Capybara: equivalence ClAss enumeration of coPhylogenY event-BAsed ReconciliAtions

Yishu Wang, Arnaud Mary, Marie-France Sagot, Blerina Sinaimeri

To cite this version:

Yishu Wang, Arnaud Mary, Marie-France Sagot, Blerina Sinaimeri. Capybara: equivalence ClAss enumeration of coPhylogenY event-BAsed ReconciliAtions. Bioinformatics, Oxford University Press (OUP), 2020, 36 (14), pp.4197-4199. 10.1093/bioinformatics/btaa498 . hal-02917341

HAL Id: hal-02917341

https://hal.inria.fr/hal-02917341

Submitted on 19 Aug 2020

Capybara: equivalence ClAss enumeration of coPhylogenY event-BAsed ReconciliAtions

Yishu Wang [1,2,*], Arnaud Mary [1,2], Marie-France Sagot [1,2] and Blerina Sinaimeri [1,2,*]

[1] Université Claude Bernard Lyon 1; CNRS UMR5558, 43, Boulevard du 11 Novembre 1918, 69622, Villeurbanne, France.
[2] Inria Grenoble, 655, Avenue de l'Europe, 38334, Montbonnot, France.

*To whom correspondence should be addressed.

Abstract

Motivation: Phylogenetic tree reconciliation is the method of choice in analysing host-symbiont systems. Despite the many reconciliation tools that have been proposed in the literature, two main issues remain unresolved: (i) listing suboptimal solutions (*i.e.* whose score is "close" to the optimal ones) and (ii) listing only solutions that are biologically different "enough". The first issue arises because the optimal solutions are not always the ones biologically most significant; providing many suboptimal solutions as alternatives for the optimal ones is thus very useful. The second one is related to the difficulty to analyse an often huge number of optimal solutions. In this paper, we propose `Capybara` that addresses both of these problems in an efficient way. Furthermore, it includes a tool for visualising the solutions that significantly helps the user in the process of analysing the results.

Availability and implementation: The source code, documentation, and binaries for all platforms are freely available at `https://capybara-doc.readthedocs.io/`.

Contact: yishu.wang@univ-lyon1.fr, blerina.sinaimeri@inria.fr

Supplementary information: Supplementary data are available at *Bioinformatics* online.

1 Introduction

Nowadays, the most used model in analysing host-symbiont systems is the *phylogenetic tree reconciliation* (Charleston, 1998; Conow *et al.*, 2010; Page, 1994). An advantage of this model is that it is abstract enough to be applied to different types of data, of which the more common have been gene-species associations (Tofigh *et al.*, 2011; Bansal *et al.*, 2012; Stolzer *et al.*, 2012). Although the host-parasite and the gene-species contexts share many similarities, they also have some important differences. In particular in the host-symbionts context the same symbiotic species may interact, and therefore be associated with more than one host species while in the gene-species context a gene is naturally associated to exactly only one species (the one it is extracted from).

Phylogenetic tree reconciliation is usually modelled as a problem of mapping the phylogenetic tree S of the symbionts into the one of the hosts H. This mapping, called reconciliation, allows the recovery of four main macroevolutionary events: cospeciation, duplication, host-switch and loss (however that some tools account for additional events *e.g.* "failure to diverge" (Conow *et al.*, 2010)). One notion of optimality of the reconciliation simply assigns a cost to each of the different types of events and then seeks to minimise the total cost (computed in an additive way). If timing information (*i.e.* the order in which the speciation events occurred in the host phylogeny for incomparable nodes) is not known (or is considered as not sufficiently reliable to be used), as is often the case, finding an optimal reconciliation that ensures global time-feasibility is NP-hard (Ovadia *et al.*, 2011; Tofigh *et al.*, 2011). A way to deal with this is to require time-feasibility only locally and thus to allow for solutions that may be biologically unfeasible. In this case, the problem can be solved in polynomial time (*e.g.* Tofigh *et al.* (2011); Bansal *et al.* (2012)).

