Towards a Faster and Accurate Supertree Inference

Diogo Telmo Neves[†], João Luís Sobral[‡]
Department of Informatics, University of Minho
Campus of Gualtar, Braga, Portugal
{[†]dneves, [‡]jls}@di.uminho.pt

***Abstract*—Phylogenetic inference is one of the most challenging and important problems in computational biology. However, computing evolutionary links on data sets containing only few thousands of taxa easily becomes a daunting task. Moreover, recent advances in next-generation sequencing technologies are turning this problem even much harder, either in terms of complexity or scale. Therefore, phylogenetic inference requires new algorithms and methods to handle the unprecedented growth of biological data.**

In this paper, we identify several types of parallelism that are available while refining a supertree. We also present four improvements that we made to SuperFine—a state-of-the-art supertree (meta)method—, which add support: i) to use FastTree as the inference tool; ii) to use a parallel version of FastTree, or RAxML, as the inference tool; iii) to exploit intra-polytomy parallelism within the so-called polytomy refinement phase; and iv) to exploit, at the same time, inter-polytomy and intra-polytomy parallelism within the polytomy refinement phase. Together, these improvements allow an efficient and transparent exploitation of hybrid-polytomy parallelism. Additionally, we pinpoint how future contributions should enhance the performance of such applications.

Our studies show groundbreaking results in terms of the achieved speedups, specially when using biological data sets. Moreover, we show that the new parallel strategy—which exploits the hybrid-polytomy parallelism within the polytomy refinement phase—exhibits good scalability, even in the presence of asymmetric sets of tasks. Furthermore, the achieved results show that the radical improvement in performance does not impair tree accuracy, which is a key issue in phylogenetic inferences.

I. Introduction

Phylogenetic inference (i.e., evolutionary tree estimation) is one of the most challenging and important problems in computational biology. Phylogenetic analyses are used in a daily basis and in a wide variety of fields, to name a few: in linguistics, in forensics, in cancer research and treatment, and in drug research and design [1]. Often, a multiple sequence alignment is used as the input to an estimation method that then try to solve an NP-Hard optimization problem. There are a large variety of tools to solve this kind of problem [2][3][4][5][6][7], some of those tools support several methods (e.g., Maximum Parsimony, and Maximum Likelihood). Ultimately, all those tools face the same problem: searching for an optimal tree within a tree search space that has a factorial growth (as shown in Table I). Therefore, tree estimation is a computational intensive process that requires a substantial time effort, even for moderately large data sets [8][9].

Some data sets are (already) composed by a set of smaller trees—the source trees—with overlapping sets of labelled leaves. Those smaller trees can be used to estimate a large tree—a so-called supertree—by applying a supertree method over the set of source trees. Matrix representation with parsimony (MRP) [10][11] is one of the several supertree methods that have been proposed and the most widely used to perform supertree estimation. Essentially, a supertree method combines smaller trees, which have overlapping sets of labelled leaves, into a larger tree on the full set of taxa [12].

TABLE I
Number of unrooted binary trees.

#Taxa	#Trees
n	$(2n-5)!!$
2	1
3	1
4	3
5	15
10	2027025
20	221643095476700000000

SuperFine [13] is a state-of-the-art supertree (meta)method that has three phases: i) the first, parses each source tree (hereafter referred as the Parse phase); ii) the second, estimates a supertree on the full set of taxa (hereafter referred as the SCM phase); and iii) the third, refines the estimated supertree (hereafter referred as the Refinement phase). Figure 1 depicts the workflow of SuperFine. The SCM phase is, essentially, an agglomerative clustering that amalgamates the source trees by applying iteratively the Strict Consensus Merger (SCM) algorithm [14]. The Refinement phase is, usually, the most computational intensive among the three phases of SuperFine [15] and its goal is to refine each polytomy[1], if possible, that the estimated supertree has.

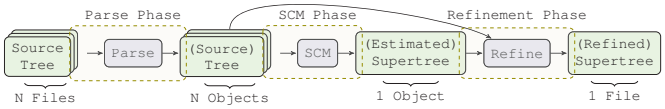

Fig. 1. SuperFine's workflow.

A parallelization of the Refinement phase is described in [15]. That parallelization strategy performs parallel calls to an inference tool (in the case, PAUP* [2]), one call per each polytomy present in the estimated supertree. We call **inter-polytomy parallelism** to this kind of parallelism. Despite its success, the performance improvement shown in [15] was limited when using biological data sets (i.e., real-world data),

[1]A polytomy is an internal node which degree—given by the number of edges that the node has—is greater than 3.

978-1-4673-7194-0/15/$31.00 ©2015 IEEE

in opposition to the excellent results achieved with simulated data sets. This is not surprising since it is known that simulated data sets have very different characteristics than those of biological data sets [16]. So, finding within the tree search space a tree that better explains the real-world data is harder and, thus, requires a much greater computational effort.

