Polynomial Supertree Methods Revisited

Malte Brinkmeyer, Thasso Griebel, and Sebastian Böcker

Department of Computer Science, Friedrich Schiller University, 07743 Jena, Germany
{malte.b,thasso.griebel,sebastian.boecker}@uni-jena.de

Abstract. Supertree methods allow to reconstruct large phylogenetic trees by combining smaller trees with overlapping leaf sets, into one, more comprehensive supertree. The most commonly used supertree method, matrix representation with parsimony (MRP), produces accurate supertrees but is rather slow due to the underlying hard optimization problem. In this paper, we present an extensive simulation study comparing the performance of MRP and the polynomial supertree methods *MinCut Supertree*, *Modified MinCut Supertree*, *Build-with-distances*, *PhySIC*, and *PhySIC_IST*. We consider both quality and resolution of the reconstructed supertrees. Our findings illustrate the trade-off between accuracy and running time in supertree construction, as well as the pros and cons of voting- and veto-based supertree approaches.

1 Introduction

In recent years, supertree methods have become a familiar tool for building large phylogenetic trees. Supertree approaches combine input trees with overlapping taxa sets into one large and more comprehensive tree. Since the introduction of the term *supertree* and the first formal supertree method [1], there has been a continuous development of supertree methods, see e.g. [2]. The supertree approach has certain advantages over standard phylogenetic reconstruction methods, both on the theoretical and practical side [3]: It allows to combine heterogeneous data sources, such as DNA hybridization data, morphological data, and protein sequences. Furthermore, it enables inference for groups where most species are represented by very few genes and sequences, and the major part of sequences is available only for few species, which makes deriving a balanced molecular phylogeny difficult. On the theoretical side, it is well known that inferring optimal trees from sequences is a computationally hard problem under the maximum likelihood (ML) [4] and the maximum parsimony (MP) criterion [5], so we have to rely on heuristics that cannot guarantee to find the optimal solution. Even for a moderate number of species, the sheer size of tree space prohibits to search for optimal trees under these criteria. Current supertree methods can roughly be subdivided into two major families: matrix representation (MR) and polynomial, mostly graph-based methods. The former encode the inner vertices of all input trees as partial binary characters in a matrix, which is analyzed using an optimization or agreement criterion to yield the supertree. Matrix representation with parsimony (MRP) [6,7], the first matrix-based method,

T.M.H. Dijkstra et al. (Eds.): PRIB 2010, LNBI 6282, pp. 183–194, 2010.

is still by far the most widely used supertree method today. Other variants have been proposed using different optimization criteria, e.g. matrix representation with flipping (MRF) [8] or matrix representation with compatibility (MRC) [9]. All MR methods have in common that the underlying optimization problems are computationally hard, and heuristic search strategies have to be used. As for ML and MP, it is unclear how close the resulting tree is to the optimal one.

Graph-based methods make use of a graph to encode the topological information given by the input trees. This graph is used as a guiding structure to build the supertree top-down from the root to the leaves. The *MinCut Supertree* algorithm (MC) [10] and a modified version, *Modified MinCut Supertree* (MMC) [11], use a minimum-cut approach to construct a supertree if the input trees are conflicting. The *Build-with-distances* algorithm (BWD) [12] is the first graph-based method that uses branch length information from the input trees to build the supertree. Ranwez et al. [13] presented a new graph-based method, the *PhySIC* algorithm. The method ensures that the reconstructed supertree satisfies two properties: it contains no clade that directly or indirectly contradicts the input trees and each clade in the supertree is present in an input tree, or is collectively induced by several input trees. Supertree methods guaranteeing the first property are called *veto* methods, that, in case of highly conflicting and/or poorly overlapping input trees, tend to produce unresolved supertrees. Scornavacca et al. [14] presented a modified version of *PhySIC*, *PhySIC_IST*, that tries to overcome this drawback by proposing non-plenary supertrees (i.e. supertrees that do not necessarily contain all taxa from the input trees), while still assuring the properties mentioned above. *PhySIC_IST* works in a stepwise fashion, iteratively adding leaves to a starting tree consisting of two nodes. In contrast to MR methods, the MC, MMC, BWD, *PhySIC* and *PhySIC_IST* algorithms have polynomial running time.

