac{n}{(\log n)^{2 - \log 3}} \tag{11}$$

The functions for the number of mapping sites stored for trees produced by the Yule ($\psi(n)$) and Uniform ($\upsilon(n)$) models is constructed by dividing the total storage requirement for each tree by n, the number of subsolutions required to solve the cophylogeny reconstruction problem using Improved Node Mapping. As a result, $\psi(n)$ and $\upsilon(n)$ are the average number of mapping sites stored for each subsolution.

Node Mapping (Drinkwater and Charleston, 2014a) solves the cophylogeny reconstruction problem by executing n steps, where for each step a new subsolution is constructed. This process requires the generation of all the mapping sites possible from all the permutations of the previously recovered mapping sites for both children. Therefore, the total worst case running time for Yule and Uniform models is defined as:

$$\begin{aligned} & O(n(\psi(n))^2) \\ & = O\left(n \times \left(\frac{n}{(\log n)^{\log 3 - 1}}\right)^2\right) \\ & = O\left(\frac{n^3}{(\log n)^{2\log 3 - 2}}\right) \end{aligned} \tag{12}$$

$$\begin{aligned} & O(n(\upsilon(n))^2) \\ & = O\left(n \times \left(\frac{n}{(\log n)^{2 - \log 3}}\right)^2\right) \\ & = O\left(\frac{n^3}{(\log n)^{4 - 2\log 3}}\right) \end{aligned} \tag{13}$$

This result provides approximately a $\log n$ speed up for Improved Node Mapping. In the next section we demonstrate the improvements offered by the proposed update to the underlying data structure for the Improved Node Mapping algorithm in practice, by comparing its runtime over both synthetic and real data sets to an implementation of Improved Node Mapping algorithm using a standard dynamic programming matrix.

4. Discussion and Analysis

In the previous section we presented a reduction in the worst case theoretical bound for the time and space complexity required to solve the cophylogeny reconstruction problem, where the host and parasite phylogenies are bound by the expected topology of trees produced by the Yule and Uniform models. In this section we compare two implementations of the Improved Node Mapping algorithm (Drinkwater and Charleston, 2014a) to demonstrate how these theoretical bounds translate in practice. The first applies the traditional dynamic programming matrix while the second is implemented using the newly proposed data structure consisting of an array of lists of mapping sites.

While tools such as Jane (Conow et al., 2010; Yodpinyanee et al., 2011) and the implementation of Slicing by Doyon *et al.* (Doyon et al., 2011a,b) are also able to solve the cophylogeny reconstruction problem, this analysis was constrained to only consider Improved Node Mapping, as this simulation aims to evaluate the complexity improvements offered by the proposed data structure. As Edge Mapping (Jane) and Slicing are implemented in such a way that the newly proposed data structure does not lend itself to reducing their worst case complexity, they were excluded from this evaluation.

This analysis focuses on demonstrating the running time reduction offered by the newly proposed data structure, as this is the limiting factor in the analysis of larger coevolutionary systems (Drinkwater and Charleston, 2014b). The simulation was therefore constructed where both algorithms were implemented using a common code base for cophylogenetic reconstruction, where the data structure used to store mapping sites was the only code not shared by each approach. Implementing both algorithms in this manner ensured that any complexity differences were due to the data structure applied.

Both implementations of the Improved Node Mapping algorithm were evaluated using two data sets. The first was an existing set of synthetic coevolutionary systems produced using CoRe-Gen (Cophylogeny Generation Model) (Keller-Schmidt et al., 2011). This data set has been previously used for coevolutionary analysis (Rosenblueth et al., 2012; Liu et al., 2014) and contains 1000 coevolutionary instances where both the host and parasite phylogenies were generated using the Yule model. Of the 1000 data sets, 47 contain trees with only a single node. These data sets were excluded, consistent with previous analysis (Drinkwater and Charleston, 2014b).

The second data set contains 102 previously published coevolutionary instances (Drinkwater and Charleston, 2014b). This set includes evolutionary data from a number of fields including but not limited to, pathogens and their hosts (Charleston and Robertson, 2002), plant-insect relationships (McLeish et al., 2007), the evolutionary dependencies between plants and fungi (Refrégier et al., 2008), parasitic (Page et al., 2004) and mutualistic (Jackson, 2004) coevolution, and biogeography (Badets et al., 2011). The aim of this data set is to ensure that the running time improvements offered by the newly proposed data structure are consistent across the full spectrum of coevolutionary instances.

Figure 3 is a plot of the running time required to approximate each of the 953 synthetic data sets. Each data point in the plot is the median running time of 100 replicates of the genetic algorithm used to approximate the cophylogeny reconstruction problem. The genetic algorithm in this case is configured to execute 10000 iterations of the Improved Node Mapping algorithm over a subset of the exponential number of fixed node orderings. The reported running times for each algorithm demonstrate that the newly proposed data structure is consistently faster than the previous algorithm with a median reduction in the running time over 953 data sets of 46%.

