Collecting reliable clades using the Greedy Strict Consensus Merger

Markus Fleischauer[1] **and Sebastian Böcker**[1]

[1]**Lehrstuhl für Bioinformatik, Friedrich-Schiller-Universität Jena, Ernst-Abbe-Platz 2, 07743 Jena, Germany**

ABSTRACT

Supertree methods combine a set of phylogenetic trees into a single supertree. Similar to supermatrix methods, these methods provide a way to reconstruct larger parts of the Tree of Life, potentially evading the computational complexity of phylogenetic inference methods such as maximum likelihood. The supertree problem can be formalized in different ways, to cope with contradictory information in the input. Many supertree methods have been developed. Some of them solve NP-hard optimization problems like the well known Matrix Representation with Parsimony, others have polynomial worst-case running time but work in a greedy fashion (FlipCut). Both can profit from a set of clades that are already known to be part of the supertree. The Superfine approach shows how the Greedy Strict Consensus Merger (GSCM) can be used as preprocessing to find these clades. We introduce different scoring functions for the GSCM, a randomization, as well as a combination thereof to improve the GSCM to find more clades. This helps, in turn, to improve the resolution of the GSCM supertree. We find this modifications to increase the number of true positive clades by 16 % while decreasing the number of false positive clades by 3 % compared to the currently used Overlap scoring.

Keywords: Supertree, Phylogeneny, Consensus

INTRODUCTION

Supertree methods are used to combine a set of phylogenetic trees with non-identical but overlapping taxon sets, into a larger supertree that contains all taxa of every input tree. Many supertree methods have been established over the years, see for example (Bininda-Emonds, 2004; Ross and Rodrigo, 2004; Chen et al., 2006; Holland et al., 2007; Scornavacca et al., 2008; Ranwez et al., 2010; Bansal et al., 2010; Snir and Rao, 2010; Swenson et al., 2012; Brinkmeyer et al., 2013; Berry et al., 2013; Gysel et al., 2013; Whidden et al., 2014); these methods complement supermatrix methods which combine rather the "raw" sequence data than the trees (von Haeseler, 2012).

Different from supermatrix methods, supertree methods allow us to analyze large datasets without constructing a multiple sequence alignment for the complete dataset, and without a phylogenetic analysis of the resulting alignment. In this context, supertree methods can be used as part of divide-and-conquer meta techniques (Huson et al., 1999a,b; Roshan et al., 2004; Nelesen et al., 2012), which break down a large phylogenetic problem into smaller subproblems that are computationally much easier to solve. The results of the subproblems are then combined using a supertree method.

Constructing a supertree is easy if no contradictory information is encoded in the input trees (Aho et al., 1981). However, resolving conflicts in a reasonable and swift way remains difficult. Matrix Representation with Parsimony (MRP) (Baum, 1992; Ragan, 1992) is still the most widely used supertree method today, as the constructed supertrees are of comparatively high quality. Since MRP is NP-hard (Foulds and Graham, 1982), heuristic search strategies have to be used. Swenson et al. (2012) introduced SuperFine which combines the Greedy Strict Consensus Merger (GSCM) (Huson et al., 1999b; Roshan et al., 2003) with MRP. The basic idea is to use a very conservative supertree method (in that case GSCM) as preprocessing for better resolving supertree methods (in that case MRP). Conservative supertree methods only resolve conflict-free clades and keep the remaining parts of the tree unresolved. We call those resolved parts of a conservative supertree *reliable clades*. Other better resolving supertree methods, such as the greedy working polynomial time FLIPCUT (Brinkmeyer et al., 2013) algorithm, may also benefit from this preprocessing.

The number of *reliable clades* returned by GSCM is highly dependent on the merging order of the source trees. Although the GSCM only returns clades that are compatible with all source trees, we find that it likewise produces clades which are not supported by any of the source trees (*random*

PeerJ PrePrints | https://dx.doi.org/10.7287/peerj.preprints.1297v3 | CC-BY 4.0 Open Access | rec: 10 Sep 2015, publ: 10 Sep 2015

clades). Obviously, random clades do not necessarily have to be part of the super tree.

With the objective to improve the GSCM as preprocessing method, we show how combining different scoring functions and randomization can be used to increase the number of reliable clades by simultaneously reducing the number of random clades. We find that it is more robust to combine a number of randomized trees using different scorings than using the same number of randomized trees using a single scoring. Compared to the currently used Overlap scoring, our method increases the number of true positive clades by 16 % and further decreases the number of false positive clades by 3 %

METHODS

Preliminaries

In this paper, we deal with graph theoretical objects called rooted (phylogenetic) trees. Let $\mathscr{V}(T)$ be the vertex set. Every leaf of a tree T is uniquely labeled and called taxon. Let $\mathscr{L}(T) \subset \mathscr{V}(T)$ be the set of all taxa in T. We call every vertex $v \in \mathscr{V}(T) \setminus \mathscr{L}(T)$ inner vertex. Every inner vertex of T induces a clade $C \subseteq \mathscr{L}(T)$. Two clades C_1 and C_2 are compatible if $C_1 \cap C_2 \in \{C_1, C_2, \emptyset\}$. Two trees are compatible if all clades are pairwise compatible. The resolution of a rooted tree is defined as $\frac{|\mathscr{V}(T)|-|\mathscr{L}(T)|}{|\mathscr{L}(T)|-1}$. Hence, a completely unresolved (star)tree has resolution 0, where a fully resolved tree has resolution 1. For a given set of trees $\mathscr{T} = \{T_1, \ldots, T_k\}$, a supertree T of $\mathscr{T}$ is simply a phylogenetic tree with leaf set $\mathscr{L}(T) = \bigcup_{T_i \in \mathscr{T}} \mathscr{L}(T_i)$. A supertree T is called consensus tree if for all input trees $T_i, T_j \in \mathscr{T} : \mathscr{L}(T_i) = \mathscr{L}(T_j)$ holds. A strict consensus of $\mathscr{T}$ is a tree that only contains clades present in all trees $T_i \in \mathscr{T}$. A semi-strict consensus of $\mathscr{T}$ contains all clades that are compatible with each clade of each $T_i \in \mathscr{T}$ (Bryant, 2003). For set of taxa $X \subset \mathscr{L}(T)$ we further define that $T_{|X}$ is the X induced subtree of T with a minimal number of edge contradictions. To create $T_{|X}$, consider the minimal subgraph $T(X)$ of T that connects elements of X and delete all inner vertices with out-degree one.