As an increasing number of supertree methods is available, simulation studies are needed to compare the behavior and performance of the methods under various conditions. The advantage of simulation studies is that the results of different methods can be compared to a known model tree and thus the methods can be compared at an absolute scale. Although several simulation studies focusing on different aspects of the investigated supertree have been carried out (e.g. [15], [16]), they have only just begun to provide useful comparisons of alternative methods. This paper focuses a special subset of supertree construction methods: we are in particular interested in the comparison of the accuracy of the MRP method as exponent of the MR based family of supertree methods, for which it has been shown that they are accurate and highly resolved but require long running times, and the mentioned polynomial supertree methods, which are swift but possibly less accurate and in case of *PhySIC* and *PhySIC_IST*, also possibly less resolved. Here, we present a large-scale simulation study conducted to compare the accuracy and the resolution of MRP, MC, MMC, BWD, *PhySIC*, and *PhySIC_IST* supertrees. Additionally, we explore new variations of BWD, trying to improve its performance. Our simulation study follows the established general scheme to assess the performance of supertree methods: (1) Construction

of a model tree under a Yule process, (2) simulation of DNA alignments along that tree, (3) random deletion of a proportion of taxa (4) reconstruction of trees by ML, (5) construction of supertree from the inferred ML trees, and, finally (6) comparison of the supertree to the model tree using distance and similarity measures and evaluation of its resolution. Our results demonstrate that the BWD and the *PhySIC_IST* method perform significantly better than MC and MMC, and are, with respect to the accuracy of the reconstructed supertree, sometimes even comparable with MRP. Moreover, as we also consider the resolution of the supertrees, our findings illuminate the trade-off between accuracy and running time in supertree construction, as well as the pros and cons of voting and veto approaches.

2 Methods under Consideration

Build and MinCut supertrees. The first graph-based supertree method is the *Build* algorithm [17], an all-or-nothing approach that encodes the input trees into a graph structure and returns a supertree only if the input trees are compatible. The *MinCut Supertree* algorithm (MC) [10] was the first extension of *Build* capable of returning a supertree if the input trees are not compatible. The incompatibilities are resolved by deleting a minimal amount of information present in the input trees in order to allow the algorithm to proceed. Page [11] presented a modified version of MC that uses more information from the input trees. By using a variation of the underlying graph structure, the *Modified MinCut Supertree* (MMC) algorithm ensures to incorporate all clades from the input trees with which no single tree directly disagrees.

Build-with-distances supertrees. Willson [12] presented another extension of *Build*, the *Build-with-distances* (BWD) algorithm that, in addition to the branching information in the input trees, uses branch lengths to build the supertree. Basically, the method follows the same recursive schema as *Build*, MC, and MMC. The main observation underlying the BWD algorithm is that branch lengths may carry phylogenetic information, such as an estimated number of mutations. Clearly, the use of branch length is only justified if these are comparable amongst the input trees, i.e. the input to the method has to be carefully selected, or the branch lengths have to be reconciled or normalized in some way. The BWD algorithm incorporates branch lengths from the input trees to add more information to the used graph. BWD uses different *support functions*, which basically estimate the evidence that two taxa should be in the same clade of the supertree. We find that in our simulation study using the *accumulated confirmed support function* (SAC) consistently outperforms other support functions. Hence, we will concentrate on SAC in our evaluations as well as a new established support function, SACmax. Details are deferred to the full version of this paper. In contrast to the minimum-cut approach used by MC and MMC, Willson uses the *bisection method* to deal with incompatible input trees.

PhySIC and PhySIC_IST supertrees. Unlike all methods mentioned before, the *PhySIC* algorithm [13] applies a *veto* philosophy. Following Ranwez et al. [13], supertree methods are either *voting* or *veto* procedures. A characteristic of the voting approach is that the input trees are asked to vote for clades in the phylogeny to be inferred; the most frequent alternatives are chosen. Voting methods resolve conflicts by using an optimization criterion in order to select between different possible topologies [18]. When input trees conflict, voting methods as MRP can infer supertrees in which clades are present that are contradicted by each of the input trees (e.g. [19]). In contrast to voting methods, the veto approach is more conservative in handling conflicts among the input trees: the inferred supertree has to respect the phylogenetic information of each source tree and is not allowed to contain any clade that is contradicted by one or more of the input trees. Thus, conflicts among the input trees are removed [18], for example by proposing multifurcations in the supertree or by pruning rogue taxa. Scornavacca et al. [14] presented *PhySIC_IST*, a modification of the *PhySIC* algorithm, aiming to circumvent a main drawback of veto supertree methods: These tend to return highly unresolved supertrees if the input trees imply a high degree of incompatibility, or do not have a high degree of overlap. To overcome this shortcoming, *PhySIC_IST* modifies the original approach non-plenary supertrees (i.e. supertrees that do not necessarily contain all taxa present in the input trees) and by using a preprocessing step called STC (Source tree correction), which analyzes and modifies the input trees concerning the conflicts they contain. Basically, it removes parts of each source tree that significantly conflict with other source trees.