The plot of all synthetic data sets as seen in Figure 3 displays a number of outliers which show little to no improvement (less than 20%) when applying the proposed data structure. Further analysis of this subset identified that in all cases these coevolutionary instances included a tree, either the host or parasite, with fewer than five leaf nodes.

As a result we filtered the synthetic data to only include coevolutionary instances where both the host and parasite trees have at least five leaf nodes. This data set includes 748 synthetically generated coevolutionary instances, which can be seen plotted in Figure 4. For this filtered set the median reduction in the running time was 52%.

Both Figures 3 and 4 provide a compelling argument that as the number of nodes in a coevolutionary instance grows, so too does the reduction in running time when applying the proposed data structure.

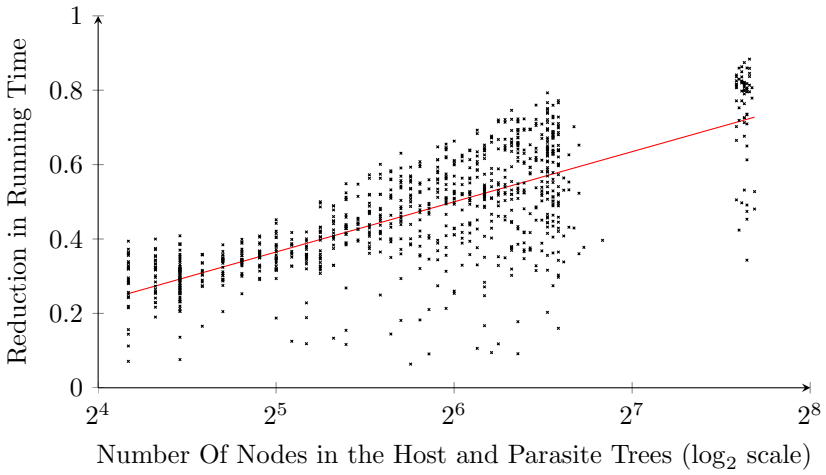

Figure 3: Running time analysis for Synthetic Data Set. The running time reduction offered by the Improved Node Mapping algorithm when implemented using the proposed data structure when run over the synthetic data set which contains 953 coevolutionary systems. Linear regression analysis clearly shows that as the size of the coevolutionary instance increases so too does the reduction in the running time.

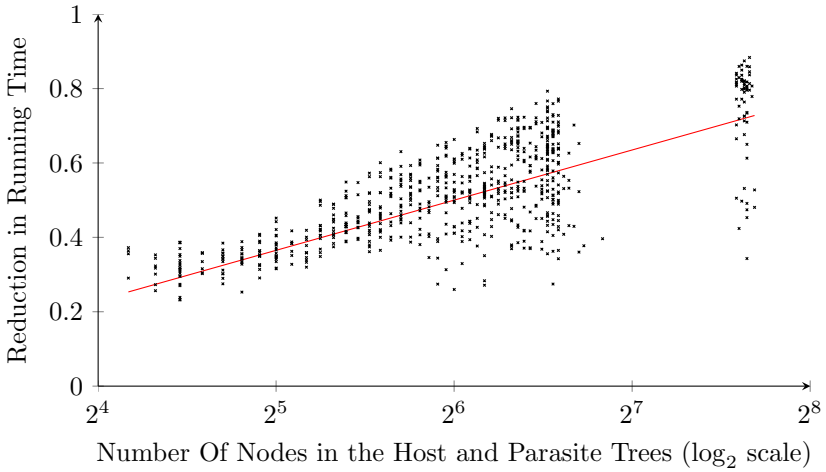

Figure 4: Running time analysis for filtered Synthetic Data Set. The running time reduction offered by the Improved Node Mapping algorithm when implemented using the proposed data structure when run over the filtered synthetic data set. This data set has discarded all coevolutionary instances where either the host or parasite tree contained less than 5 leaves. This filtered set consists of 748 synthetically generated coevolutionary systems. Linear regression analysis clearly shows that as the size of the coevolutionary instance increases so too does the reduction in the running time.

The differences in the number of outliers between Figure 3 and Figure 4 demonstrate, however, that for sufficiently small trees the improvements offered by the proposed data structure are potentially offset.

In the case of the real data set, a similar trend as seen in the filtered set of synthetic data was evident, as shown in Figure 5. As with the synthetic data set, 100 replicates were run for each coevolutionary instance, where the median running time was recorded on the presented plot. Over the 102 data sets there was a median reduction in the running time for the algorithm using the newly proposed data structure of approximately 48%.

