RASCAL: A randomised approach for coevolutionary analysis

February 4, 2016

Benjamin Drinkwater*
Affiliation:
School of Information Technologies, The University of Sydney, NSW, 2006, Australia
email: benjamin.drinkwater@sydney.edu.au
telephone: +61 (0)2 9351 3423

Michael A. Charleston
Affiliation:
School of Physical Sciences, University Of Tasmania, TAS, 7005, Australia
email: michael.charleston@utas.edu.au
telephone: +61 (0)3 6226 2444

Abstract

A popular method for coevolutionary inference is cophylogenetic reconstruction where the branch length of the phylogenies have been previously derived. This approach, unlike the more generalized reconstruction techniques which are NP-Hard, can reconcile the shared evolutionary history of a pair of phylogenetic trees in polynomial time. This approach while proven to be highly successful requires a high polynomial running time. This is quickly becoming a limiting factor of this approach due to the continual increase in size of coevolutionary data sets. One existing method which combats this issue proposes a trade-off of accuracy for an asymptotic time complexity reduction. This technique in almost 70% of cases converges on Pareto optimal solutions in linear time. We build on this prior work by proposing an alternate linear time algorithm (RASCAL) that offers a significant accuracy increase, with RASCAL converging on Pareto optimal solutions in 85% of cases and unlike prior methods can ensure, with high probability, that all optimal solutions can be recovered, provided sufficient replicates are performed.

Key words: Coevolution, Phylogeny, Randomisation, NP-Hard.

Availability: The source files and synthetic coevolutionary systems applied in the analysis presented herein are available for download at:

`http://sydney.edu.au/engineering/it/~bdri5538/software/RASCAL.zip`

1 Background

Coevolutionary interrelationships are a strong driver of evolutionary variation [Ricklefs, 2010]. These relationships may be of mutual benefit where each species provides an evolutionary advantage for all parties involved, such as the reproductive dependence between fig trees and their pollinator wasps [Anstett et al., 1997], or alternatively, may be parasitic, where only a subset of the participants benefit, such as parasitic pinworms' exploitation of primates [Ronquist, 1997].

Often coevolutionary interactions are studied at a macro scale and consider a pair of phylogenetic trees, the host (H) and parasite (P), and the degree of congruence that exists between these phylogenies based on the known associations (φ) between their extant taxa [Charleston and Libeskind-Hadas, 2014]. One common technique for identifying the congruence between H and P is cophylogeny mapping [Charleston, 2002]. This approach, similarly to the duplication, transfer and loss (DTL) model used to infer species trees from gene trees [Page and Charleston, 1997], aims to map P (a gene tree) into H (a species tree) where the associated cost is minimised [Bansal et al., 2013].

Mapping P into H requires four recoverable events: codivergence, duplication, host switch and loss [Page and Charleston, 1998]. These four events are capable of reconciling all possible coevolutionary permutations as seen in Figure 1, where parasites are restricted to only inhabiting a single host [Ronquist, 1995]; the permutation of the problem considered herein.

Reconciling a parasite's evolutionary history with respect to its host requires an input set which consists of a pair of bifurcating trees, the host, H, and parasite, P, the associations between their extant taxa, φ, and a set of costs for each coevolutionary event, known as an event cost vector, $V = (C, D, S, L)$. The output consists of a map, Φ, which includes the frequency of each evolutionary event induced by the map, often described as an event count

vector, $N = (\alpha, \beta, \gamma, \delta)$ [Libeskind-Hadas et al., 2014]. To reconcile biological representative maps, cophylogeny mapping algorithms often aim to minimise the cost, E, of the resultant map, Φ, where E is defined as:

$$E = \alpha C + \beta D + \gamma S + \delta L \tag{1}$$

Parsimony or event-cost methods are often less preferable to maximum likelihood techniques, in particular when using arbitrarily chosen cost vectors. An approach that applies a cost vector, however, holds the potential of reconstructing the most likely evolutionary history, if the cost vector is derived from the negative log likelihood probabilities of each of the associated events [Arvestad et al., 2003]. As a result, there is a strong driver for fast event-based methods which can then be integrated into a coevolutionary likelihood framework [Charleston, 2003].

Unfortunately the reconstruction of the minimum cost map where all four events are permitted, *the cophylogeny reconstruction problem*, is NP-Hard [Ovadia et al., 2011, Tofigh et al., 2011]. Due to the computational intractability of this problem a number of heuristics have been proposed. One such heuristic, ignores the relative ordering of the parasite's evolutionary history as defined by the divergence events in P [Merkle and Middendorf, 2005]. Ignoring the relative ordering of the divergence event in P provides an estimation of the cophylogeny reconstruction problem in polynomial time [Merkle et al., 2010]. An alternate approach favours fixing the internal node ordering of the host tree reducing this problem to the dated tree reconciliation (DTR) problem which can be solved in polynomial time [Libeskind-Hadas and Charleston, 2009]. This approach, however, is forced to consider a potentially exponential number of internal node orderings to guarantee optimality [Libeskind-Hadas, 2011].