In the literature, there have been proposed different reconciliation software packages in the context of both host-symbionts and gene-species, such as `Eucalypt` (Donati *et al.*, 2015), `AngST` (David and Alm, 2011), `CoRe-Pa` (Merkle *et al.*, 2010), `Jane 4` (Conow *et al.*, 2010), `Mowgli` (Doyon *et al.*, 2011), `Notung` (Stolzer *et al.*, 2012), `Ranger-DTL` (Bansal *et al.*, 2012), `XSpace` Libeskind-Hadas *et al.* (2014) and `ecceTERA` (Jacox *et al.*, 2016). For a survey on the different features each method presents, and their possible limitations, see *e.g.* Donati *et al.*, 2015; Jacox *et al.*, 2016. Despite the considerable work done in the area, two main issues remain unsolved. The first is outputting or listing solutions that are "near" to the optimal. The motivation in this case is that sometimes these solutions may be more biologically significant than the optimal ones. Just to give an example, if for a given instance all the optimal solutions are time-unfeasible, then it is certainly important to be able to enumerate solutions that are "close" to the optimal if among them, at least one is time-feasible. Some first steps in this direction have been taken in `Jane 4` (Conow *et al.*, 2010) where the user is allowed to manually change the reconciliation in order to get a time-feasible one. However, there is no algorithm in the literature that produces all optimal and sub-optimal reconciliations.

The second issue is related to the complexity of the output. Indeed, in many cases the user has to deal with a number of optimal solutions that is unrealistically large, making it practically impossible to analyse each one of them separately. For instance, in `XSpace` (Libeskind-Hadas *et al.*, 2014) the authors focus on solutions that are Pareto-optimal.

However, for real datasets this number can be large and the algorithm although polynomial can be slow in practice. In this paper, we propose some different ways to define equivalence classes for grouping the reconciliations that may be considered "similar". Once this notion of equivalence is defined, one could group the optimal reconciliations in equivalence classes and output a single reconciliation for each class *without having to first generate all of the optimal reconciliations*.

We propose a tool called `Capybara` that as far as we know, addresses both issues for the first time and that can be used for different types of data, (such as for example host-symbionts or gene-species).

2 Methods

`Capybara` is based on the model presented in Tofigh *et al.* (2011) and Donati *et al.* (2015) (see the Supplementary Material). We are given an undated host tree H, a symbiont tree S and a function σ mapping the leaves of S to the leaves of H. We denote by $|H|$ and $|S|$ the total number of nodes in the host and symbiont trees, respectively.

`Capybara` includes all the basic features of the existing software such as: computing one (but possibly time-unfeasible) optimal solution in $O(|H|^2|S|)$ time, computing the number of optimal solutions in $O(|H|^2|S|)$ time and listing *all* the optimal reconciliations in polynomial delay. We recall that the computational time of a listing algorithm depends on the number of items to output (which may be exponential), and a way to measure the efficiency of such an algorithm is to evaluate the delay between two consecutive outputs. Our algorithm needs $O(|H|^2|S|)$ to produce the first reconciliation and then only $O(|S|)$ time to list each subsequent one. `Capybara` ensures only the local time-feasibility but includes a filtering procedure based on Stolzer *et al.* (2012) and Donati *et al.* (2015) to decide the global time-feasibility of a given reconciliation in $O(|H|^2)$ time. The novelty of `Capybara` rests on the features described next that, to our knowledge, have not been considered in the literature.