Due to the change in the underlying data structure of the Improved Node Mapping algorithm, it is important to verify that the running time improvement has not come at the expense of the algorithm's accuracy. While we proved that all mapping sites required to recover an optimal map are recorded in the

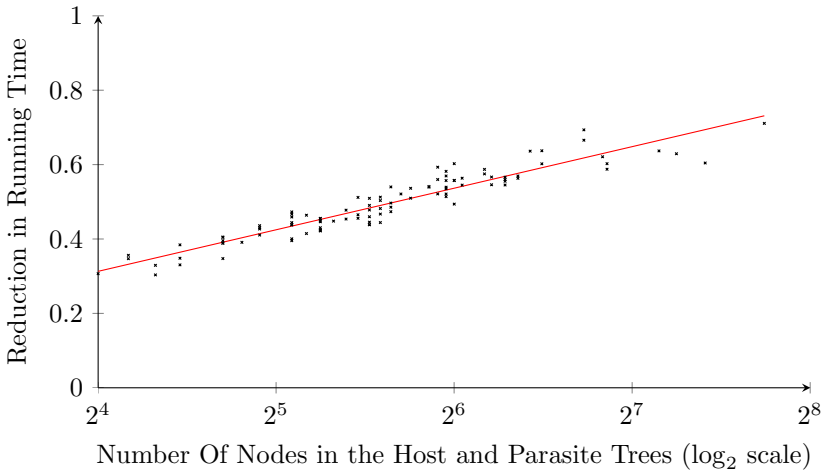

Figure 5: Running time analysis for Biological Data Set. The running time reduction offered by the Improved Node Mapping algorithm when implemented using the proposed data structure when run over the real data set which contains 102 previously published coevolutionary systems. Linear regression analysis clearly shows that as the size of the coevolutionary instance increases so to does the reduction in the running time.

proposed data structure, we also confirm that there was no degradation to its accuracy in practice, by comparing the resultant map costs generated for all real and synthetic coevolutionary instances. As both algorithms are run within a metaheuristic framework, a genetic algorithm for this experiment, it is expected that there will be a slight variation for the cost of the recorded maps. This variation, however, was expected to be minimal as both algorithms were run for the same number of iterations with equal population sizes.

Over the synthetic data set there was a 0.001% difference in the reported costs across all 953 data sets, while over the real data set there was a 0.003% difference in the reported costs across the 102 data sets. In both cases the newly proposed data structure performed marginally worse, although this minimal difference showed no statistical significance.

This demonstrates that both algorithms consistently converge on cophylogeny mapping with an equivalent cost, while the algorithm applying the newly proposed data structure runs in almost half the time. This reduction grows as the coevolutionary systems considered grow in size, as the reduction achieved using the proposed data structure is in terms of a logarithmic factor. This was evident when comparing the median running time reduction for the largest coevolutionary systems for both the real and synthetic data sets. For the synthetic data set the median decrease in the running time for all coevolutionary systems with at least 100 nodes was 78%. This subset of instances had a median size of 198 nodes. For the real data set the median running time reduction for all coevolutionary systems with at least 100 nodes was 63%. This subset of instances had a median size of 129 nodes. In both cases this was a significantly larger reduction compared to the complete data sets which only showed a reduction of 46% and 48% respectively, providing further evidence of a logarithmic reduction in the running time of the Improved Node Mapping algorithm.

5. Conclusion

This research emphasises the importance of data structure selection in the development of algorithms for bioinformatics applications, particularly for those processing phylogenetic data, and demonstrates the benefits possible when targeting the topology of phylogenetic data when designing new, or optimising existing algorithms. This research proves that not only is a reduction in running time possible when optimising algorithms to take advantage of the evolutionary data's topology, but that such a reduction can be logarithmic in terms of the size of the coevolutionary instance. As a result, the presented method provides the ability for coevolutionary analysis of larger coevolutionary systems than has previously been possible, using the fixed node ordering permutation of the cophylogeny reconstruction problem.

Appendix

Proof for Theorem 1

PROOF. By induction we prove that $L_i = \frac{2^{i-1}n}{3^i}$ for all trees produced by the Yule model

True for $i = 1$

$$\begin{aligned} L_i &= \frac{2^{i-1}n}{3^i} \\ L_1 &= \frac{n}{3} \\ L_1 &= C(L_0) \end{aligned} \tag{14}$$

Assume true for $i = k$

$$L_k = \frac{2^{k-1}n}{3^k} \tag{15}$$

Prove true for $i = k + 1$

$$\begin{aligned} L_{k+1} &= C\left(L_0 - \sum_{j=1}^{k} L_j\right) \\ E(L_{k+1}) &= E\left(C\left(L_0 - \sum_{j=1}^{k} L_j\right)\right) \\ &= \frac{1}{3} \times \left(E(L_0) - \sum_{j=1}^{k} E(L_j)\right) \\ &= \frac{1}{3} \times \left(n - \sum_{j=1}^{k} \frac{2^{j-1}n}{3^j}\right) \\ &= \frac{1}{3} \times \left(n - \frac{n}{2} \times \sum_{j=1}^{k} \frac{2^j}{3^j}\right) \\ &= \frac{n}{6} \times \left(2 - \sum_{j=1}^{k} \left(\frac{2}{3}\right)^j\right) \\ &= \frac{n}{6} \times \left(2 - \left(\frac{1 - \frac{2}{3}^{k+1}}{1 - \frac{2}{3}} - 1\right)\right) \\ &= \frac{n}{6} \times \left(2 - 3 + \frac{2^{k+1}}{3^k} + 1\right) \\ &= \frac{n}{6} \times \left(\frac{2^{k+1}}{3^k}\right) \\ &= \frac{2^k n}{3^{k+1}} \end{aligned} \tag{16}$$

Therefore $L_i = \frac{2^{i-1}n}{3^i}$ for all $i \geq 1$.