Both of these approaches face limitations to either the accuracy or the expected running

time. When ignoring the relative ordering of the internal nodes in P it is possible to reconstruct a map, Φ, where the order of evolutionary events as inferred from Φ contradict the initial ordering as defined by the parasite's phylogenetic history. Such a map is often referred to as time-inconsistent or biologically infeasible [Doyon et al., 2011a,b]. While it has been shown that such solutions are uncommon [Arvestad et al., 2003], in cases where this does occur this technique is often unable to recover a biologically feasible alternative [Doyon et al., 2011b, Drinkwater and Charleston, 2014b].

The desire to ensure that recovered maps are time-consistent has led to a strong focus on improving the computational complexity of algorithms which solve the DTR problem. These approaches often apply dynamic programming to reconstruct an optimal map in polynomial time [Libeskind-Hadas and Charleston, 2009]. Early implementations using this approach ran in $O(n^7)$ [Conow et al., 2010] which was a limiting factor, but subsequent algorithms continued to decrease their asymptotic complexity making this approach more feasible [Doyon et al., 2011b, Yodpinyanee et al., 2011, Bansal et al., 2012].

While the running time of algorithms for the DTR problem have continued to decrease, this approach is still limited due to there being, in the worst case, an exponential number of internal node orderings possible for a bifurcating tree [Libeskind-Hadas and Charleston, 2009]. Most methods mitigate this to a degree by applying a metaheuristic to traverse only a subset of the exponential search space terminating in a fixed period of time [Conow et al., 2010]. While this approach ensures that recovered maps are time-consistent, it cannot guarantee optimality. This approach, however, has proven popular due to its ability to converge on reasonable maps in a short period of time for the majority of coevolutionary data sets.

While this approach has successfully been used to analyse instances with up to 200 taxa [Cruaud et al., 2012], it is infeasible for the analysis of cases of Wolbachia and their

insect hosts which have upwards of 1.2 million taxa [Novotny et al., 2002, Hilgenboecker et al., 2008]. To avoid relying solely on algorithms that may provide biologically infeasible solutions, a significant reduction to the associated running time for the DTR problem must be achieved.

This has led to a new field of algorithmic development which aims to further reduce the time complexity of DTR problem from the current bound of $O(n^2 \log n)$ [Bansal et al., 2012]. One previously proposed method, TreeCollapse, applies a linear time greedy algorithm which trades off accuracy for speed, while still guaranteeing that the recovered maps are time-consistent [Drinkwater and Charleston, 2014b]. This algorithm has been shown to report optimal maps in 69% of cases while achieving more than a $O(n)$ speed up compared to existing algorithms.

We continue this inquiry, but rather than solving the DTR problem greedily, we propose a modification to the existing Improved Node Mapping algorithm [Drinkwater and Charleston, 2014a] by applying random sampling of mapping sites for a given node in P when constructing the dynamic programming matrix. This updated algorithm, which we call RASCAL (RAndomised Sampling for Cophylogenetic AnaLysis), only maintains a subset of the dynamic programming matrix. We show that with enough repeated runs within a genetic algorithm, even a small random selection of mapping sites can result in RASCAL inferring highly accurate maps.

2 Methodology and Implementation

The algorithm we describe herein is a modification of the existing Improved Node Mapping algorithm which solves the DTR problem optimally in $O(n^3)$ [Drinkwater and Charleston, 2014a]. We prove that by reducing the number of mapping sites (sub solutions), $\Phi(p_i)$, stored for each parasite node $p_i \in P$ to a value k which is strictly less than $O(n)$ results in

a asymptotic time and space decrease.

To reconstruct the optimal mapping sites for p_i, $\Phi(p_i)$, requires that $O(n^2)$ mapping sites be reconstructed for each node p_i where n in this context is the number of mapping sites stored for each of p_i's children. Once the $O(n^2)$ mapping sites are recovered only a subset needs to be retained [Drinkwater and Charleston, 2014a]. This subset of $O(n)$ mapping sites contains the minimum cost sub solutions associated with each node in the host tree [Libeskind-Hadas and Charleston, 2009].

The recovery of all coevolutionary events stored in $\Phi(p_i)$ can be recovered in constant time (see proof in [Drinkwater and Charleston, 2014a]). As a result, the time complexity of this algorithm is directly bounded by the number of mapping sites stored for each node $p_i \in P$. This result is part of a larger proof which demonstrated that for a subset of tree topologies that asymptotically less than $O(n)$ mapping sites are required to solve the DTR problem optimally [Drinkwater and Charleston, 2015]. It is this characteristic that we aim to further exploit to reduce the time and space complexity bound for the DTR problem using random sampling for all tree topologies.

Conversely, while this technique does offer the potential for a time complexity reduction for Improved Node Mapping, it cannot be applied to Bansal et al.'s [2012] asymptotically faster algorithm. This is due to the $O(n \log n)$ preprocessing step required for each parasite node which is independent of the number of mapping sites stored. Therefore, while Bansal et al.'s [2012] solution is more efficient when solving the problem optimally, its design prohibits any asymptotic time complexity decrease by reducing the number of sub solutions that are retained at each iteration [Drinkwater and Charleston, 2015].

2.1 Using random sampling

The Node Mapping algorithm's computational complexity is directly proportional to the number of mapping sites, $\Phi(p_i)$, stored for each parasite node $p_i \in P$ [Doyon et al., 2011a, Libeskind-Hadas and Charleston, 2009]. Our solution proposes that only a subset of these mapping sites $\overline{\Phi(p_i)} \subseteq \Phi(p_i)$ be retained at each iteration where $|\overline{\Phi(p_i)}| \leq k$ and k is asymptotically less than $O(n)$.