Strict Consensus Merger (SCM)

For a given pair of trees T_1 and T_2 with overlapping taxon set, the SCM (Huson et al., 1999b; Roshan et al., 2003) calculates a supertree as follows. Let $X = \mathscr{L}(T_1) \cap \mathscr{L}(T_2)$ be the set of common taxa and $T_{1|X}$ and $T_{2|X}$ the X induced subtrees. Calculate $T_X = \textsc{strictConsensus}(T_{1|X}, T_{2|X})$. Insert all subtrees, removed from T_1 and T_2 to create $T_{1|X}$ and $T_{2|X}$, into T_X without violating any of the clades in T_1 or T_2. If removed subtrees of T_1 and T_2 attach to the same edge e in T_X, a collision occurs. In that case all subtrees attaching to e will be inserted to the same point on e by creating a polytomy (see Figure 1).

Note that neither the strict consensus nor the collision handling inserts clades to the supertree T_X that conflict with any of the source trees.

Algorithm 1 Strict Consensus Merger

```
function SCM(tree $T_1$, tree $T_2$)
    $X \leftarrow \mathscr{L}(T_1) \cap \mathscr{L}(T_2)$
    if $|X| \geq 3$ then                              ▷ Otherwise, the merged tree will be unresolved.
        calculate $T_{1|X}$ and $T_{2|X}$
        $T_X \leftarrow \textsc{strictConsensus}(T_{1|X}, T_{2|X})$
        for all removed subtrees of $T_1$ and $T_2$ do
            if collision then           ▷ Subtrees of $T_1$ and $T_2$ attach to the same edge $e$ in $T_X$ (Fig. 1)
                Insert all colliding subtrees to the same point on $e$ by generating a polytomy.
            else
                Reinsert subtree into $T_X$ without violating any of the bipartition in $T_1$ or $T_2$.
            end if
        end for
        return $T_X$
    end if
end function
```

Greedy Strict Consensus Merger (GSCM)

The GSCM algorithm generalizes the SCM idea to combine a set $\mathscr{T} = \{T_1, T_2, \ldots, T_k\}$ of input trees to a supertree T with $\mathscr{L}(T) = \bigcup_{i=1}^{k} \mathscr{L}(T_i)$ by pairwise merging trees until only the supertree is

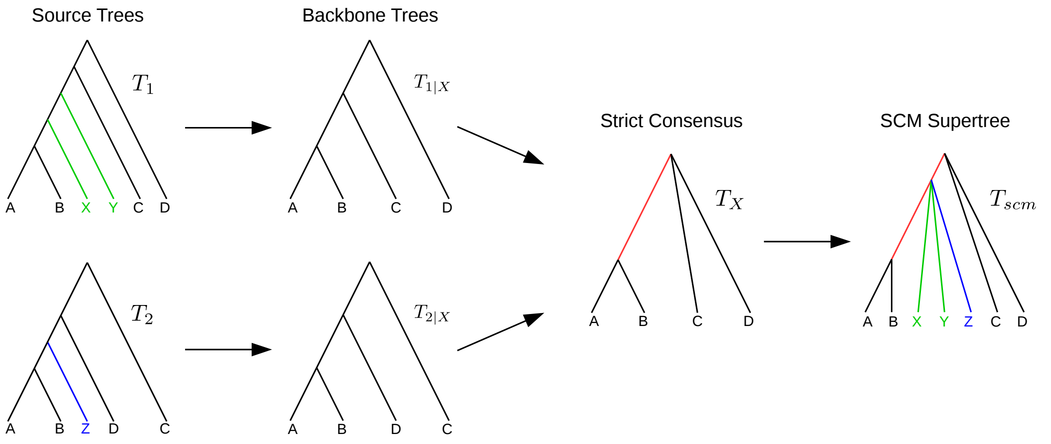


Figure 1. Exemplary SCM run including collision handling. The backbone trees $T_{1|X}$ and $T_{2|X}$ are merged using the strict consensus. The remaining subtrees of T_1 and T_2 are colored in green and blue, respectively. Both subtrees attach to the same edge in T_X (red). The green and blue subtrees are inserted into T_X by generating a polytomy (collision handling).

left. Let $score(T_i, T_j)$ be a function returning an arbitrary score of two trees T_i and T_j. The trees are selected greedily maximizing $score(T_i, T_j)$. Since the SCM does not insert clades that contradict with any of the source trees, the GSCM returns a supertree that only contains clades that are compatible with all source trees.