Matrix Representation with Parsimony (MRP). MRP encodes the inner vertices of all input trees as partial binary characters in a matrix, which is analyzed using the parsimony criterion as objective function. Two different coding schemes have been suggested to decompose trees into an matrix representation: the Baum-Ragan (BR) and the Purvis (PU) coding scheme. Furthermore, two kinds of parsimony can be used: reversible Fitch parsimony and irreversible Camin-Sokal parsimony. MRP with BR and Fitch is commonly used and generally accepted as standard method for supertree construction.

3 Simulation Study

In this section we present a large scale simulation study conducted to evaluate the accuracy and resolution of the methods MRP, MC, MMC, *PhySIC*, *PhySIC_IST*, and BWD (with modifications). An overview of the simulation layout can be found in Figure 1. Each step is described in detail below.

Generating Model Trees and DNA Sequences. We generated model trees according to a stochastic Yule birth process using the default parameters of the YULE_C procedure from the program r8s [20] with either 48, 96 and 144 taxa. For each model tree size we generated 100 different model tree replicates. By the use of the program Seq-gen v1.3.2 [21], nucleotide sequences were simulated

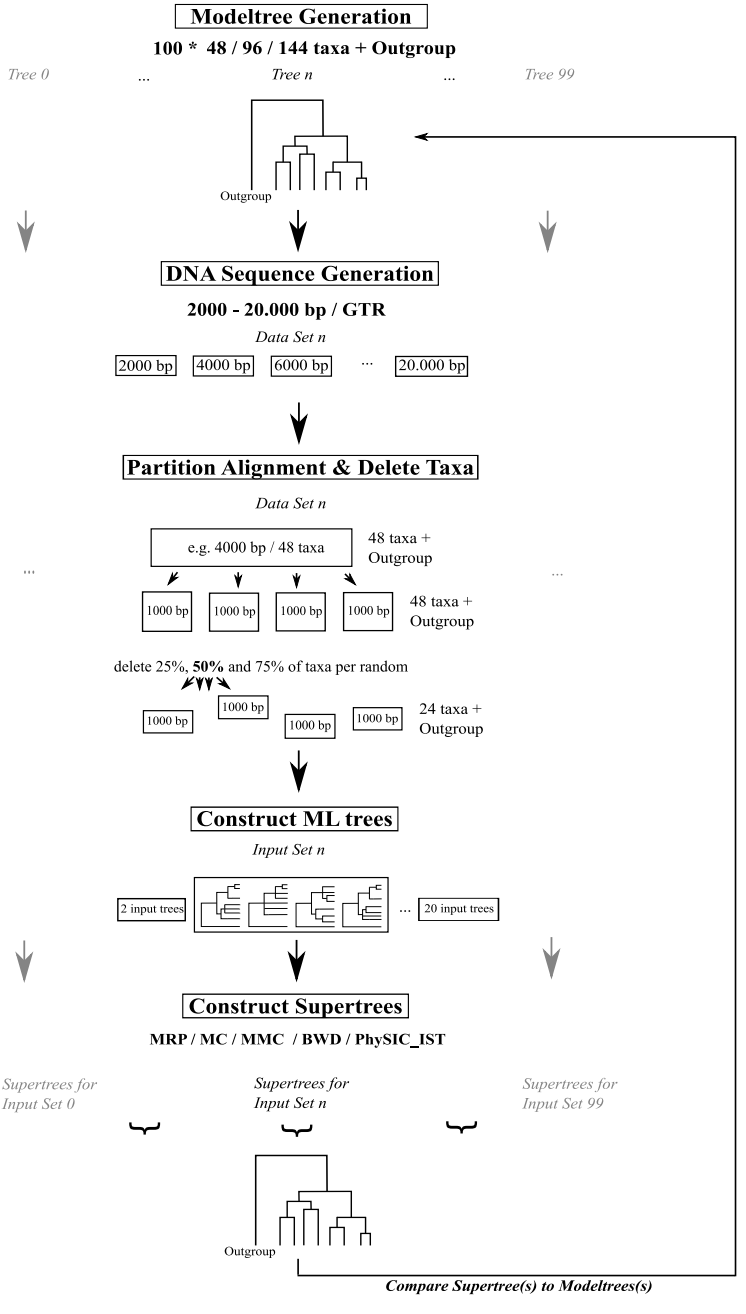

Fig. 1. Simulation pipeline overview

along each of the model trees according to the general time reversible process (GTR) model [22] with parameters Lset Base = (0.3468 0.3594 0.0805), Rmat = (0.6750 27.9597 1.1677 0.4547 20.8760), gamma rate heterogeneity $\alpha = 1.1999$ and PINVAR = 0.4954, taken from [23]. For each model tree we generated sequences ranging from 2000 to 20000 base pairs in steps of 2000, yielding in ten different sequence alignments per model tree.

Generating Input Trees. All models of molecular substitution implemented in Seq-Gen assume evolution is independent and identical at each site. Hence, contiguous blocks of sequences represent randomly subdivided data set. We partitioned each alignment into blocks of 1000-base pair data sets and randomly deleted 25%, 50% and 75% of sequences from each alignment to simulate different taxa overlaps observed in real data sets. For each resulting alignment block we inferred a maximum likelihood tree using RAxML v 7.0.0. [24] with default parameters. This yields in sets ranging from 2 to 20 input trees belonging to one model tree.

Supertree construction. MRP supertrees were estimated using PAUP* 4.0b10 [25] with TBR branch swapping as heuristic search, random addition of sequences and a maximum 10.000 trees in memory. The search time for a single MRP supertree run was delimited by 300 seconds. The strict consensus tree of all most-parsimonious trees was used as final MRP tree. We computed MC as well as the BWD supertrees using our own implementations embedded in our software framework EPoS[1]. MMC trees were generated using Rod Page's implementation[2]. For the *PhySIC* and *PhySIC_IST* supertrees we used the implementations provided from the authors of the corresponding papers[3][4]. To test a broader range of the *PhySIC_IST* STC preprocess (-c option), we used 0, 0.5 and 1 as parameters. In our setting, the results for 0 and 0.5 are similar; therefore, only the 0 and 1 parameter results are shown. In the following we will refer these as *PhySIC IST 0* and *PhySIC IST 1*.