Proof for Theorem 2

PROOF. By induction we prove that $L_i = \frac{3^{i-1}n}{4^i}$ for all trees produced by the Uniform model

True for $i = 1$

$$\begin{aligned} L_i &= \frac{3^{i-1}n}{4^i} \\ L_1 &= \frac{n}{4} \\ L_1 &= C(L_0) \end{aligned} \tag{17}$$

Assume true for $i = k$

$$L_k = \frac{3^{k-1}n}{4^k} \tag{18}$$

Prove for $i = k + 1$

$$\begin{aligned} L_{k+1} &= C\left(L_0 - \sum_{j=1}^{k} L_j\right) \\ E(L_{k+1}) &= E\left(C\left(L_0 - \sum_{j=1}^{k} L_j\right)\right) \\ &= \frac{1}{4} \times \left(E(L_0) - \sum_{j=1}^{k} E(L_j)\right) \\ &= \frac{1}{4} \times \left(n - \sum_{j=1}^{k} \frac{3^{j-1}n}{4^j}\right) \\ &= \frac{1}{4} \times \left(n - \frac{n}{3} \times \sum_{j=1}^{k} \frac{3^j}{4^j}\right) \\ &= \frac{n}{12} \times \left(3 - \sum_{j=1}^{k} \left(\frac{3}{4}\right)^j\right) \\ &= \frac{n}{12} \times \left(3 - \left(\frac{1 - \left(\frac{3}{4}\right)^{k+1}}{1 - \frac{3}{4}} - 1\right)\right) \\ &= \frac{n}{12} \times \left(3 - 4 + \frac{3^{k+1}}{4^k} + 1\right) \\ &= \frac{n}{12} \times \left(\frac{3^{k+1}}{4^k}\right) \\ &= \frac{3^k n}{4^{k+1}} \end{aligned} \tag{19}$$

Therefore $L_i = \frac{3^{i-1}n}{4^i}$ for all $i \geq 1$.

Proof for Theorem 3

PROOF. The worst case space complexity can be evaluated by simplifying the two geometric series first defined in Equation (8). We define these two series as $\alpha(n)$ and $\beta(n)$ in the form:

$$\alpha(n) = n \sum_{i=1}^{\lfloor \log_2 (\log_3 (n)+1) \rfloor} \frac{2^{(i-1)} 3^{(2^i-1)}}{3^i} \tag{20}$$

$$\beta(n) = n \sum_{i=\lfloor \log_2 (\log_3 (n)+1) \rfloor + 1}^{n-1} \frac{2^{(i-1)} n}{3^i} \tag{21}$$

These two equations will be evaluated in terms of only n so as to evaluate the worst case space complexity for trees constructed under a Yule process.

Simplifying $\alpha(n)$

Lemma 3. $\frac{2^{(i-1)} 3^{(2^i-1)}}{3^i}$ *is super increasing.*

For a function to be super increasing the following must be true:

$$\frac{a_{n+1}}{a_n} > 2 \tag{22}$$

From Equation (20) we have the following.

$$\begin{aligned} \frac{a_{i+1}}{a_i} &= \frac{2^i 3^{(2^{i+1}-1)}}{3^{i+1}} \times \frac{3^i}{2^{(i-1)} 3^{(2^i-1)}} \\ &= \frac{2 \times 3^{(2^{i+1}-1-(2^i-1))}}{3} \\ &= \frac{2 \times 3^{(2^{i+1}-2^i)}}{3} \\ &= \frac{2 \times 3^{(2^i)}}{3} \text{ which is greater than 2 if } i \geq 0 \end{aligned} \tag{23}$$

Therefore, $\frac{2^{(i-1)} 3^{(2^i-1)}}{3^i}$ is super increasing for all $i \geq 1$. As a result the geometric series α can be approximated by only computing the final term of the series and multiplying this result by 2. As the series is in terms of a super increasing function this result is guaranteed to be greater than the actual series allowing for us to derive a worst case bound for this series.