Algorithm 2 Greedy Strict Consensus Merger

```
function PICKOPTIMALTREEPAIR(trees $\mathscr{S} \subseteq \{T_1, T_2, \ldots, T_k\}$)
    Pick two trees $\{T_i, T_j\} \subseteq \mathscr{S}$ which maximizes $score(T_i, T_j)$
    return $T_i, T_j$
end function
function GSCM(trees $\{T_1, T_2, \ldots, T_k\}$)
    $\mathscr{S} \leftarrow \{T_1, T_2, \ldots, T_k\}$
    while $|\mathscr{S}| \geq 2$ do
        $T_i, T_j \leftarrow$ PICKOTPIMALTREEPAIR($\mathscr{S}$)
        $\mathscr{S} \leftarrow \mathscr{S} \setminus \{T_i, T_j\}$
        $T_{scm} \leftarrow$ SCM$(T_i, T_j)$
        $\mathscr{S} \leftarrow \mathscr{S} \cup \{T_{scm}\}$
    end while
    return $T_{scm}$
end function
```

Tree merging order

Although the SCM of two trees is deterministic, the order in which the GSCM selects the tree pairs for merging affects the resulting supertree. Therefore, we came up with several scoring schemes for the greedy merging. In addition, we use the original *SCM-Resolution* scoring (Roshan et al., 2003) and the *Overlap* scoring (Swenson et al., 2012). In the following, we use the five scorings that produce the best GSCM supertrees with respect to resolution and number of unique clades (in comparison to the supertrees using any of the other scorings).

SCM-Clade scoring: maximizing the number of clades in the SCM tree:

$$score(T_i, T_j) = |\mathscr{V}(\text{SCM}(T_i, T_j))| - |\mathscr{L}(\text{SCM}(T_i, T_j))|$$

Collision scoring: minimizing the number of collisions:

$$score(T_i, T_j) = -(\text{ number of edges in SCM}(T_i, T_j)\text{, where a collision occured})$$

Unique Taxa scoring: minimizing the number of unique taxa:

$$score(T_i, T_j) = -|\mathscr{L}(T_i) \Delta \mathscr{L}(T_j)|$$

PeerJ PrePrints | https://dx.doi.org/10.7287/peerj.preprints.1297v3 | CC-BY 4.0 Open Access | rec: 10 Sep 2015, publ: 10 Sep 2015

SCM-Resolution scoring (Roshan et al., 2003): maximizing the resolution of the SCM tree:

$$score(T_i,T_j) = \frac{|\mathscr{V}(\textsc{scm}(T_i,T_j))| - |\mathscr{L}(\textsc{scm}(T_i,T_j))|}{|\mathscr{L}(\textsc{scm}(T_i,T_j))| - 1}$$

Overlap scoring (Swenson et al., 2012): maximizing the number of common taxa:

$$score(T_i,T_j) = |\mathscr{L}(T_1) \cap \mathscr{L}(T_2)|$$

Combining multiple scorings

Supertrees created with the GSCM using different scorings contain different clades. To collect as much clades as possible, we compute several GSCM supertrees using different scoring functions and combine them afterwards.

The GSCM supertrees contain only clades compatible with all source trees. This might lead to the assumption that GSCM supertrees created with different scorings are also pairwise compatible. Even though the collision handling of the SCM is very conservative it causes incompatibilities between the different supertrees. There exist two types of clades in the supertrees: *reliable clades* we want to collect and *random clades* (induced by collision handling) we want to eliminate since they are not supported by any of the source trees (see Figure 2). Both, reliable clades and random clades are compatible to all clades of the source trees. However, only random clades can be incompatible among different supertrees. Hence, by removing incompatible clades from the supertrees we only eliminate random clades but none of the reliable clades.

Both, assembling reliable clades from the different GSCM supertrees and eliminating random clades can be done in one step by combining the supertrees using a semi strict consensus algorithm (Bryant, 2003). It should be noted that random clades are only eliminated if they induce a conflict between at least two supertrees (see Figure 2). Hence, there is no guarantee to eliminate all random clades.

Combined scoring:

Let *Combined4* be the combination of the SCM-Resolution, SCM-Clade, Unique Taxa and the Collision scoring.

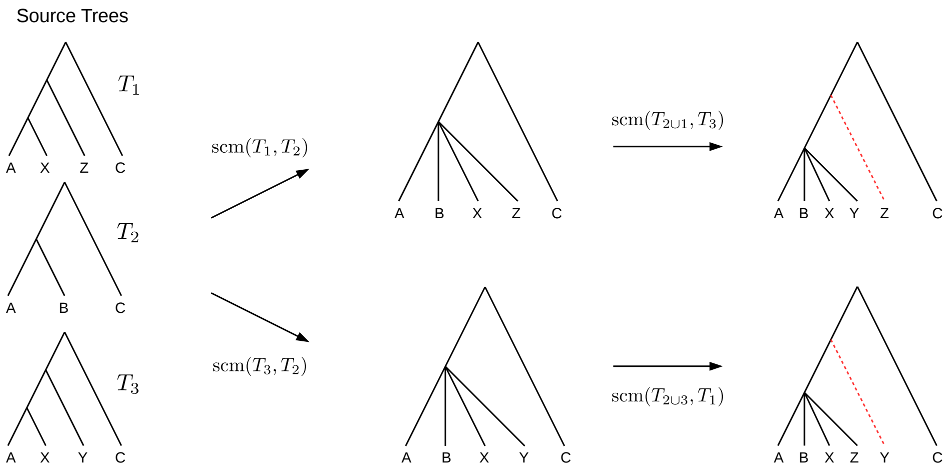


Figure 2. The collisions handling of the SCM inserts random clades into the supertree. Different order of merging the source trees causes different random clades. Random clades are not supported by any of the source trees: there is no source tree supporting the clade $(A,B,X,\mathbf{Y})|(\mathbf{Z},C)$ or supporting the clade $(A,B,X,\mathbf{Z})|(\mathbf{Y},C)$.