Measuring accuracy and resolution. To evaluate the accuracy of the supertrees build by the different methods we compared the supertrees to the model trees using different distance and similarity scores, namely the Robinson-Foulds metric (*RF distance*) [26], the maximum agreement subtree score, *MAST score* [27], and the *triplet distance* [11]. We stress that each of these methods has its particular shortcomings, for a discussion and implementation details see the full version of this paper. The resolution was measured as the number of clades in the inferred supertree relative to the total number of clades on a fully binary tree of the same size (n - 2 for an unrooted tree, where n = number of taxa). Resolution varies between 0 and 1, where 0 indicates a unresolved bush and 1 indicates a complete binary supertree.

[1] `http://bio.informatik.uni-jena.de/epos/`
[2] `http://darwin.zoology.gla.ac.uk/~rpage/supertree/`
[3] `http://www.atgc-montpellier.fr/physic/binaries.php`
[4] `http://www.atgc-montpellier.fr/physic_ist/`

4 Results

Results of our simulation for 48 taxa are reported in Figure 2, where we plot resolution and triplet distance against the number of input trees. In Figure 3, we use our simulations on 96 taxa and plot MAST score and RF distance against number of input trees. One would expect that results improve if more input data becomes available, as this helps us to identify bogus information. Hence, triplet distance and RF distance should decrease, whereas the MAST score should increase when more input trees are available to the supertree method. We now discuss the observed patterns in more detail.

Resolution. In our setting *PhySIC* mostly returns star trees. The two variations of the BWD algorithm build the most resolved supertrees compared to all other methods, independent from the deletion frequency the number of input trees. In general, this also holds for MMC and MC. In case of 25% deletion frequency, MRP behaves similar to MMC and MC, but is significantly less resolved than all others at 75% deletion frequency. In case of 25% and 50% deletion frequency *PhySIC_IST* 0 produces more resolved supertrees than *PhySIC_IST* 1. In comparison to all methods, the *PhySIC_IST* 1 supertrees are least resolved. With 75% deletion frequency, the resolutions of the *PhySIC_IST* 0 and *PhySIC_IST* 1 supertrees are quite similar. In general, one can see that BWD as an advanced graph-based supertree method outperforms the classical parsimony approach (MRP) as well as the conservative, veto based algorithm (*PhySIC_IST*) in terms of resolution. The results also clearly show that the more conservative *PhySIC_IST* 1 produces less resolved trees than *PhySIC_IST* 0, reflecting the influence of the STC parameter.