$$
\begin{aligned}
\alpha(n) &= n \sum_{i=1}^{\lfloor \log_2(\log_3(n)+1)\rfloor} \frac{2^{(i-1)}3^{(2^i-1)}}{3^i} \\
&= \frac{n}{2} \sum_{i=1}^{\lfloor \log_2(\log_3(n)+1)\rfloor} \left(\frac{2}{3}\right)^i \times 3^{(2^i-1)} \\
&\leq 2 \times \frac{n}{2} \times \left(\frac{2}{3}\right)^{\lfloor \log_2(\log_3(n)+1)\rfloor} \times 3^{2^{\lfloor \log_2(\log_3(n)+1)\rfloor}-1} \\
&\leq n \times \left(\frac{2}{3}\right)^{\lfloor \log_2(\log_3(n)+1)\rfloor} \times 3^{(\log_3 n+1-1)} \\
&= n^2 \times \left(\frac{2^{\lfloor \log_2(\log_3(n)+1)\rfloor}}{2^{(\log_2 3\lfloor \log_2(\log_3(n)+1)\rfloor}}\right) \\
&= n^2 \times 2^{\lfloor \log_2(\log_3(n)+1)\rfloor(1-\log_2 3)} \\
&\leq n^2 \times (\log_3(n)+1)^{1-\log_2 3} \\
&= \frac{n^2}{(\log_3(n)+1)^{\log_2 3-1}}
\end{aligned} \tag{24}
$$

Simplifying $\beta(n)$

$$
\begin{aligned}
\beta(n) &= n \sum_{i=\lfloor \log_2(\log_3(n)+1)\rfloor+1}^{n-1} \frac{2^{(i-1)}n}{3^i} \\
&= \frac{n^2}{2} \sum_{i=\lfloor \log_2(\log_3(n)+1)\rfloor+1}^{n-1} \left(\frac{2}{3}\right)^i \\
&= \frac{3n^2}{2} \times \left(\left(\frac{2}{3}\right)^{\lfloor \log_2(\log_3(n)+1)\rfloor+1} - \left(\frac{2}{3}\right)^n\right) \text{ as } \sum_{i=a}^{b} r^i = \frac{r^a - r^{b+1}}{1-r} \\
&= \frac{3n^2}{2} \times \left(\frac{2}{3} \times \left(\frac{2}{3}\right)^{\lfloor \log_2(\log_3(n)+1)\rfloor} - \frac{2}{3} \times \left(\frac{2}{3}\right)^{n-1}\right) \\
&= n^2 \times \left(\left(\frac{2}{2^{\log 3}}\right)^{\lfloor \log_2(\log_3(n)+1)\rfloor} - \left(\frac{2}{3}\right)^{n-1}\right) \\
&\leq n^2 \times \left(\frac{1}{(\log_3 n+1)^{\log_2 3-1}} - \left(\frac{2}{3}\right)^{n-1}\right) \\
&= \frac{n^2}{(\log_3 n+1)^{\log_2 3-1}} - n^2\left(\frac{2}{3}\right)^{n-1} \\
&\approx \frac{n^2}{(\log_3 n+1)^{\log_2 3-1}} \text{ as } \lim_{n\to\infty} n^2\left(\frac{2}{3}\right)^{n-1} = 0
\end{aligned} \tag{25}
$$

Combining the result

$$O\left(n + n \sum_{i=1}^{\lfloor \log_2 (\log_3 (n)+1)\rfloor} \frac{2^{(i-1)}3^{(2^i-1)}}{3^i} + n \sum_{i=\lfloor \log_2 (\log_3 (n)+1)\rfloor+1}^{n-1} \frac{2^{(i-1)}n}{3^i}\right)$$
$$= O\left(n + \frac{n^2}{(\log_3 (n)+1))^{\log_2 3-1}} + \frac{n^2}{(\log_3 n+1)^{\log_2 3-1}}\right) \tag{26}$$

As $\log_3 n$ is equal to $\frac{\log_2 n}{\log_2 3}$ by removing all constant factors we can derive the following worst case complexity of:

$$O\left(\frac{n^2}{(\log n)^{\log 3-1}}\right) \tag{27}$$

Proving Theorem 3.

Proof for Theorem 4

Proof. The worst case space complexity can be evaluated by simplifying the two geometric series first defined in Equation (9). We define these two series as $\gamma(n)$ and $\delta(n)$ in the form:

$$\gamma(n) = n \sum_{i=1}^{\lfloor \log_2 (\log_3 (n)+1)\rfloor} \frac{3^{(i-1)}3^{(2^i-1)}}{4^i} \tag{28}$$

$$\delta(n) = n \sum_{i=\lfloor \log_2 (\log_3 (n)+1)\rfloor+1}^{n-1} \frac{3^{(i-1)}n}{4^i} \tag{29}$$

These two equations will be evaluated in terms of only n so as to evaluate the worst case space complexity for trees constructed under the Uniform model.