Randomized GSCM

Generating many different GSCM supertrees increases the probability of both, detecting all reliable clades and eliminating all random clades. To generate a larger number of GSCM supertrees, randomizing the tree merging order of the GSCM algorithm is more suitable than coming up with lots of different tree selection scorings. To this end, we replace picking an optimal pair of trees (see Algorithm 2) by picking a random pair of trees (see Algorithm 3).

Algorithm 3 Function for randomization step of the GSCM

```
function PICKRANDOMTREEPAIR(trees $\mathscr{S} \subseteq \{T_1, T_2, \dots, T_k\}$)
    Randomly pick a pair of the trees $\{T_i, T_j\} \subseteq \mathscr{S}$ with probability
        $$P(T_i,T_j) = \frac{score(T_i,T_j)}{\sum_{T_a,T_b \in \mathscr{S}, a \neq b} score(T_a,T_b)}, i \neq j$$
    return $T_i, T_j$
end function
```

Running the randomized GSCM for different scoring functions multiple times allows us to generate a large number of supertrees containing different clades. The resulting trees can be combined using a semi strict consensus as described above.

EXPERIMENTAL SETUP

To evaluate the different modifications of the GSCM algorithm we use a simulated dataset which is based on the SMIDGen protocol (Swenson et al., 2010). The SMIDGen protocol follows data collection processes used by systematists when gathering empirical data, e.g., the creation of several densely-sampled *clade-based source trees*, and a sparsely-sampled *scaffold source tree*. All source trees are rooted using an outgroup.

We generate 30 model trees with 500 taxa. For each model tree, we generate a set of 15 clade-based source trees and four scaffold source trees containing 20 %, 50 %, 75 %, or 100 % of the taxa in the model tree (the *scaffold density*). We use them as four different source tree sets: each of them containing the set of clade-based trees and one of the scaffold trees respectively. In addition, we create 30 model trees with 1000 taxa where we generate 30 clade-based source trees and again four different scaffold source trees. The results of the 1000 taxa dataset are similar to the 500 taxa dataset. Hence we exclude the 1000 taxa dataset from further evaluation. For the results of the 1000 taxa dataset we refer to the Appendix (Figures 5 and 6).

We calculate false negative rates (FNR) and false positive rates (FPR) between a supertree and the corresponding model tree. FNR is the ratio of clades that are not in the supertree but should be. FPR is the ratio of clades that are in the supertree but not in the model tree.

Unlike the Robinson Foulds distance, FNR and FPR contain information on the resolution of the supertree. The generated model trees are fully resolved. In case the supertree is fully resolved too, we get $FNR = FPR$. Otherwise, if $FNR > FPR$ the supertree is not fully resolved.

As mentioned above, we try to improve the GSCM as a preprocessing method and thus want to minimize the number of false positive clades we have to tolerate to get a true positive clade. This is reflected by $\frac{FPR}{TPR}$, where $TPR = 1 - FNR$.

The scaffold factor highly influences both, false negative and false positive rates. Thus, for every scaffold we normalize the rates to be between zero and one.

RESULTS AND DISCUSSION

We find that the scaffold factor highly influences the quality of the supertrees (see Figure 3 top). In general, all scorings profit from a large scaffold tree. In particular, for a scaffold factor of 100 % nearly all scorings perform equally good and better than for all other scaffold factors.

Having a source tree that already contains all taxa simplifies the supertree computation for the GSCM algorithm. Starting with the scaffold tree and merging the remaining source trees in arbitrary order leads to the optimal solution. No collision can occur, when the taxa set of the one trees is a subset of the taxa of the other tree. However, the SCM-Resolution scoring does not pick the scaffold tree in the first step and therefore not necessarily leads to an optimal solution. In contrast, the Overlap scoring which is working worst for all other scaffold tree sizes, produces an optimal solution for a scaffold factor of 100 %. However, the 100 % data has no practical relevance and reveals no differences between the performance of the scorings. Thus we will exclude the 100 % datasets from the further analysis.

Comparing the different scorings, we find that in general, the FNR differs more than the FPR (see Figure 3 top). For a better comparison between the scorings, we normalize the rates to be between zero and one respectively (see Figure 3 bottom). The Overlap scoring has the worst FNR and worst FPR of all individual scorings. This also leads to the by far worst $\frac{FPR}{TPR}$ (see Figure 4a).