Triplet Distance. In the majority of cases, MC algorithm performs worst compared to all other algorithms and an increasing number input of trees has no positive effect on the accuracy. The MMC algorithm generally performs better than MC, but its accuracy also does not significantly increase with the number of input trees, except for the case of 25% deletion frequency. Both BWD methods perform better than MC/MMC but their accuracy also does not significantly benefit from a growing number of input trees. In case of 25% and 50% deletion frequency, *PhySIC_IST* 1 produces less accurate supertrees with an increasing number of input trees. This can be explained by the decreasing resolution, which has direct impact on the number of matching triplets. In contrast, the accuracy of *PhySIC_IST* 0 is relatively stable and independent of the deletion frequency and the number of input trees. MRP always performs better than the algorithms mentioned so far. The number of input trees has in general a slight positive effect on the accuracy.

MAST score. In general, the MC algorithm provides supertrees with the worst MAST score compared to all other methods. Only in the case of 25% deletion frequency MC performs slightly better than *PhySIC_IST* 1. *PhySIC_IST* 1 behaves generally like the MC algorithm. *PhySIC_IST* 0 produces supertrees with

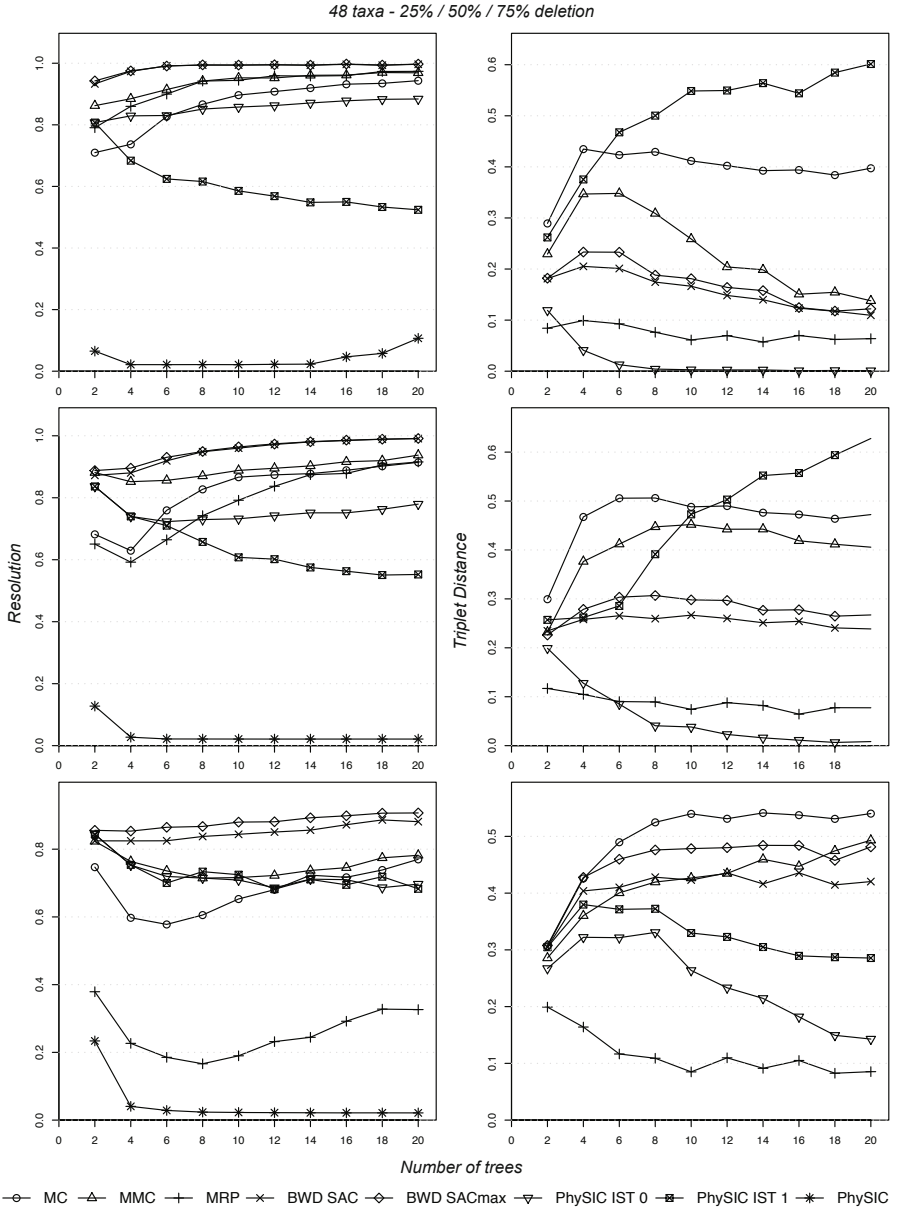

Fig. 2. The left column of the figure shows the average resolution of the supertrees constructed from model trees with 48 taxa and different taxon deletion rates (top 25%, middle 50%, bottom 75%). The right column shows the average triplet distances of the supertrees constructed from model trees with 48 taxa and different taxon deletion rates (top 25%, middle 50%, bottom 75%).