Listing suboptimal reconciliations For a given input, one can hypothetically consider all the possible reconciliations in an increasing order based on their costs (the ordering between solutions having the same cost is arbitrary). Given an integer K, `Capybara` efficiently outputs the first K reconciliations in this order. If K is larger than the number of optimal reconciliations, then `Capybara` outputs all optimal reconciliations and also sub-optimal ones. The algorithm is a non trivial extension of the dynamic programming algorithm for listing all optimal reconciliations. The DP algorithm works in two steps: first it fills a dynamic programming matrix of size $|S| \times |H|$, then it traverses the matrix through backtrack arcs to output the solutions one by one with a delay of $O(|S|)$ (for more details we refer to Donati *et al.* (2015)). To extend the algorithm, we modified both of these steps. We first modified the way in which the dynamic programming matrix is filled. Each cell of the $|S| \times |H|$, matrix, labeled by a symbiont/host association $(s : h)$, instead of representing all reconciliations of the subtree of S rooted at s mapping s to h and having minimum cost, in our algorithm represents at most K such reconciliations of the subtree sorted by cost. At any given cell $(s : h)$, we can compute the K best costs only from the combinations of the K best costs of the two subproblems for the two child subtrees of s. After combining the two subproblems in every possible way, we then take all the combinations that achieve the K best costs while keeping them in sorted order.

The most important modification is in the backtrack structure of the DP algorithm of `Eucalypt`. To combine the solutions of the two subproblems, we need to add the costs of two previously filled cells, and thus we need to compute the K best values in the Cartesian sum of two lists of costs in these cells. Each list has size $O(K)$ (as we keep the first K solutions). We use an algorithm proposed in Frederickson and Johnson (1984) that solves in $O(K)$ time the problem of selecting the first K elements in the Cartesian sum of two sorted lists of size $O(K)$. After computing the $O(|H|)$ combinations of subproblems for a given cell, we obtain $O(|H|)$ sorted lists of costs, of size $O(K)$ each, and we can take the K best costs using a simple merge procedure. In summary, the preprocessing step requires $O(K|H|^2|S|)$ computational time and listing each reconciliation takes $O(|S|)$ time.

Listing equivalence classes Many previous studies pointed out the fact that, even when the time-feasibility filter is applied, and even for very small input size, in some cases the number of reconciliations of minimum cost can be huge (see, *e.g.* Donati *et al.* (2015); Huber *et al.* (2018a,b)). Many of these reconciliations are very similar and thus may be considered equivalent. A user would wish to look at one single solution in each equivalence class and thus to only consider solutions that are "different enough", thereby getting an overview of the diversity of all optimal solutions. We defined three different notions of equivalence between reconciliations.

a. Event vectors The *event vector* of a reconciliation is a vector of four integers representing the number of occurrences of cospeciation, duplication, host-switch, and loss events in it. Two reconciliations are *event-vector equivalent* if they have the same event vectors (*i.e.* if the number of each event is the same for both of them). Event vectors are already used in the literature as for example in `XSpace` (Libeskind-Hadas *et al.*, 2014) where reconciliations are identified by their event vector. The frequency of each of the events provides already some information about their coevolution. For instance, a high number of cospeciations may be an indication of coevolution, a high number of host-switches may indicate that the symbiont may have a high degree of flexibility in the host choice.

b. Event partitions The *event partition* of a reconciliation is a partition of the internal nodes of the symbiont tree into three subsets according to the event with which the nodes are associated: cospeciation, duplication, and host-switch. Two reconciliations are *event-partition equivalent* if, for each internal node in the symbiont tree, the event (cospeciation, duplication, host-switch) assigned by each of the two reconciliations is the same. In this case, we are interested not only in the number of the events but also in where the symbiont tree these events have taken place.

c. Cospeciation-duplication equivalence Two reconciliations are *cospeciation-duplication-equivalent* if they are event-partition equivalent and each internal node of the symbiont tree that is associated with a cospeciation or duplication is mapped to the same host node by both reconciliations. In other words, two reconciliations in the same class can differ only in the host nodes to which are mapped those symbiont nodes associated with the host-switch events.

A naive algorithm might list all the optimal reconciliations and then group them in equivalence classes. Unfortunately, this is not always feasible in practice, as the number of optimal solutions can be too large to be listed. For instance, for the dataset *Wolbachia* and their arthropod hosts (Simões *et al.*, 2011; Simões, 2012) where each tree has 387 leaves, one can reach $\approx 10^{40}$ optimal reconciliations. We then developed an algorithm that is able to efficiently enumerate only one representative per equivalence class without having to first generate all the reconciliations (for the theoretical aspects of the algorithm see Wang *et al.* (2020)).