Although there are many techniques that may be applied to the selection of the subset, $\overline{\Phi(p_i)}$, our proposed method uses random sampling. Random sampling, unlike greedy approaches, has the advantage that while offering no guarantee that recovered solutions are optimal it can in principle, ensure, with high probability, that all optimal solutions can be recovered, provided sufficient repetitions are performed. This complements its use within a metaheuristic framework involving repeated executions of this underlying algorithm in an attempt to infer the global optima.

2.2 Storing k random samples

The selection of a random subset of mapping sites, $\overline{\Phi(p_i)}$, requires an update to the Improved Node Mapping algorithm [Drinkwater and Charleston, 2014a], in particular, providing an adaptive data structure which allows for a random subset of size k to be retained for each node p_i, along with a method to procure the random subset at each iteration. This functionality has been integrated into the RASCAL algorithm as seen in Figure 2.

Node Mapping algorithms have traditionally stored the minimum cost mapping sites in a two-dimensional matrix of size $O(n^2)$ [Yodpinyanee et al., 2011, Bansal et al., 2012]. While still possible to use a two-dimensional matrix, this time of size $O(kn)$, we have instead stored the sub solutions within an array of lists. This is due to a recent result showing a logarithmic reduction in running time is possible when using an array of lists for a select

subset of tree topologies [Drinkwater and Charleston, 2015]. While this approach does not provide an asymptotic reduction for all tree topologies, it is expected to perform faster in practice compared to using the traditional two dimensional matrix.

The newly proposed update to the Improved Node Mapping algorithm ensures that only k items are retained for each node $p_i \in P$ and that this subset of size k is procured in linear time and therefore RASCAL's time and space complexity is defined as follows:

Definition 1 *RASCAL reconciles a time-consistent map for the DTR problem in* $O(k^2n)$ $\forall$ $k \geq 1$*, requiring* $O(kn)$ *space.*

3 Results and Analysis

We examined the accuracy and running time of Bansal et al.'s DTR algorithm, Improved Node Mapping, TreeCollapse and RASCAL over a previously published catalogue of biological data sets along with two synthetically generated data sets produced using CoRe-Gen [Keller-Schmidt et al., 2011]. All existing algorithms incorporated in this analysis were implemented in Java to allow for a consistent evaluation. Java was chosen as the majority of coevolutionary analysis tools have been implemented in Java due to its excellent support for cross platform computation [Conow et al., 2010]. For consistency, the one algorithm which does not have a Java implementation, Bansal et al.'s [2012] DTR algorithm incorporated in RANGER-DTL was reimplemented in Java to provide an unbiased running time comparison.

To evaluate RASCAL's accuracy it was compared to two algorithms that are able to solve the DTR problem optimally, along with the greedy algorithm TreeCollapse. This comparison evaluated each algorithm's performance over two previously published data sets. The first includes 953 previously generated synthetic coevolutionary systems [Keller-Schmidt et al., 2011], while the second data set contains 102 previously published biological systems covering a wide range of coevolutionary scenarios such as, parasitism [Page et al., 2004],

plant-insect networks [McLeish et al., 2007], mutualistic coevolution [Jackson, 2004], both Müllerian and Batesian mimicry [Cuthill and Charleston, 2012, Ceccarelli and Crozier, 2007], and biogeography [Badets et al., 2011]. Both sets have been used previously to compare the accuracy of coevolutionary analysis tools [Drinkwater and Charleston, 2014b, 2015].

To compare the running time performance of both RASCAL and the previously published algorithms included in this analysis required a new set of synthetically generated data specifically procured for this study. This set, unlike prior data sets, contains significantly larger tanglegrams including synthetic coevolutionary data sets with 5000 taxa, compared with existing data sets only containing tanglegrams with up to 400 taxa. A larger data set was essential for this analysis to provide a robust evaluation of how RASCAL's running time compares with these existing algorithms when reconciling the coevolutionary associations of large data sets. This data set, however, could not be used for an accuracy comparison as such an analysis would have required 28 years of computation for the slower algorithms included in this study to converge, when executed within a metaheuristic framework.

The analysis of RASCAL's performance in terms of accuracy and running time in practice is broken into three sections. The first compares RASCAL's accuracy using three random sampling parameters; $\lceil\sqrt{n}\rceil$, $\lceil\log n\rceil$ and 4, against algorithms that solve the DTR problem optimally. RASCAL's linear time implementation is then compared against TreeCollapse, the most accurate linear time algorithm for solving the DTR problem. Finally, we compare the running time of RASCAL against these previously published algorithms providing a comprehensive analysis of how the worst case theoretical complexity of each algorithm compares to their practical running time performance.

3.1 RASCAL's accuracy

RASCAL is designed to run within a metaheuristic framework similar to other methods to estimate the cophylogeny reconstruction problem [Libeskind-Hadas and Charleston, 2009, Conow et al., 2010, Yodpinyanee et al., 2011]. The metaheuristic applied within this evaluation was a genetic algorithm applying a population size of 100 and executing 100 iterations for each problem instance. This configuration is consistent with previous coevolutionary analysis using tools such as Jane [Conow et al., 2010].