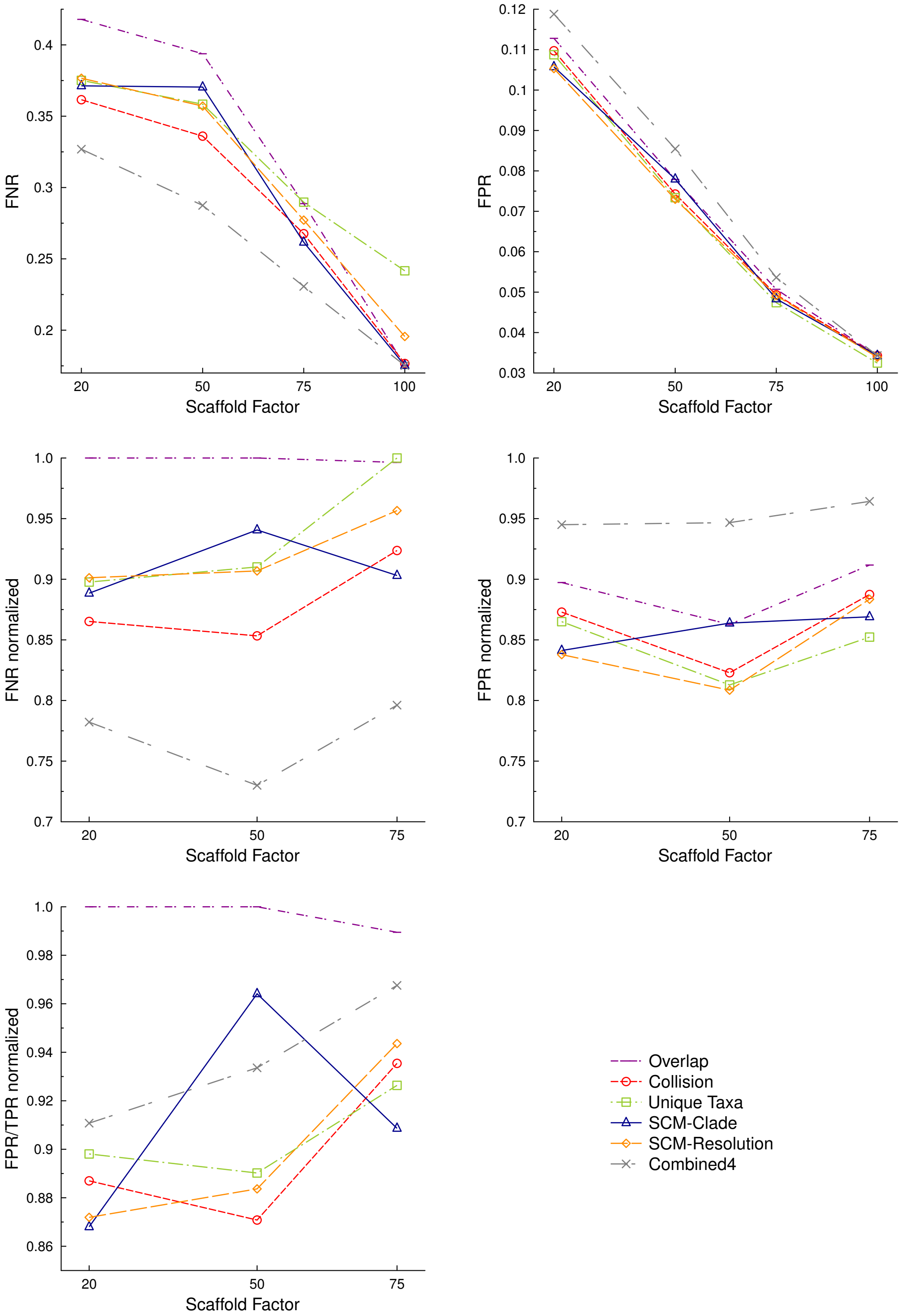
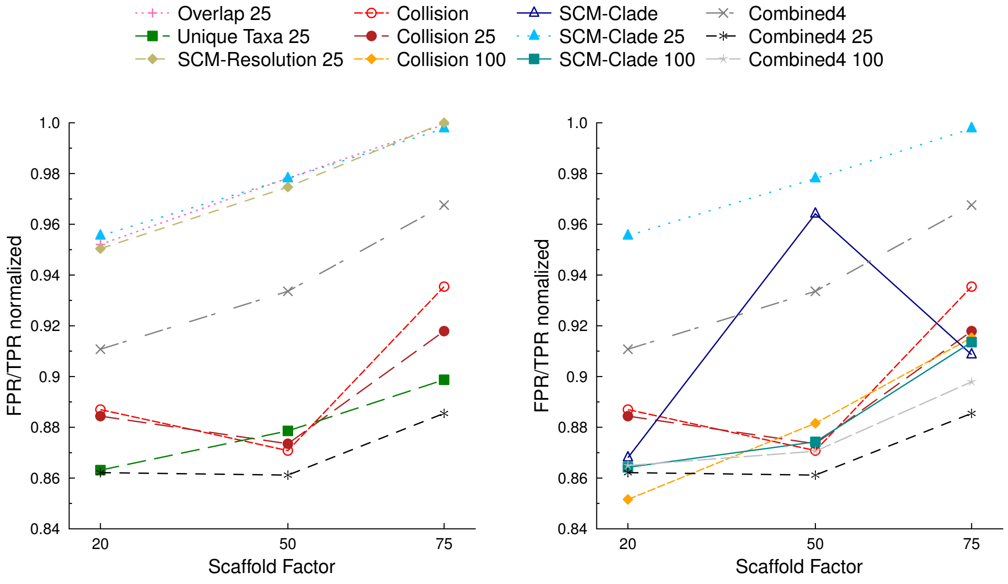


Figure 3. *FNR* and *FPR* for the different scorings on the 500 taxa dataset. The upper charts show the unmodified *FNR* (left) and *FPR* (right) for all scaffold factors. The bottom charts show only scaffold factor 20 %, 50 % and 75 % and are normalized for each scaffold factor. Combined4 is the semi strict consensus of the supertrees calculated using the Collision scoring, Unique Taxa scoring, SCM-Resolution scoring and the SCM-Clade scoring.



(a) Comparison of different scorings with (Collision 25, SCM-Clade 25, Overlap 25, Unique Taxa 25, SCM-Resolution 25, Combined4 25) and without (Collision) randomization.