Simplifying $\gamma(n)$

Lemma 4. $\frac{3^{(i-1)}3^{(2^i-1)}}{4^i}$ *is super increasing.*

For a function to be super increasing the following must be true:

$$\frac{a_{n+1}}{a_n} > 2 \tag{30}$$

From Equation (28) we have the following.

$$\begin{aligned}\frac{a_{i+1}}{a_i} &= \frac{3^i 3^{(2^{i+1}-1)}}{4^{i+1}} \times \frac{4^i}{3^{(i-1)}3^{(2^i-1)}} \\ &= \frac{3 \times 3^{(2^{i+1}-1-(2^i-1))}}{4} \\ &= \frac{3 \times 3^{(2^{i+1}-2^i)}}{4} \\ &= \frac{3 \times 3^{(2^i)}}{4} \text{ which is greater than 2 if } i \geq 0\end{aligned} \tag{31}$$

As $\frac{3^{(i-1)}3^{(2^i-1)}}{4^i}$ is super increasing for all $i \geq 1$. As a result the geometric series can be approximated by only computing the final term of the series and multiplying this result by 2. As the series is in terms of

a super increasing function this result is guaranteed to be greater than the actual series allowing for us to derive a worst case bound for this series.

$$\begin{aligned}
\gamma(n) &= n \sum_{i=1}^{\lfloor \log_2(\log_3(n)+1)\rfloor} \frac{3^{(i-1)}3^{(2^i-1)}}{4^i} \\
&= \frac{n}{3} \sum_{i=1}^{\lfloor \log_2(\log_3(n)+1)\rfloor} \left(\frac{3}{4}\right)^i \times 3^{(2^i-1)} \\
&\leq 2 \times \frac{n}{3} \times \left(\frac{3}{4}\right)^{\lfloor \log_2(\log_3(n)+1)\rfloor} \times 3^{2^{\lfloor \log_2(\log_3(n)+1)\rfloor}-1} \\
&\leq \frac{2n}{3} \times \left(\frac{3}{4}\right)^{\lfloor \log_2(\log_3(n)+1)\rfloor} \times 3^{(\log_3 n+1-1)} \\
&= \frac{2n^2}{3} \times \left(\frac{2^{(\log_2 3\lfloor \log_2(\log_3(n)+1)\rfloor}}{2^{(2\lfloor \log_2(\log_3(n)+1)\rfloor}}\right) \\
&= \frac{2n^2}{3} \times 2^{\lfloor \log_2(\log_3(n)+1)\rfloor(\log_2 3-2)} \\
&\leq \frac{2n^2}{3} \times (\log_3(n)+1)^{\log_2 3-2} \\
&= \frac{2n^2}{3(\log_3(n)+1)^{2-\log_2 3}}
\end{aligned} \tag{32}$$

Simplifying $\delta(n)$

$$\begin{aligned}
\delta(n) &= n \sum_{i=\lfloor \log_2(\log_3(n)+1)\rfloor+1}^{n-1} \frac{3^{(i-1)}n}{4^i} \\
&= \frac{n^2}{3} \sum_{i=\lfloor \log_2(\log_3(n)+1)\rfloor+1}^{n-1} \left(\frac{3}{4}\right)^i \\
&= \frac{4n^2}{3} \times \left(\left(\frac{3}{4}\right)^{\lfloor \log_2(\log_3(n)+1)\rfloor+1} - \left(\frac{3}{4}\right)^n\right) \text{ as } \sum_{i=a}^{b} r^i = \frac{r^a - r^{b+1}}{1-r} \\
&= \frac{4n^2}{3} \times \left(\frac{3}{4} \times \left(\frac{3}{4}\right)^{\lfloor \log_2(\log_3(n)+1)\rfloor} - \frac{3}{4} \times \left(\frac{3}{4}\right)^{n-1}\right) \\
&= n^2 \times \left(\left(\frac{2^{\log_2 3}}{2^2}\right)^{\lfloor \log_2(\log_3(n)+1)\rfloor} - \left(\frac{3}{4}\right)^{n-1}\right) \\
&\leq n^2 \times \left(\frac{1}{(\log_3 n+1)^{2-\log_2 3}} - \left(\frac{3}{4}\right)^{n-1}\right) \\
&= \frac{n^2}{(\log_3 n+1)^{\log_2 3-1}} - n^2\left(\frac{3}{4}\right)^{n-1} \\
&\approx \frac{n^2}{(\log_3 n+1)^{\log_2 3-1}} \text{ as } \lim_{n\to\infty} n^2\left(\frac{3}{4}\right)^{n-1} = 0
\end{aligned} \tag{33}$$

Combining the result

$$O\left(n + n \sum_{i=1}^{\lfloor \log_2 (\log_3 (n)+1) \rfloor} \frac{3^{(i-1)} 3^{(2^i-1)}}{4^i} + n \sum_{i=\lfloor \log_2 (\log_3 (n)+1) \rfloor + 1}^{n-1} \frac{3^{(i-1)} n}{4^i}\right)$$
$$= O\left(n + \frac{2n^2}{3(\log_3 (n)+1))^{2-\log_2 3}} + \frac{n^2}{(\log_3 n + 1)^{2-\log_2 3}}\right) \quad (34)$$

As $\log_3 n$ is equal to $\frac{\log_2 n}{\log_2 3}$ by removing all constant factors we can derive the following worst case complexity of:

$$O\left(\frac{n^2}{(\log n)^{2-\log 3}}\right) \quad (35)$$

Proving Theorem 4.