Each method was rerun 100 times for each problem instance to provide a robust set of replicates to evaluate RASCAL's accuracy. The distribution of these replicates is recorded in Figure 3 along with noting the frequency that RASCAL converges on the best known solution in Table 2.

These results demonstrate that RASCAL is able to recover robust reconstructions with a high degree of accuracy. Even in the case where k is a constant, RASCAL is able to converge on the optimal solution in 85% of cases, over the biological data set, where even algorithms which optimally solve the DTR problem converge on the optimal in only 98% of cases when solving the cophylogeny reconstruction problem. This performance is even more impressive when considering RASCAL's performance over the synthetic data set, where it was able to infer the optimal reconstruction in 96% of cases compared with the best known algorithms which are still unable to recover all the optimal solutions, even with an additional order of magnitude increase in their asymptotic complexity.

3.2 Comparing RASCAL and TreeCollapse

TreeCollapse is the only linear time algorithm prior to this work capable of reconciling time consistent solutions to the cophylogeny reconstruction problem. This approach is known to not always recover the optimal solutions, however, it can provide robust estimations in a

short period of time. RASCAL, where k is set to a constant value such as 4, is the second such algorithm that can recover time consistent solutions. In this section we compare which of these methodologies provides the most accurate underlying linear time approach for solving the cophylogeny reconstruction problem within a metaheuristic framework, using the same data sets leveraged in the previous analysis.

It is clear from Table 3 and Figure 4 that RASCAL out performs TreeCollapse over both the synthetic (a) and biological (b) data sets. RASCAL converges on almost 40% more optimal solutions over the synthetic data set and almost 18% more optimal solutions over the biological data set compared to TreeCollapse. These results argue strongly that RASCAL is the most accurate linear time algorithm for solving the cophylogeny reconstruction problem. In the following section we will show that RASCAL is not only more accurate but that it is also significantly faster in practice compared to TreeCollapse.

3.3 RASCAL's running time in practice

The data set used to evaluate the running time improvements provided by RASCAL was generated using CoRe-Gen [Keller-Schmidt et al., 2011] and included tanglegrams with up to 5000 leaves. Using significantly larger data sets compared to previous analyse can reveal how the asymptotic complexity improvements offered by RASCAL translate into actual improvements in running time, particularly as larger coevolutionary data sets require analysis.

Figure 5 demonstrates the significant reduction in running time possible when using RASCAL. The results highlight that not only is RASCAL asymptotically faster but in practice provides significant running time reduction. Consider the case where there are 5000 taxa. In this case RASCAL, configured where $k = 4$, is able to reconcile a solution 155 times faster than Bansal et al.'s DTR algorithm. In practice when integrated within a metaheuristic framework this translates to reconstructing a solution that has a 91% chance of being opti-

mal in 5 minutes compared to the having a 99% chance of being optimal in 12 hours. Further, this running time analysis demonstrates that RASCAL not only outperforms TreeCollapse in terms of accuracy but is also approximately 3 times faster, in practice.

4 Conclusion

In this work we have introduced a new approach to solve the dated tree reconciliation problem (DTR). Our new approach, RASCAL, has been shown to perform particularly well at solving the cophylogeny reconstruction problem within a metaheuristic framework. Our method has been shown to have an accuracy degradation of only 8% while reducing the asymptotic running time by more than a factor of n. Further, RASCAL's accuracy, when configured to only retain constant number of sub solution at each iteration, is the best known linear time algorithm capable of solving the DTR problem recovering almost 20% more optimal solutions compared to prior linear time implementations. What is even more impressive is that while prior linear time algorithms, such as TreeCollapse, are unable to solve certain problem instances due to being trapped in local optima, RASCAL is able to infer the optimal coevolutionary interrelationships for any system provided sufficient replicates are run, proving itself a valuable tool to assist in the inquiry of larger coevolutionary data sets that have previously been impossible to analyse.

Acknowledgements

This work was supported by the Australian Postgraduate Award and the William and Catherine Mcllrath Scholarship awarded to BD.

Author Disclosure Statement

No competing financial interests exist.

References

M. C. Anstett, M. Hossaert-McKey, and F. Kjellberg. Figs and fig pollinators: evolutionary conflicts in a coevoled mutualism. *Trends in Ecology & Evolution*, 12(3):94–99, 1997.

L. Arvestad, A.-C. Berglund, J. Lagergren, and B. Sennblad. Bayesian gene/species tree reconciliation and orthology analysis using MCMC. *Bioinformatics*, 19(suppl 1):i7–i15, 2003.

M. Badets, I. Whittington, F. Lalubin, J.-F. Allienne, J.-L. Maspimby, S. Bentz, L. H. Du Preez, D. Barton, H. Hasegawa, V. Tandon, et al. Correlating Early Evolution of Parasitic Platyhelminths to Gondwana Breakup. *Systematic Biology*, 60(6):762–781, 2011.

M. S. Bansal, E. J. Alm, and M. Kellis. Efficient algorithms for the reconciliation problem with gene duplication, horizontal transfer and loss. *Bioinformatics*, 28(12):i283–i291, 2012.

M. S. Bansal, E. J. Alm, and M. Kellis. Reconciliation revisited: Handling multiple optima when reconciling with duplication, transfer, and loss. *Journal of Computational Biology*, 20(10):738–754, 2013.