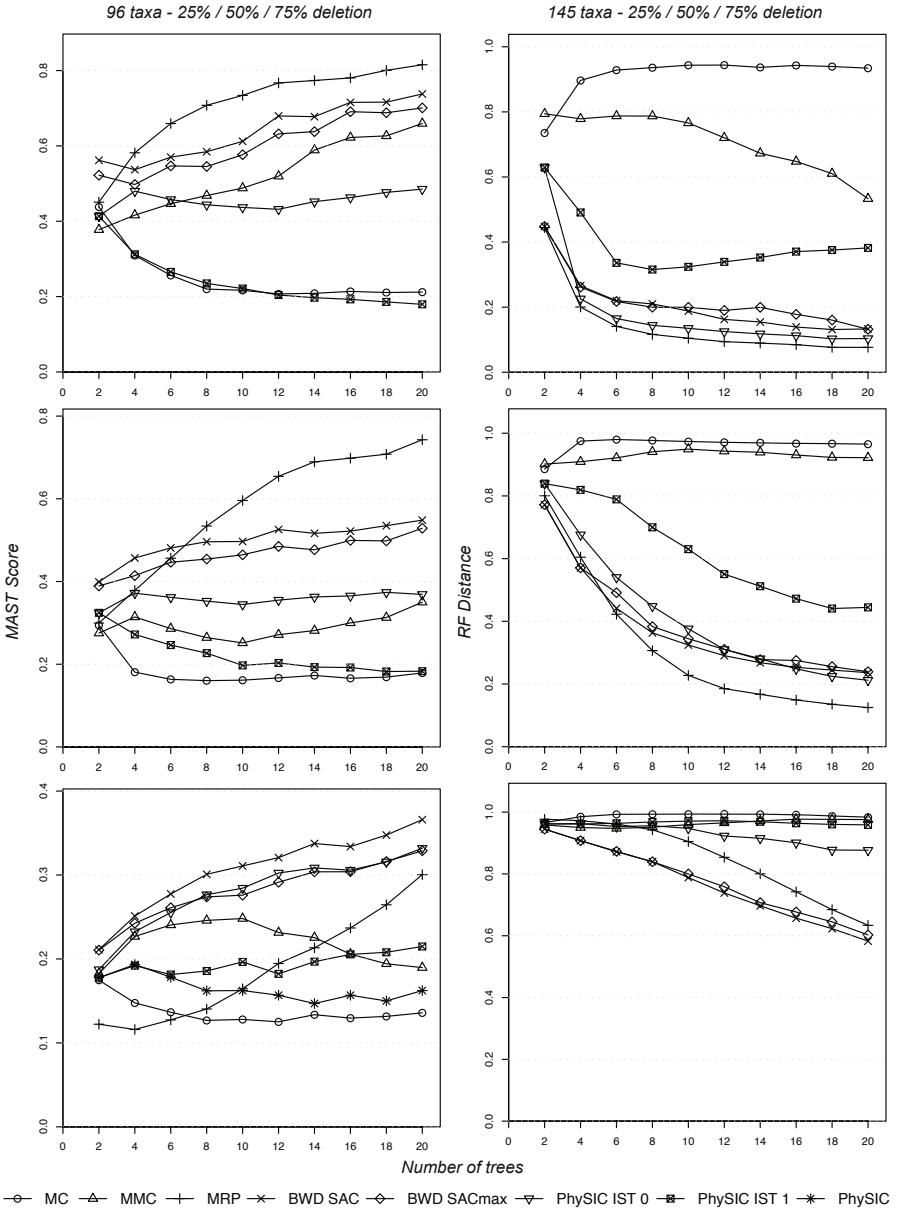

Fig. 3. The left column of the figure shows the average MAST scores of the supertrees constructed from model trees with 96 taxa and different taxon deletion rates (top 25%, middle 50%, bottom 75%). The right column shows the average RF-Distance of the supertrees constructed from model trees with 145 taxa. Note that the MAST values are similarity scores and RF values are distances.

a considerably better MAST scores than MC and *PhySIC_IST* 1, but the number of input trees has no significant effect on the MAST score. MMC algorithm performs slightly better than *PhySIC_IST* 0 and 1 as well as the MC method in case of 25% deletion frequency With 25% deletion frequency MMC's MAST score increases with more input, in both other cases the score is relatively constant. The MRP method performs better than all other methods in the case of 25% and 50% deletion frequency and significantly benefits from a growing number of source trees. With 75% deletion frequency the MAST score of all methods under consideration are quite low and MRP can only outperform *PhySIC_IST* 1, *PhySIC_IST* 0, MC and MMC with a large number of input trees. For 75% deletion frequency, the BWD methods outperform MRP and show an increasing MAST score with an increasing number of input trees. With 25% and 50% deletion frequency, both BWD methods are only outperformed by MRP. In both cases the number of input trees has a positive effect on the MAST score.

RF distance. For all combinations of model tree sizes and deletion probabilities, the MC methods performs worst compared to all other methods. As with the triplet distance and the MAST score, MMC shows an improvement over the original method. The *PhySIC_IST* 1 performs generally better than MC and MMC. The number of input trees has in general a positive effect on the RF distance. In case of 25% and 50% deletion frequency all other methods perform similar, although MRP produces slightly better results.