(b) SCM-Clade, Collision and Combined4 scoring with different numbers of random iterations (0, 25, 100)

Figure 4. $\frac{FPR}{TPR}$ for the different scorings with and without randomization for the different scaffold factors (20%,50%,75%) on the 500 taxa dataset. The values are normalized per scaffold factor. Combined4 is the semi strict consensus of the supertrees calculated by Collision, Unique Taxa, SCM-Resolution and the SCM-Clade scoring. The integer value behind scoring names represents the number of randomized iterations

Regarding the $\frac{FPR}{TPR}$, the SCM-Clade scoring is better than all other scores for scaffold factors 20 % and 75 %. However, it is second worst for scaffold factor 50 %. The results of the remaining three individual scorings (Unique Taxa scoring, SCM-Resolution scoring, Collision scoring) highly fluctuate for different scaffold factors: for 20 % the SCM-Resolution scoring is the second best, for 50 % the Collision scoring is the best, and for 75 % the Unique Taxa scoring is the second best.

As no scoring function clearly outperforms the others, we combine the GSCM supertrees computed with the different scorings using the semi strict consensus. Since the Overlap scoring performs too bad, we only combine the remaining four scorings. This combined supertree has by far the best *FNR* for all scaffold densities, but also the worst *FPR* (see Figure 3 bottom). Thus, besides collecting all reliable clades from the four supertrees, the combination using the semi strict consensus does not eliminate enough random clades. The $\frac{FPR}{TPR}$ (see Figure 4a) of the combined supertrees is still better than for the supertrees computed based on the Overlap scoring but worse than all other individual scorings. Nevertheless, the number of true positive clades increased heavily.

To improve the *FPR* we use randomization of the tree merging order generating 25 supertrees for each scoring which are combined using the semi strict consensus (see Figure 4a). For the Unique Taxa scoring and Collision scoring, the randomization over 25 iterations improves the quality of the supertrees in comparison to the non randomized algorithm. For scaffold factor 20 % and 75 %, the Unique Taxa scoring has the best $\frac{FPR}{TPR}$. For scaffold factor 50 %, the Collision scoring with and without randomization is slightly better than the Unique Taxa scoring. The SCM-Resolution scoring and SCM-Clade scoring perform worse using randomization. The Overlap scoring slightly improves but remains worse than non-randomized scorings. We exclude the Overlap scoring from the further analysis. We now again combine the randomized supertrees of the remaining four individual scorings using the semi strict consensus. The quality of the combined supertrees clearly improves over the individual supertrees for all scaffold factors.

The scoring functions can be categorized in two groups: those who maximize the resolution of SCM tree (SCM-Resolution and SCM-Clade scoring) and those who minimize the number of collisions (Unique Taxa and Collision scoring). The scorings minimizing collisions improve using randomization while the resolution maximizing scorings get worse. The resolution maximizing scorings apparently generate a large number of collision-induced random clades.

We further tested, whether increasing the number of randomized trees to 100 solves the problem of too many random clades and thus improves quality (see Figure 4b). We compare the SCM-Clade scoring (as representative of resolution maximizing scorings) and Collision scoring (as representative of collision minimizing scorings). In general, both scorings improve with the increased number of randomized input trees. However, the combined supertrees that are generated using 25 randomized supertrees for each scoring have the best $\frac{FPR}{TPR}$. Note that 25 supertrees for four different scorings also results in 100 randomized supertrees to be combined.

Overall, we find the combined supertrees that are generated using 25 randomized supertrees for each scoring to be the best resolved supertrees. This method is also the most robust against the different scaffold densities.

In comparison to the Overlap scoring, which is currently implemented in SuperFine Swenson et al. (2012), the combination of four scorings (Collision scoring, Unique Taxa scoring, SCM-Resolution scoring, and SCM-Clade scoring) with 25 randomized supertrees per scoring increases the *TPR* by 16 % and decreases the *FPR* by 3 %.

CONCLUSION

We presented several novel scoring functions for the GSCM algorithm. We found that collisions can destroy source tree clades and introduce random clades to the supertree. Thus, the scorings that minimize the number of collisions perform best. Combining multiple GSCM supertrees using a semi strict consensus method helps to better resolve the supertree.

We find that collision-minimizing scorings work well with randomization. Combining multiple randomized supertrees increases the number of true positive clades. For resolution maximizing scorings, randomization also increases the number of true positive clades but in addition introduces more random clades to the supertrees. Thinking of a preprocessing method, those false positive clades will have a dreadful influence on the quality of the final supertree. Thus, the number of trees to combine has to be sufficiently large to increase the probability of deleting random clades.

The overall best performance is achieved with a combination of four scorings (Collision scoring, Unique Taxa scoring, SCM-Resolution scoring, and SCM-Clade scoring) with 25 randomized supertrees per scoring, resulting in 100 supertrees to be combined. This method achieves a better $\frac{FPR}{TPR}$ than calculating 100 randomized supertrees with a single scoring function. It is also more robust against different input data (scaffold factor).

REFERENCES

Aho, A. V., Sagiv, Y., Szymanski, T. G., and Ullman, J. D. (1981). Inferring a tree from lowest common ancestors with an application to the optimization of relational expressions. *SIAM J Comput*, 10(3):405–421.

Bansal, M. S., Burleigh, J. G., Eulenstein, O., and Fernández-Baca, D. (2010). Robinson-foulds supertrees. *Algorithms Mol Biol*, 5:18.

Baum, B. R. (1992). Combining trees as a way of combining data sets for phylogenetic inference, and the desirability of combining gene trees. *Taxon*, 41(1):3–10.

Berry, V., Bininda-Emonds, O. R. P., and Semple, C. (2013). Amalgamating source trees with different taxonomic levels. *Syst Biol*, 62(2):231–249.

Bininda-Emonds, O. R. P. (2004). The evolution of supertrees. *Trends Ecol Evol*, 19(6):315–322.