F. Ceccarelli and R. Crozier. Dynamics of the evolution of Batesian mimicry: molecular phylogenetic analysis of ant-mimicking *Myrmarachne* (Araneae: Salticidae) species and their ant models. *Journal of Evolutionary Biology*, 20(1):286–295, 2007.

M. Charleston. Recent results in cophylogeny mapping. *Advances in parasitology*, 54:303–330, 2003.

M. Charleston and R. Libeskind-Hadas. Event-Based Cophylogenetic Comparative Analysis. In *Modern Phylogenetic Comparative Methods and Their Application in Evolutionary Biology*, pages 465–480. Springer, 2014.

M. A. Charleston. Principles of cophylogenetic maps. In *Biological Evolution and Statistical Physics*, pages 122–147. Springer, 2002.

C. Conow, D. Fielder, Y. Ovadia, and R. Libeskind-Hadas. Jane: a new tool for the cophylogeny reconstruction problem. *Algorithms for Molecular Biology*, 5(1):16, 2010.

A. Cruaud, N. Rønsted, B. Chantarasuwan, L. S. Chou, W. L. Clement, A. Couloux, B. Cousins, G. Genson, R. D. Harrison, P. E. Hanson, et al. An Extreme Case of Plant–Insect Codiversification: Figs and Fig-pollinating Wasps. *Systematic Biology*, 61(6):1029–1047, 2012.

J. H. Cuthill and M. Charleston. Phylogenetic Codivergence Supports Coevolution of Mimetic *Heliconius* Butterflies. *PloS one*, 7(5):e36464, 2012.

J.-P. Doyon, V. Ranwez, V. Daubin, and V. Berry. Models, Algorithms and Programs for Phylogeny Reconciliation. *Briefings in Bioinformatics*, 12(5):392–400, 2011a.

J.-P. Doyon, C. Scornavacca, K. Y. Gorbunov, G. J. Szöllősi, V. Ranwez, and V. Berry. An Efficient Algorithm for Gene / Species Trees Parsimonious Reconciliation with Losses, Duplications and Transfers. In *Comparative Genomics*, pages 93–108. Springer, 2011b.

B. Drinkwater and M. A. Charleston. An improved node mapping algorithm for the cophylogeny reconstruction problem. *Coevolution*, 2(1):1–17, 2014a.

B. Drinkwater and M. A. Charleston. Introducing TreeCollapse: a novel greedy algorithm to solve the cophylogeny reconstruction problem. *BMC Bioinformatics*, 15(Suppl 16):S14, 2014b.

B. Drinkwater and M. A. Charleston. A time and space complexity reduction for coevolutionary analysis of trees generated under both a Yule and Uniform model. *Computational Biology and Chemistry*, 2015.

R. A. Fisher, F. Yates, et al. Statistical tables for biological, agricultural and medical research. *Statistical tables for biological, agricultural and medical research.*, (Third Edition), 1949.

K. Hilgenboecker, P. Hammerstein, P. Schlattmann, A. Telschow, and J. H. Werren. How many species are infected with *Wolbachia*?–a statistical analysis of current data. *FEMS Microbiology Letters*, 281(2):215–220, 2008.

A. P. Jackson. Cophylogeny of the Ficus microcosm. *Biological Reviews*, 79(4):751–768, 2004.

S. Keller-Schmidt, N. Wieseke, K. Klemm, and M. Middendorf. Evaluation of Host Parasite Reconciliation Methods using a new Approach for Cophylogeny Generation. Technical report, Bioinformatics Leipzig, 2011.

R. Libeskind-Hadas. Figs, Wasps, Gophers, and Lice: a Computational Exploration of Coevolution. *Bioinformatics for biologists*, pages 227–247, 2011.

R. Libeskind-Hadas and M. Charleston. On the computational complexity of the reticulate cophylogeny reconstruction problem. *Journal of Computational Biology*, 16(1):105–117, 2009.

R. Libeskind-Hadas, Y.-C. Wu, M. S. Bansal, and M. Kellis. Pareto-optimal phylogenetic tree reconciliation. *Bioinformatics*, 30(12):i87–i95, 2014.

M. McLeish, B. Crespi, T. Chapman, and M. Schwarz. Parallel diversification of Australian gall-thrips on *Acacia*. *Molecular phylogenetics and evolution*, 43(3):714–725, 2007.

D. Merkle and M. Middendorf. Reconstruction of the cophylogenetic history of related phylogenetic trees with divergence timing information. *Theory in Biosciences*, 123(4): 277–299, 2005.

D. Merkle, M. Middendorf, and N. Wieseke. A parameter-adaptive dynamic programming approach for inferring cophylogenies. *BMC Bioinformatics*, 11(Suppl 1):S60, 2010.

V. Novotny, Y. Basset, S. E. Miller, G. D. Weiblen, B. Bremer, L. Cizek, and P. Drozd. Low host specificity of herbivorous insects in a tropical forest. *Nature*, 416(6883):841–844, 2002.

Y. Ovadia, D. Fielder, C. Conow, and R. Libeskind-Hadas. The Cophylogeny Reconstruction Problem is NP-Complete. *Journal of Computational Biology*, 18(1):59–65, 2011.