5 Conclusion

We have presented a large-scale simulation study to assess and compare the accuracy and the resolution of polynomial supertree methods and the *de facto* standard supertree method MRP. Our results show that recent, polynomial supertree methods can sometimes compete with the classical MRP approach while providing a significantly better running time (which did not exceed a few seconds for all polynomial methods). The BWD method that incorporates branch length information from the input trees, significantly enhances the graph-based approaches concerning accuracy and resolution, without sacrificing short running times. For example, the MAST score at 75% deletion (Fig. 3 left) is consistently better for BWD than for MRP, for any number of input trees. Veto approach such as *PhySIC* have certain appealing properties but also certain drawbacks: the resolution of reconstructed supertree rapidly decreases when there are too many conflicts among input trees, and/or small taxon overlap. *PhySIC_IST*, in combination with the STC preprocessing, significantly enhances the veto approach in terms of resolution and accuracy, but at the cost that taxa are not included in the supertree.

For medium-sized studies with hundreds of taxa and tens of trees, we propose to use several of the supertree methods presented here, and to manually compare the results. But when the sheer size of the problem renders it impossible to use matrix-representation methods such as MRP, then novel polynomial-time methods such as BWD and *PhySIC_IST* will greatly improve the quality of results,

compared to early methods such as MC or MMC. Although formal supertree methods have been around for a quarter of a century, our simulation also show that there is still much room for improvement, and that novel ideas and methods can greatly improve the quality of constructed supertree.