Improving, even further, the performance of the sequential and parallel versions of SuperFine is extremely valuable to domains where supertree estimation is required, specially when using real-world data. For instance, parallelism available within the refinement of each polytomy was never exploited, we call **intra-polytomy parallelism** to this kind of parallelism. Neither it was the combination of exploiting inter-polytomy parallelism and intra-polytomy parallelism, we call **hybrid-polytomy parallelism** to this kind of parallelism.

In this paper, we present four improvements that we made to SuperFine, which add support: i) to use FastTree [4] as the inference tool; ii) to use a parallel version of FastTree, or RAxML [6], as the inference tool; iii) to exploit intra-polytomy parallelism within the Refinement phase; and iv) to exploit at the same time inter-polytomy and intra-polytomy parallelism within the Refinement phase. Together, these improvements allow an efficient and transparent exploitation of hybrid-polytomy parallelism. Additionally, we pinpoint how future contributions should enhance the performance of such applications.

Our studies show groundbreaking results in terms of the achieved speedups, specially when using biological data sets. Moreover, we show that the new parallel strategy—which exploits the hybrid-polytomy parallelism within the Refinement phase—exhibits good scalability, even in the presence of asymmetric sets of tasks. Furthermore, the achieved results show that the radical improvement in performance does not impair tree accuracy, which is a key issue in phylogeny inference.

In our studies we used PAUP* 4.0b10, RAxML 8.0.22, FastTree 2.1.7, and sequential and parallel implementations of SuperFine. We used the 1000-taxon simulated data set, studied originally in [13], and several biological data sets. The 1000-taxon simulated data set is composed by clade-based source trees and scaffold source trees, and has four scaffold densities (20%, 50%, 75%, and 100%). Each clade-based source tree is a dense sample within a specific clade of the model tree. Each scaffold source tree is a random sampling of a proportion of the taxa throughout the model tree. The biological data sets used were:

- CPL (Comprehensive Papilionoid Legumes), 2228 taxa, 39 source trees, studied originally in [17];
- Marsupials, 267 taxa, 158 source trees, studied originally in [18];
- Placental Mammals, 116 taxa, 726 source trees, studied originally in [19];
- Seabirds, 121 taxa, 7 source trees, studied originally in [20]; and
- THPL (Temperate Herbaceous Papilionoid Legumes), 558 taxa, 19 source trees, studied originally in [21].

II. SuperFine Overview

Usually, the estimated supertree (see Figure 1) is not fully resolved, which means it has polytomies. In [13], Swenson *et al.* have presented SuperFine and provided a detailed explanation on how to improve the quality of the estimated supertree by refining each polytomy. Refining a polytomy implies performing an inference operation over a matrix that represents that polytomy (see inference phase in Figure 2). Those inference operations—one per each polytomy—dominate the running time of SuperFine's Refinement phase, being negligible the time spent in the remaining operations of the Refinement phase.

As aforementioned, the Refinement phase is, usually, the most computationally intensive among the three phases of SuperFine (see Figure 1). However, we can, now, be more precise and pinpoint the inference operation as the most computational intensive operation of SuperFine. Thus, the performance of the sequential version of SuperFine can be improved if one use a faster inference tool. Another possibility is enhancing the parallelization of SuperFine. In [15], the cost of the Refinement phase was reduced by exploiting (only) inter-polytomy parallelism. However, as we will show ahead, the exploitation of available parallelism within an inference operation—intra-polytomy parallelism exploitation—may radically contribute to reduce the running time spent in SuperFine's Refinement phase, and this was never explored before. In the same way, hybrid-polytomy parallelism was also never explored before.

In [13] and [15] PAUP* was used as the inference tool. PAUP* is an excellent phylogeny software package that has been widely accepted and used in countless phylogenetic studies. However, PAUP* does not provide support for multithreading. Nowadays, in the so-called multicore era, the lack of multithreading support is a major drawback to parallelism exploitation. Thus, we decide to explore RAxML and FastTree as inference tools. The former is a widely used phylogeny software package that has many options. The latter has fewer options than RAxML but can be used to establish fair comparisons with RAxML, to the extent of SuperFine's requirements. Above all, each of these tools—RAxML and FastTree—provide support for multithreading and its source code is freely available. Nevertheless, FastTree has one critical limitation: its parallelization strategy is bounded by three OpenMP [22] parallel sections, which are used while doing nearest neighbor interchange (NNI) moves to improve the maximum likelihood of a tree [4].

III. Improving Sequential SuperFine

As mentioned earlier in Section II, the inference operation is the most computational intensive operation of SuperFine. Thus, using a faster inference tool may yield a significant performance improvement on the sequential version of SuperFine. This fact has been studied in [23]. However, the setup SuperFine+FastTree has never been explored before, but the setup SuperFine+RAxML was explored in [23]. So, we add a new extension to SuperFine to provide support for using FastTree as the inference tool.