R. D. Page and M. A. Charleston. From Gene to Organismal Phylogeny: Reconciled Trees and the Gene Tree/Species Tree Problem. *Molecular Phylogenetics and Evolution*, 7(2): 231–240, 1997.

R. D. Page and M. A. Charleston. Trees within trees: phylogeny and historical associations. *Trends in Ecology & Evolution*, 13(9):356–359, 1998.

R. D. Page, R. H. Cruickshank, M. Dickens, R. W. Furness, M. Kennedy, R. L. Palma, and V. S. Smith. Phylogeny of *Philoceanus complex* seabird lice (*Phthiraptera: Ischnocera*) inferred from mitochondrial dna sequences. *Molecular phylogenetics and evolution*, 30(3): 633–652, 2004.

R. E. Ricklefs. Evolutionary diversification, coevolution between populations and their antagonists, and the filling of niche space. *Proceedings of the National Academy of Sciences*, 107(4):1265–1272, 2010.

F. Ronquist. Reconstructing the history of host-parasite associations using generalised parsimony. *Cladistics*, 11(1):73–89, 1995.

F. Ronquist. Phylogenetic approaches in coevolution and biogeography. *Zoologica scripta*, 26(4):313–322, 1997.

A. Tofigh, M. Hallett, and J. Lagergren. Simultaneous Identification of Duplications and Lateral Gene Transfers. *IEEE/ACM Transactions on Computational Biology and Bioinformatics (TCBB)*, 8(2):517–535, 2011.

A. Yodpinyanee, B. Cousins, J. Peebles, T. Schramm, and R. Libeskind-Hadas. Faster Dynamic Programming Algorithms for the Cophylogeny Reconstruction Problem. *HMC CS Technical Report*, 2011.

Table 1: The worst case asymptotic complexity of the six algorithms considered in this study. As part of our analysis we evaluate how the asymptotic complexity bound of each algorithm considered compares to their running time in practice.

Algorithm	Worst Case Asymptotic Complexity
Bansal et al.'s	$O(n^2 \log n)$
Improved Node Mapping	$O\left(\frac{n^3}{\log n}\right)$
RASCAL ($k = 4$)	$O(n)$
RASCAL ($k = \log n$)	$O(n \log^2 n)$
RASCAL ($k = \sqrt{n}$)	$O(n^2)$
TreeCollapse	$O(n)$

Table 2: The rate at which RASCAL and the Optimal DTR algorithms converge on the best known biologically feasible solution for the cophylogeny reconstruction problem for both the synthetic and biological data sets. As 100 replicates were run for each data set there were 95300 samples for the synthetic data set and 10200 samples for the biological data set. It is important to note that the optimal DTR algorithms will not always converge on the optimal solution over all possible node timings as a metaheuristic only searches a subset of the exponential search space.

	Frequency that algorithm reports the optimal solution	
Sampling Rate	Synthetic Data	Biological Data
Optimal DTR algorithm's	95243 (99.94%)	10018 (98.22%)
$k = \lceil \sqrt{n} \rceil$	94084 (98.72%)	9530 (93.42%)
$k = \lceil \log n \rceil$	93737 (98.35%)	9327 (91.44%)
$k = 4$	91915 (96.44%)	8666 (84.96%)

Table 3: The frequency with which RASCAL and TreeCollapse converge on the best known biologically feasible solution for the cophylogeny reconstruction problem for both the synthetic and biological data sets. As 100 replicates were run for each data set there were 95300 samples for the synthetic data set and 10200 samples for the biological data set.

	Frequency that algorithm reports the optimal solution	
	Synthetic Data	Biological Data
RASCAL ($k = 4$)	91915 (96.44%)	8666 (84.96%)
TreeCollapse	54564 (57.26%)	6978 (68.41%)

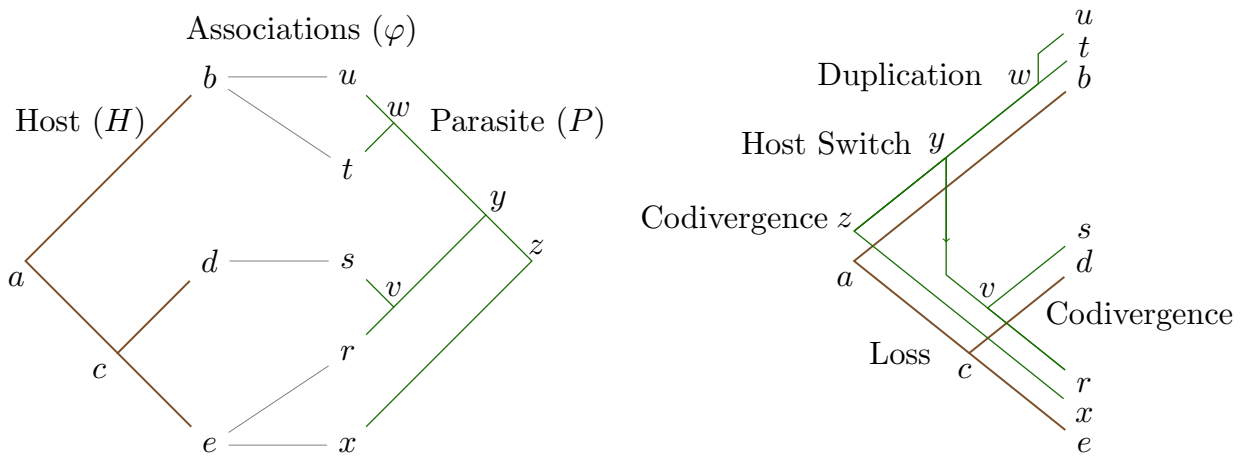

Figure 1: A tanglegram (left) and one of its three possible optimal maps (right). What is unique about this possible map, Φ, is that it is the only optimal map which consists of all four coevolutionary events, codivergence (node v and node z), duplication (node w), host switch (node y) and loss (edge (z, x) at host node c).