Competing interests

The authors declare that they have no competing interests.

Author's contributions

BD is responsible for the creation and implementation of the proposed data structure along with contributing to this manuscript. MAC is responsible for the original problem definition along with contributing to this manuscript.

Acknowledgements

We thank Anastasios Viglas for the valuable guidance in how to approach the analysis undertaken within this work which has greatly benefited this final manuscript. This work was supported by an Australian Postgraduate Award to BD and an Australian Research Council Grant (grant number DP1094891) to MAC.

References

Aldous D. The continuum random tree ii: an overview. Stochastic analysis 1991;167:23–70.

Aldous D. The continuum random tree iii. The Annals of Probability 1993;:248–89.

Aldous D. Probability distributions on cladograms. In: Random discrete structures. New York City: Springer; 1996. p. 1–18.

Aldous DJ. Stochastic models and descriptive statistics for phylogenetic trees, from yule to today. Statistical Science 2001;:23–34.

Anderson R, May R. Coevolution of hosts and parasites. Parasitology 1982;85(02):411–26.

Badets M, Whittington I, Lalubin F, Allienne J, Maspimby J, Bentz S, Du Preez LH, Barton D, Hasegawa H, Tandon V, et al. Correlating early evolution of parasitic platyhelminths to gondwana breakup. Systematic biology 2011;60(6):762–81.

Brodie ED, Ridenhour B, Brodie E. The evolutionary response of predators to dangerous prey: hotspots and coldspots in the geographic mosaic of coevolution between garter snakes and newts. Evolution 2002;56(10):2067–82.

Cavalli-Sforza LL, Edwards AW. Phylogenetic analysis. models and estimation procedures. American journal of human genetics 1967;19(3 Pt 1):233.

Charleston M, Perkins S. Lizards, Malaria, and Jungles in the Caribbean. Tangled Trees: Phylogeny, Cospeciation and Coevolution, University of Chicago Press, Chicago 2003;:65–92.

Charleston M, Robertson D. Preferential host switching by primate lentiviruses can account for phylogenetic similarity with the primate phylogeny. Systematic biology 2002;51(3):528–35.

Charleston MA. Recent results in cophylogeny mapping. Advances in parasitology 2003;54:303–30.

Charleston MA, Perkins SL. Traversing the tangle: algorithms and applications for cophylogenetic studies. Journal of biomedical informatics 2006;39(1):62–71.