Visualising Reconciliations `Capybara` includes an animated web tool, `Capybara Viewer`, that allows to visualise each equivalence class as a coloured symbiont tree. The internal nodes of the symbiont tree are coloured according to the event. This is particularly useful in the case of large trees. In such cases indeed, the colours will allow a global view of the distribution of the events in the tree. More details can be found in `https://capybara-doc.readthedocs.io/`.

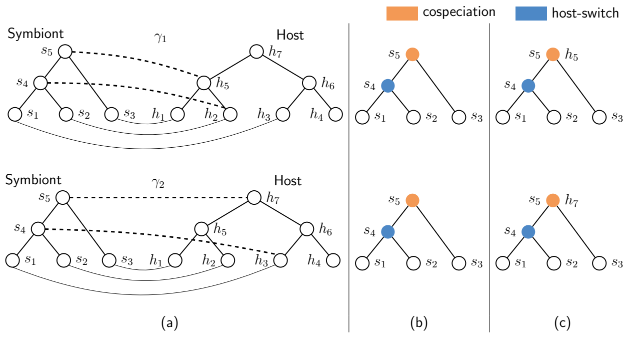

Fig. 1. Example of two reconciliations for the same dataset. Notice that the event vectors for the first and the second reconciliation are $(1, 0, 1, 0)$ and $(1, 0, 1, 2)$, respectively, thus they are not event-vector equivalent. In (b) and (c) we show that the two reconciliations are event-partition equivalent but not cospeciation-duplication-equivalent. The internal nodes of the symbiont tree are coloured according to the event. The correspondence between colour and event is shown in the legend.

3 Discussion

We describe as a proof-of-concept the utility of Capybara by applying it to seven real datasets (see the Supplementary Material). Concerning the usefulness of listing also suboptimal solutions, we consider the cases where all the optimal solutions are time-unfeasible. We then switch to the K-best algorithm and try to find time-feasible reconciliations of suboptimal costs by setting the parameter K to be strictly larger than the number of solutions of minimum cost. We observed that, even though this number can be exponentially large in theory, in practice it is possible to find suboptimal solutions that are time-feasible by setting K to a value that is approximately ten times greater than the number of optimal solutions. For the dataset *SFC* and a typical cost vector $(-1, 1, 1, 1)$, there were 40 optimal reconciliations and none of them was time-feasible. By listing the first 1000 solutions, we have already ≈ 330 time-feasible ones (see Table 1 in the Supplementary Material).

To see the utility of the equivalence classes, we focus on the *Wolbachia* dataset. This dataset with all the cost vectors considered has at least $\approx 10^{42}$ optimal solutions which are thus impossible to list. Nevertheless, if we look for instance at the number of optimal event vectors, there are at most 11 (see Table 2 and 3 in the Supplementary Material). Capybara is able to list these optimal event vectors *without* listing all the optimal reconciliations. By doing this, we observe that the dataset can be explained by a large number of host-switches and cospeciations, and there have been probably no duplication. This already gives an idea of the variety of the solutions obtained and is thus helpful in drawing conclusions about the coevolutionary history of the system. Finally, Capybara is efficient in practice, for example to list the event vectors and to count the number of reconciliations for each one of them (which is generally the most expensive task), it took 0.25s for the dataset *SFC* and less than 3 hours for *Wolbachia* on a single core of a 6-core MacBook Pro 2.2 GHz i7. Furthermore, we have released a Python package which includes the essential Capybara. This allows the automatic processing of a large number of files and the use of interpreters such as PyPy3.6 to speed up the computation.