Algorithm 1 RASCAL(H, P, φ, V, k)

```
1: Φ is an array of lists which is worst case O(|P| × k)
2: L ← is a list of nodes in P
3: Sort the nodes in L by their distance from the root of P
4: for p_i ∈ L do
5:     if p_i is a leaf then
6:         Φ[p_i] ← leaf h_i ∈ H which p_i is associated with as defined in φ
7:     else
8:         l, r ← the left and right children of p_i
9:         for h_i ∈ Φ[l] do
10:            for h_j ∈ Φ[r] do
11:                Φ[p_i][h_k] ← minimum cost event for p_i at node h_k
12:            end for
13:        end for
14:        Randomly shuffle list Φ[p_i]
15:        while |Φ[p_i]| > k do
16:            remove first element of list Φ[p_i]
17:        end while
18:    end if
19: end for
20: return Φ(P)
```

Figure 2: An updated version of the Improved Node mapping algorithm which takes in a host tree (H), parasite tree (P), the associations between there extant taxa (φ), the event costs for each evolutionary event (V) and the sampling size (k). The updates to the algorithm can be seen on lines 14 to 17. By including lines (14-17) only k mapping sites are retained for each parasite node. This results in the asymptotic running time of lines 9 to 13 being reduced to $O(k^2)$ compared to the running time of $O(n^2)$ from the initial Improved Node Mapping algorithm. It is important to note in this updated algorithm that the data structure, Φ, is indexed first on the parasite node and then secondly on the host node. The array (indexed by parasite nodes) is always size $|P|$ with the list (indexed by host nodes) of mapping locations bound to size k. Finally, the random shuffle step on line 14 is implemented using the Fisher-Yates shuffle algorithm [Fisher et al., 1949] to ensure that $\Phi[p_i]$ can be procured in linear time and that the RASCAL algorithm can infer the map Φ in $O(nk^2)$.

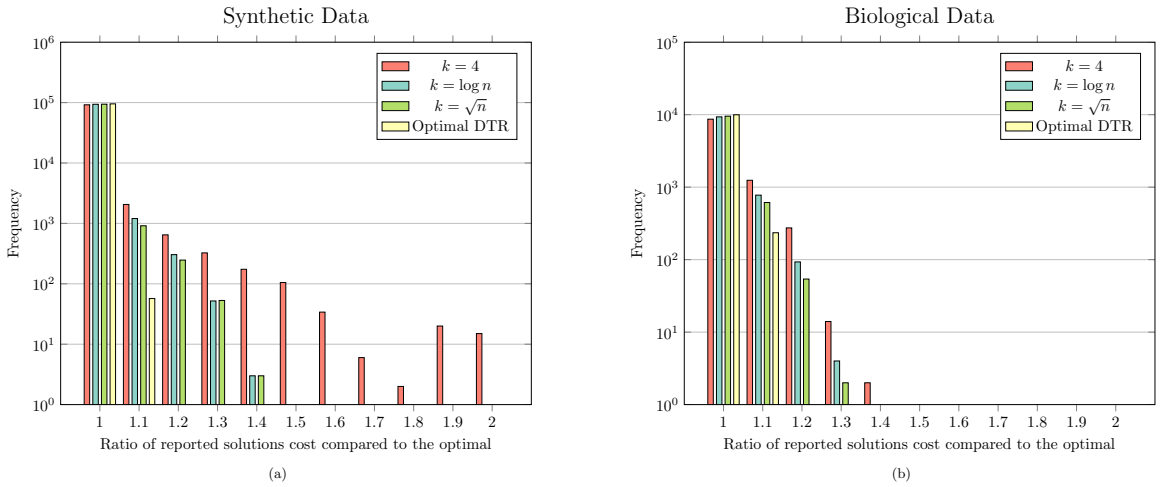

Figure 3: The degradation of accuracy for RASCAL for three different values of k ($\sqrt{n}$, $\log n$ and 4) run over both synthetic (a) and biological (b) data sets. Of note is that while for the synthetic data there is a higher rate at which RASCAL converges on the optimal (best known solution), it also has the largest variation in accuracy, with a number of reported solutions twice the optimal cost. This is compared with the biological data set where, while reconciling fewer optimal solutions, RASCAL was always able to ensure that it never reported a map which cost more than 40% that of the optimal.