Brinkmeyer, M., Griebel, T., and Böcker, S. (2013). FlipCut supertrees: Towards matrix representation accuracy in polynomial time. *Algorithmica*, 67(2):142–160.

Bryant, D. (2003). A classification of consensus methods for phylogenetics. In *Bioconsensus*, pages 163–184. DIMACS: Series in Discrete Mathematics and Theoretical Computer Science.

Chen, D., Eulenstein, O., Fernández-Baca, D., and Sanderson, M. (2006). Minimum-flip supertrees: Complexity and algorithms. *IEEE/ACM Trans Comput Biology Bioinform*, 3(2):165–173.

Foulds, L. and Graham, R. L. (1982). The Steiner problem in phylogeny is NP-complete. *Adv Appl Math*, 3(1):43–49.

Gysel, R., Gusfield, D., and Stevens, K. (2013). Triangulation heuristics for maximum character compatibility. *2013 IEEE 3rd International Conference on Computational Advances in Bio and medical Sciences (ICCABS)*.

Holland, B., Conner, G., Huber, K., and Moulton, V. (2007). Imputing supertrees and supernetworks from quartets. *Syst Biol*, 56(1):57–67.

Huson, D. H., Nettles, S. M., and Warnow, T. J. (1999a). Disk-Covering, a fast-converging method for phylogenetic tree reconstruction. *J Comput Biol*, 6(3-4):369–386.

Huson, D. H., Vawter, L., and Warnow, T. J. (1999b). Solving large scale phylogenetic problems using DCM2. In *Proc. of Intelligent Systems for Molecular Biology (ISMB 1999)*, pages 118–129.

Nelesen, S., Liu, K., Wang, L.-S., Linder, C. R., and Warnow, T. (2012). Dactal: divide-and-conquer trees (almost) without alignments. *Bioinformatics*, 28(12):i274–i282.

Ragan, M. A. (1992). Phylogenetic inference based on matrix representation of trees. *Mol Phylogenet Evol*, 1(1):53–58.

Ranwez, V., Criscuolo, A., and Douzery, E. J. P. (2010). SuperTriplets: A triplet-based supertree approach to phylogenomics. *Bioinformatics*, 26(12):i115–i123.

Roshan, U., Moret, B., Warnow, T., and Williams, T. (2003). Greedy strict-consensus merger: A new method to combine multiple phylogenetic trees. Technical report, Department of Computer Science, University of Texas at Austin.

Roshan, U., Moret, B., Warnow, T., and Williams, T. (2004). Rec-I-DCM3: A fast algorithmic technique for reconstructing large phylogenetic trees. In *Proc. of IEEE Computational Systems Bioinformatics Conference (CSB 2004)*, pages 98–109.

Ross, H. and Rodrigo, A. (2004). An assessment of matrix representation with compatibility in supertree construction. In Bininda-Emonds, O. R., editor, *Phylogenetic supertrees: Combining information to reveal the Tree of Life*, volume 4 of *Computational Biology Book Series*, chapter 2, pages 35–63. Kluwer Academic.

Scornavacca, C., Berry, V., Lefort, V., Douzery, E. J. P., and Ranwez, V. (2008). PhySIC_IST: Cleaning source trees to infer more informative supertrees. *BMC Bioinformatics*, 9:413.

Snir, S. and Rao, S. (2010). Quartets MaxCut: a divide and conquer quartets algorithm. *IEEE/ACM Trans Comput Biology Bioinform*, 7(4):704–718.

Swenson, M. S., Barbancon, F., Warnow, T., and Linder, C. R. (2010). A simulation study comparing supertree and combined analysis methods using SMIDGen. *Algorithms Mol Biol*, 5(1):8.

Swenson, M. S., Suri, R., Linder, C. R., and Warnow, T. (2012). SuperFine: Fast and accurate supertree estimation. *Syst Biol*, 61(2):214–227.

von Haeseler, A. (2012). Do we still need supertrees? *BMC Biol*, 10:13.

Whidden, C., Zeh, N., and Beiko, R. G. (2014). Supertrees based on the subtree prune-and-regraft distance. *Syst Biol*, 63(4):566–581.

PeerJ PrePrints | https://dx.doi.org/10.7287/peerj.preprints.1297v3 | CC-BY 4.0 Open Access | rec: 10 Sep 2015, publ: 10 Sep 2015