4 Conclusion

We propose Capybara which for the fist time addresses two main issues that have remained unsolved in the area: (i) listing suboptimal solutions, (ii) listing only solutions that are biologically different “enough”. We introduce three definitions of biologically equivalent reconciliations and test the practical effectiveness of Capybara on five real datasets. Future extensions will include in particular other biologically inspired definitions of equivalence between reconciliations and better visualisation techniques that will help with the analysis of the output. Concerning the latter, it is particularly interesting to find ways to represent multiple event partitions or equivalence classes in a compact manner.

References

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

Charleston, M. A. (1998). Jungles: a new solution to the host/parasite phylogeny reconciliation problem. *Mathematical Biosciences*, **149**(2), 191–223.

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

David, L. and Alm, E. (2011). Rapid evolutionary innovation during an archaean genetic expansion. *Nature*, **469**, 93–96.

Donati, B., Baudet, C., Sinaimeri, B., Crescenzi, P., and Sagot, M. (2015). Eucalypt: efficient tree reconciliation enumerator. *Algorithms for Molecular Biology*, **10**(1).

Doyon, J.-P., Scornavacca, C., Gorbunov, K. Y., Szöllősi, G. J., Ranwez, V., and Berry, V. (2011). An efficient algorithm for gene/species trees parsimonious reconciliation with losses, duplications and transfers. In E. Tannier, editor, *Proceedings of the 8th annual RECOMB Satellite Workshop on Comparative Genomics (RECOMB-CG 2010)*, volume 6398 of *Lecture Notes in Bioinformatics*, pages 93–108, Ottawa, Canada. Spring-Verlag Berlin Heidelberg.

Frederickson, G. N. and Johnson, D. B. (1984). Generalized selection and ranking: Sorted matrices. *SIAM J. Comput.*, **13**(1), 14–30.

Huber, K. T., Moulton, V., Sagot, M.-F., and Sinaimeri, B. (2018a). Exploring and Visualizing Spaces of Tree Reconciliations. *Systematic Biology*, **68**(4), 607–618.

Huber, K. T., Moulton, V., Sagot, M.-F., and Sinaimeri, B. (2018b). Geometric medians in reconciliation spaces of phylogenetic trees. *Information Processing Letters*, **136**, 96 – 101.

Jacox, E., Chauve, C., Szöllősi, G. J., Ponty, Y., and Scornavacca, C. (2016). ecceTERA: Comprehensive gene tree-species tree reconciliation using parsimony. *Bioinformatics*.

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

Merkle, D., Middendorf, M., and Wieseke, N. (2010). A parameter-adaptive dynamic programming approach for inferring cophylogenies. *BMC Bioinformatics*, **11**(Supplementary 1), 10 pages.

Ovadia, Y., Fielder, D., Conow, C., and Libeskind-Hadas, R. (2011). The cophylogeny reconstruction problem is np-complete. *Journal of Computational Biology*, **18**(1), 59–65.

Page, R. D. M. (1994). Parallel phylogenies: reconstructing the history of host-parasite assemblages. *Cladistics*, **10**(2), 155–173.

Simões, P. M. (2012). *Diversity and dynamics of Wolbachia-host associations in arthropods from the Society archipelago, French Polynesia*. Ph.D. thesis, University of Lyon 1, France.

Simões, P. M., Mialdea, G., Reiss, D., Sagot, M.-F., and Charlat, S. (2011). *Wolbachia* detection: an assessment of standard PCR protocols. *Mol. Ecol. Resour.*, **11**(3), 567–572.

Stolzer, M. L., Lai, H., Xu, M., Sathaye, D., Vernot, B., and Durand, D. (2012). Inferring duplications, losses, transfers and incomplete lineage sorting with nonbinary species trees. *Bioinformatics*, **28**(18), i409–i415.

Tofigh, A., Hallett, M., and Lagergren, J. (2011). Simultaneous identification of duplications and lateral gene transfers. *Journal of IEEE/ACM Transactions on Computational Biology and Bioinformatics*, **8**(2), 517–535.

Wang, Y., Mary, A., Sagot, M.-F., and Sinaimeri, B. (2020). Lazy listing of equivalence classes – a paper on dynamic programming and tropical circuits.