References

1. Gordon, A.D.: Consensus supertrees: The synthesis of rooted trees containing overlapping sets of labelled leaves. J. Classif. 3, 335–348 (1986)
2. Bininda-Emonds, O.R.P. (ed.): Phylogenetic Supertrees: Combining Information to Reveal the Tree of Life. Computational Biology Book Series, vol. 4. Kluwer Academic, Dordrecht (2004)
3. Bininda-Emonds, O.R.P.: Supertree construction in the genomic age. Methods Enzymol. 395, 745–757 (2005)
4. Roch, S.: A short proof that phylogenetic tree reconstruction by maximum likelihood is hard. IEEE/ACM Trans. Comput. Biol. Bioinform. 3(1), 92–94 (2006)
5. Foulds, L.R., Graham, R.L.: The Steiner problem in phylogeny is NP-complete. Adv. Appl. Math. 3, 43–49 (1982)
6. Baum, B.R.: Combining trees as a way of combining data sets for phylogenetic inference, and the desirability of combining gene trees. Taxon 41(1), 3–10 (1992)
7. Ragan, M.A.: Matrix representation in reconstructing phylogenetic relationships among the eukaryotes. Biosystems 28(1-3), 47–55 (1992)
8. Chen, D., Eulenstein, O., Fernández-Baca, D., Sanderson, M.: Minimum-flip supertrees: complexity and algorithms. IEEE/ACM Trans. Comput. Biol. Bioinform. 3(2), 165–173 (2006)
9. Ross, H.A., Rodrigo, A.G.: An assessment of matrix representation with compatibility in supertree construction. In: Bininda-Emonds, O.R.P. (ed.) Phylogenetic Supertrees (combining information to reveal the tree of life), vol. 3, pp. 35–63. Kluwer Academic Publishers, Dordrecht (2004)
10. Semple, C., Steel, M.: A supertree method for rooted trees. Discrete Appl. Math. 105(1-3), 147–158 (2000)
11. Page, R.D.M.: Modified mincut supertrees. In: Guigó, R., Gusfield, D. (eds.) WABI 2002. LNCS, vol. 2452, pp. 537–552. Springer, Heidelberg (2002)
12. Willson, S.J.: Constructing rooted supertrees using distances. Bull. Math. Biol. 66(6), 1755–1783 (2004)
13. Ranwez, V., Berry, V., Criscuolo, A., Fabre, P.-H., Guillemot, S., Scornavacca, C., Douzery, E.J.P.: PhySIC: a veto supertree method with desirable properties. Syst. Biol. 56(5), 798–817 (2007)
14. Scornavacca, C., Berry, V., Lefort, V., Douzery, E.J.P., Ranwez, V.: PhySIC_IST: cleaning source trees to infer more informative supertrees. BMC Bioinformatics 9, 413 (2008)
15. Bininda-Emonds, O.R.P., Sanderson, M.J.: Assessment of the accuracy of matrix representation with parsimony analysis supertree construction. Syst. Biol. 50(4), 565–579 (2001)
16. Levasseur, C., Lapointe, F.-J.: Total evidence, average consensus and matrix representation with parsimony: What a difference distances make. Evol. Bioinform. 2, 249–253 (2006)
17. Aho, A.V., Sagiv, Y., Szymanski, T.G., Ullman, J.D.: Inferring a tree from lowest common ancestors with an application to the optimization of relational expressions. SIAM J. Comput. 10(3), 405–421 (1981)

18. Thorley, J.L., Wilkinson, M.: A view of supertree methods. In: Jannowitz, M.F., Lapointe, F.J., McMorris, F.R., Roberts, F.S. (eds.) Bioconsensus, vol. 61. The American Mathematical Society, Providence (2003)
19. Goloboff, P.A., Pol, D.: Semi-strict supertrees. Cladistics 18(5), 514–525 (2002)
20. Sanderson, M.J.: r8s: inferring absolute rates of molecular evolution and divergence times in the absence of a molecular clock. Bioinformatics 19(2), 301–302 (2003)
21. Rambaut, A., Grassly, N.C.: Seq-Gen: an application for the Monte Carlo simulation of DNA sequence evolution along phylogenetic trees. Comput. Appl. Biosci. 13(3), 235–238 (1997)
22. Yang, Z.: Maximum likelihood phylogenetic estimation from DNA sequences with variable rates over sites: approximate methods. J. Mol. Evol. 39(3), 306–314 (1994)
23. Higdon, J.W., Bininda-Emonds, O.P., Beck, R.M.D., Ferguson, S.H.: Phylogeny and divergence of the pinnipeds (carnivora: Mammalia) assessed using a multigene dataset. BMC Evol. Biol. 7, 216 (2007)
24. Stamatakis, A.: RAxML-VI-HPC: maximum likelihood-based phylogenetic analyses with thousands of taxa and mixed models. Bioinformatics 22(21), 2688–2690 (2006)
25. Swafford, D.: Paup*: Phylogenetic analysis using parsimony (*and other methods), Version 4 (2002)
26. Robinson, D.F., Foulds, L.R.: Comparison of phylogenetic trees. Math. Biosci. 53(1-2), 131–147 (1981)
27. Gordon, A.D.: On the assessment and comparison of classifications. In: Tomassine, R. (ed.) Analyse de Données et Informatique, Le Chesnay, INRIA, France, pp. 149–160 (1980)