Conow C, Fielder D, Ovadia Y, Libeskind-Hadas R. Jane: a new tool for the Cophylogeny Reconstruction Problem. Algorithms for Molecular Biology 2010;5(1):16.
Cruaud A, Rønsted N, Chantarasuwan B, Chou LS, Clement WL, Couloux A, Cousins B, Genson G, Harrison RD, Hanson PE, et al. An extreme case of plant–insect codiversification: figs and fig-pollinating wasps. Systematic biology 2012;61(6):1029–47.
Cuthill JH, Charleston M. Phylogenetic Codivergence Supports Coevolution of Mimetic Heliconius Butterflies. PloS One 2012;7(5):e36464.
Doyon JP, Ranwez V, Daubin V, Berry V. Models, Algorithms and Programs for Phylogeny Reconciliation. Briefings in Bioinformatics 2011a;12(5):392–400.
Doyon JP, Scornavacca C, Gorbunov KY, Szöllősi GJ, Ranwez V, Berry V. An Efficient Algorithm for Gene / Species Trees Parsimonious Reconciliation with Losses, Duplications and Transfers. In: Comparative Genomics. New York City: Springer; 2011b. p. 93–108.
Drinkwater B, Charleston MA. An Improved Node Mapping Algorithm for the Cophylogeny Reconstruction Problem. Coevolution 2014a;2(1):1–17.
Drinkwater B, Charleston MA. Introducing TreeCollapse: A novel greedy algorithm to solve the Cophylogeny Reconstruction Problem (Accepted 21st July, 2014). BMC Bioinformatics 2014b;.
Fox L. Defense and Dynamics in Plant-Herbivore Systems. American Zoologist 1981;21(4):853–64.
Gregory TR. Understanding evolutionary trees. Evolution: Education and Outreach 2008;1(2):121–37.
Hafner MS, Nadler SA. Phylogenetic trees support the coevolution of parasites and their hosts. Nature 1988;.
Harding E. The probabilities of rooted tree-shapes generated by random bifurcation. Advances in Applied Probability 1971;:44–77.
Jackson AP. Cophylogeny of the ficus microcosm. Biological Reviews 2004;79(4):751–68.
Jackson AP. The effect of paralogous lineages on the application of reconciliation analysis by cophylogeny mapping. Systematic biology 2005;54(1):127–45.
Juan D, Pazos F, Valencia A. High-confidence prediction of global interactomes based on genome-wide coevolutionary networks. Proceedings of the National Academy of Sciences 2008;105(3):934–9.
Keller-Schmidt S, Wieseke N, Klemm K, Middendorf M. Evaluation of host parasite reconciliation methods using a new approach for cophylogeny generation. Technical Report; Working paper from Bioinformatics Leipzig. Available from http://www. bioinf. uni-leipzig. de/working/11-013; 2011.
Libeskind-Hadas R, Charleston M. On the Computational Complexity of the Reticulate Cophylogeny Reconstruction Problem. Journal of Computational Biology 2009;16(1):105–17.
Liu L, Li XY, Huang XL, Qiao GX. Evolutionary relationships of pemphigus and allied genera (hemiptera: Aphididae: Eriosomatinae) and their primary endosymbiont, buchnera aphidicola. Insect science 2014;21(3):301–12.
McKenzie A, Steel M. Distributions of cherries for two models of trees. Mathematical biosciences 2000;164(1):81–92.
McLeish M, Crespi B, Chapman T, Schwarz M. Parallel diversification of Australian gall-thrips on *Acacia*. Molecular phylogenetics and evolution 2007;43(3):714–25.
Merkle D, Middendorf M. Reconstruction of the cophylogenetic history of related phylogenetic trees with divergence timing information. Theory in Biosciences 2005;123(4):277–99.
Merkle D, Middendorf M, Wieseke N. A parameter-adaptive dynamic programming approach for inferring cophylogenies. BMC Bioinformatics 2010;11(Suppl 1):S60.
Mooers AO, Heard SB. Inferring evolutionary process from phylogenetic tree shape. Quarterly Review of Biology 1997;:31–54.
Mu J, Joy DA, Duan J, Huang Y, Carlton J, Walker J, Barnwell J, Beerli P, Charleston M, Pybus O, et al. Host switch leads to emergence of plasmodium vivax malaria in humans. Molecular biology and evolution 2005;22(8):1686–93.
Ovadia Y, Fielder D, Conow C, Libeskind-Hadas R. The Cophylogeny Reconstruction Problem is NP-Complete. Journal of Computational Biology 2011;18(1):59–65.
Page RD, Cruickshank RH, Dickens M, Furness RW, Kennedy M, Palma RL, Smith VS. Phylogeny of *Philoceanus complex* seabird lice (Phthiraptera: Ischnocera) inferred from mitochondrial dna sequences. Molecular phylogenetics and evolution 2004;30(3):633–52.
Page RDM. Tangled trees: Phylogeny, cospeciation, and coevolution. Chicago: University of Chicago Press, 2002.
Page RDM, Charleston MA. From gene to organismal phylogeny: reconciled trees and the gene tree/species tree problem. Molecular phylogenetics and evolution 1997;7(2):231–40.
Paterson AM, Palma RL, Gray RD. Drowning on arrival, missing the boat, and x-events: How likely are sorting events. Tangled Trees: Phylogeny, Cospeciation, and Coevolution 2003;:287–309.
Poulin R. Evolutionary Ecology of Parasites. Princeton: Princeton University Press, 2011.
Refrégier G, Le Gac M, Jabbour F, Widmer A, Shykoff JA, Yockteng R, Hood ME, Giraud T. Cophylogeny of the anther smut fungi and their caryophyllaceous hosts: prevalence of host shifts and importance of delimiting parasite species for inferring cospeciation. BMC Evolutionary Biology 2008;8(1):100.
Ronquist F. Reconstructing the history of host-parasite associations using generalised parsimony. Cladistics 1995;11(1):73–89.
Ronquist F. Phylogenetic approaches in coevolution and biogeography. Zoologica scripta 1997;26(4):313–22.
Rosen DE. Vicariant patterns and historical explanation in biogeography. Systematic Biology 1978;27(2):159–88.
Rosenblueth M, Sayavedra L, Sámano-Sánchez H, Roth A, Martínez-Romero E. Evolutionary relationships of flavobacterial and enterobacterial endosymbionts with their scale insect hosts (hemiptera: Coccoidea). Journal of evolutionary biology 2012;25(11):2357–68.
Yodpinyanee A, Cousins B, Peebles J, Schramm T, Libeskind-Hadas R. Faster Dynamic Programming Algorithms for the Cophylogeny Reconstruction Problem. HMC CS Technical Report 2011;.
Yule GU. A mathematical theory of evolution, based on the conclusions of dr. jc willis, frs. Philosophical Transactions of the

Royal Society of London Series B, Containing Papers of a Biological Character 1924;:21–87.