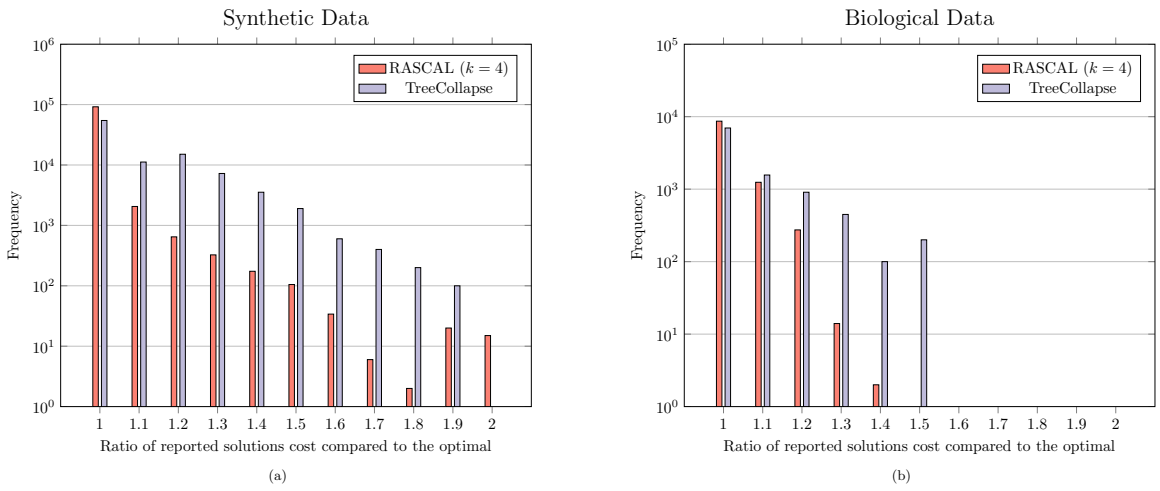

Figure 4: A comparison of the accuracy of TreeCollapse and RASCAL where k is a constant. Over both the synthetic (a) and biological (b) data sets RASCAL outperforms TreeCollapse. Of note is that not only does RASCAL find significantly more optimal solutions over both data sets (as also seen in Table 3) but that the distribution of solutions is skewed far more favourably to the left for RASCAL, demonstrating a significant accuracy increase when using the linear time version of RASCAL.

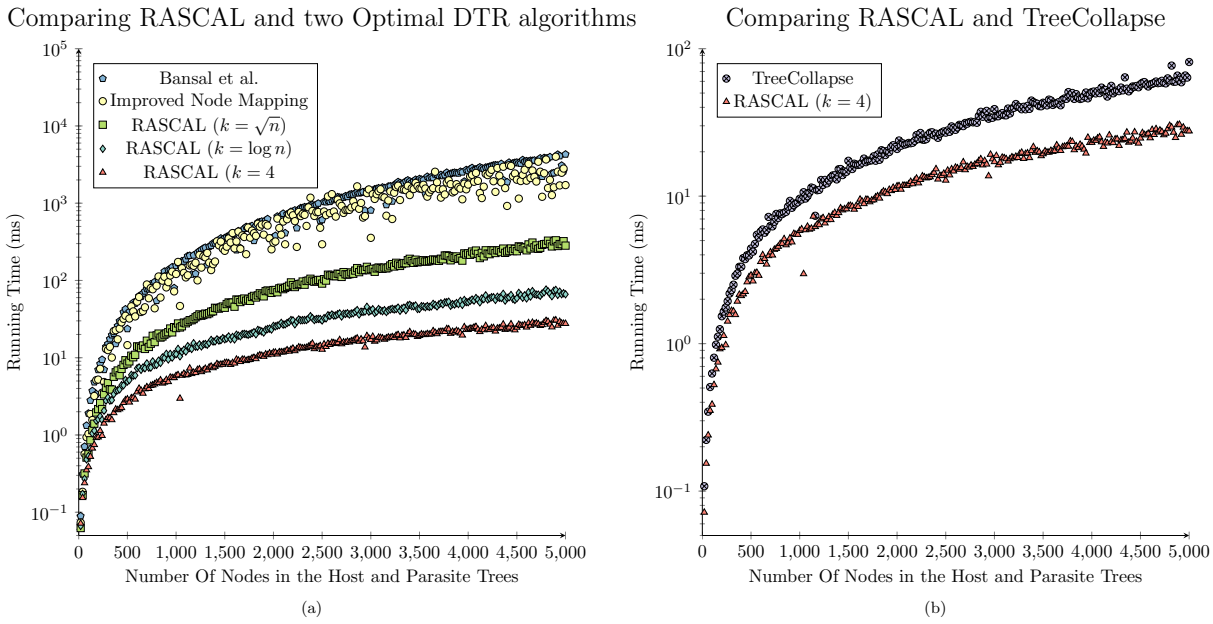

Figure 5: The running time of two optimal algorithms Bansal et al.'s DTL reconciliation algorithm, and Improved Node Mapping compared against three flavours of RASCAL ($\sqrt{n}$, $\log n$ and 4) (a), along with a comparison of TreeCollapse and RASCAL's linear time implementation, $k = 4$, (b). These results were obtained by recording each algorithms running time over 100 repeated runs for each synthetic coevolutionary system with the median running time recorded in the presented plots. These results show that compared to the fastest known algorithms that solve the DTR algorithm optimally that RASCAL provides a significant in-practice running time improvement. Further these results show that RASCAL, where $k = 4$ is the fastest algorithm that can estimate the cophylogeny reconstruction problem in linear time while still guaranteeing that solutions are time-consistent.