APPENDIX

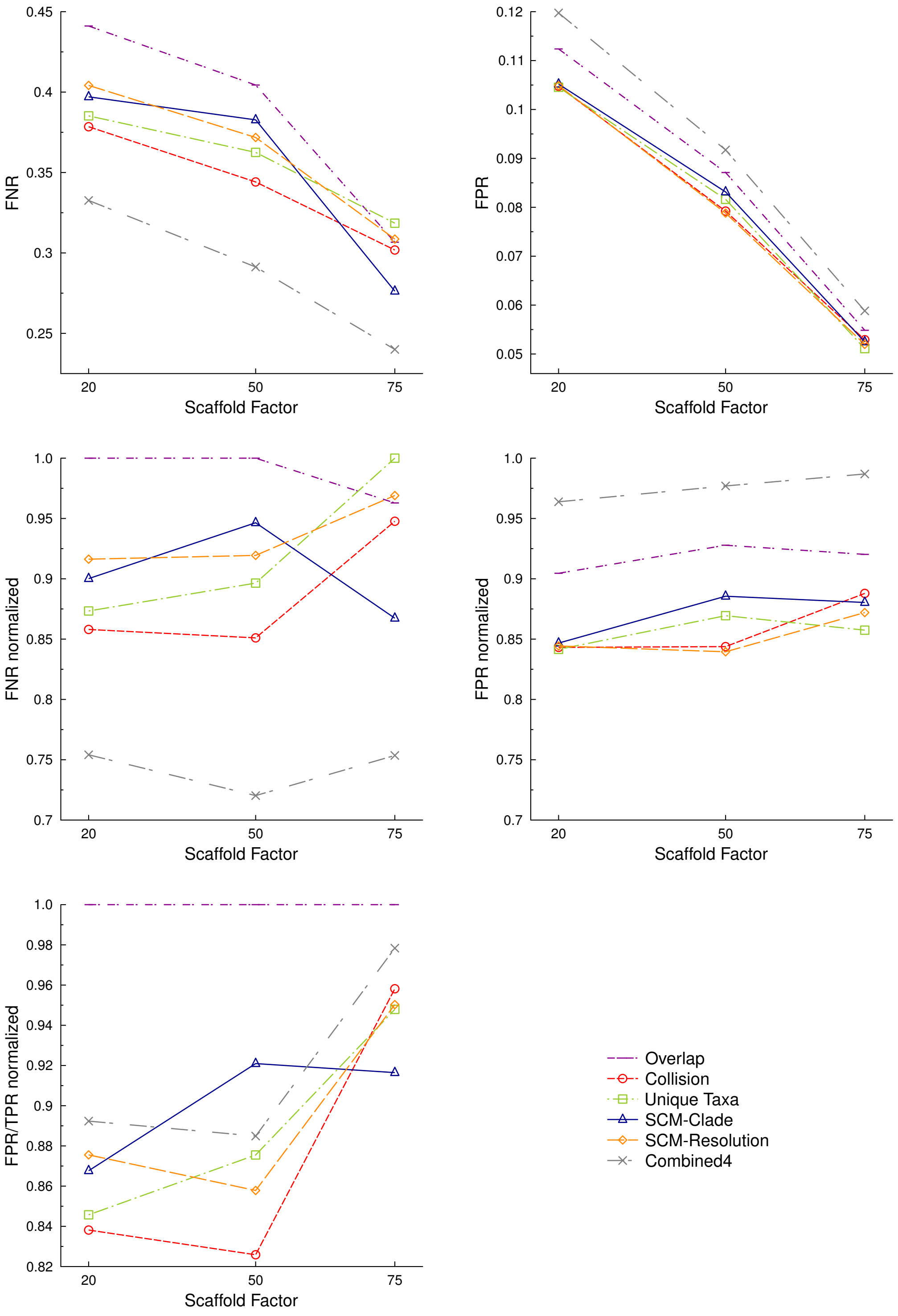
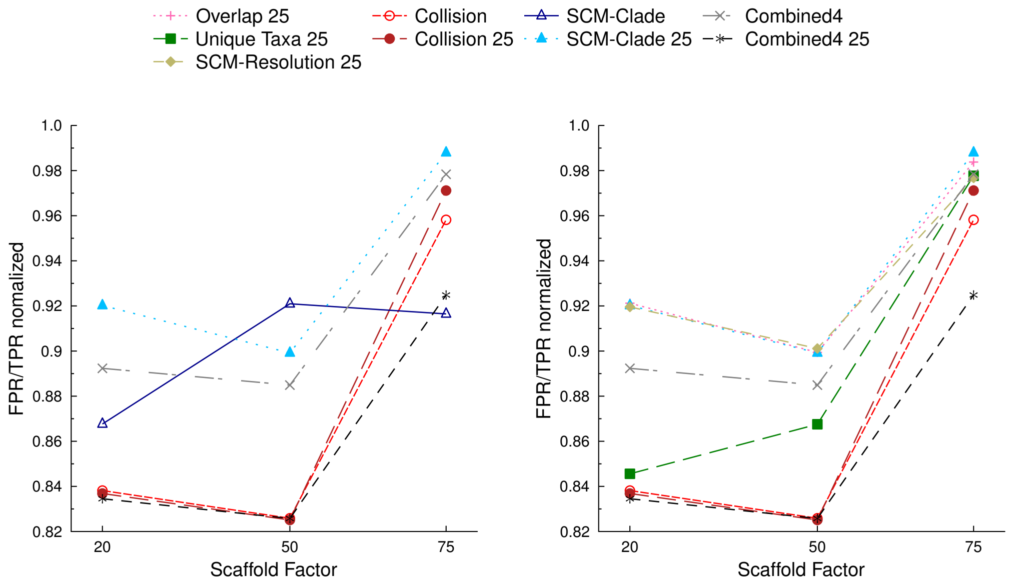


Figure 5. *FNR* and *FPR* for the different scorings on the 1000 taxa dataset. The upper charts show the unmodified *FNR* (left) and *FPR* (right) for all scaffold factors. The bottom charts show only scaffold factor 20 %, 50 % and 75 % and are normalized for each scaffold factor. Combined4 is the semi strict consensus of the supertrees calculated using the Collision scoring, Unique Taxa scoring, SCM-Resolution scoring and the SCM-Clade scoring.



(a) Comparison of different scorings with (Collision 25, SCM-Clade 25, Overlap 25, Unique Taxa 25, SCM-Resolution 25, Combined4 25) and without (Collision) randomization.

(b) SCM-Clade, Collision and Combined4 scoring with different numbers of random iterations (0 and 25)

Figure 6. $\frac{FPR}{TPR}$ for the different scorings with and without randomization for the different scaffold factors (20%,50%,75%) on the 1000 taxa dataset. The values are normalized per scaffold factor. Combined4 is the semi strict consensus of the supertrees calculated by Collision, Unique Taxa, SCM-Resolution and the SCM-Clade scoring. The integer value behind scoring names represents the number of randomized iterations