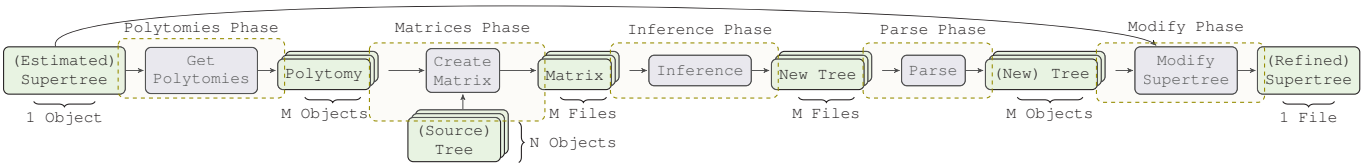

Fig. 2. Workflow of SuperFine's Refinement phase (a detailed version of Refinement phase shown in Figure 1).

A. Calibration

Among many other options that were also tested, RAxML and FastTree provide support to perform phylogenetic analyses under the General Time Reversible (GTR) model of nucleotide substitution under the Gamma model of rate heterogeneity. To the best of our knowledge, RAxML uses the Gamma4 model while FastTree uses the Gamma20 model. Moreover, these tools also provide support to perform phylogenetic analyses under the GTR model of nucleotide substitution under the CAT approximation. From RAxML's manual and from [24], under certain circumstances, the approximation GTRCAT is adequate to perform "phylogenetic analyses at a significantly lower computational cost (about 4 times faster) and memory consumption (4 times lower)". The `-f E`[2] option allows RAxML to have a similar logic as the FastTree program. These insights were useful to decide which commands one should use to conduct fair comparisons, those are shown in Table II.

TABLE II
COMMANDS TO CALL RAXML AND FASTTREE SEQUENTIAL VERSIONS.

Tool	Command
RAxML	`raxmlHPC-AVX -f E -p 7 -m GTRGAMMA -s data -n tree`
	`raxmlHPC-AVX -f E -p 7 -m GTRCAT -s data -n tree`
FastTree	`FastTree -gtr -gamma -nt -out tree data`
	`FastTree -gtr -cat 25 -nt -out tree data`

IV. IMPROVING PARALLEL SUPERFINE

For each biological dataset, we started by getting the estimated tree (see Figure 1). Table III shows an overview of the polytomies present in each of those trees. Then, for each estimated tree, we obtained the set of files that represent the polytomies (i.e., the output of matrices phase shown in Figure 2).

TABLE III
POLYTOMIES OVERVIEW PER ESTIMATED SUPERTREE.

		CPL	Marsupials	Pla. Mam.	Seabirds	THPL
# Polytomies		105	18	1	10	36
Degree	**Minimum**	3	3	114	4	3
	Maximum	531	199	114	12	94
	Sum	1287	273	114	71	312
	Median	4	4	114	6-7	4
	Mean	12.3	15.2	114.0	7.1	8.7

A. Calibration

The insights that we used to establish which commands should be used with the sequential version were also useful to let us decide which commands should be used with the parallel version of SuperFine. Those commands are shown in Table IV.

TABLE IV
COMMANDS TO CALL RAXML AND FASTTREE PARALLEL VERSIONS.

Tool	Command
RAxML	`raxmlHPC-PTHREADS-AVX -f E -p 7 -T` *#THREADS* `-m GTRGAMMA -s data -n tree`
	`raxmlHPC-PTHREADS-AVX -f E -p 7 -T` *#THREADS* `-m GTRCAT -s data -n tree`
FastTree	`export OMP_NUM_THREADS=`*#THREADS* `; FastTreeMP -gtr -gamma -nt -out tree data`
	`export OMP_NUM_THREADS=`*#THREADS* `; FastTreeMP -gtr -cat 25 -nt -out tree data`

B. Inter-Polytomy and Intra-Polytomy Parallelism

Inter-polytomy parallelism can be exploited when there is more than one polytomy present in the estimated supertree (see first row of Table III, being the exception the Placental Mammals data set which has a single polytomy). However, the decision to exploit intra-polytomy parallelism is not that simple. Essentially, the exploitation of intra-polytomy parallelism should be reserved to those polytomies that have a higher degree, which, typically, should be much higher than the degree of the majority of the remaining polytomies. Examples of such polytomies are shown in the third row of Table III, being the exception the Seabirds data set (whose polytomies have smaller degrees and, thus, are easier to refine, including the largest polytomy). A detailed study about the (negative) impact that higher degree polytomies have on the performance of the Refinement phase is provided in [15]. One of the main conclusions of that study is that higher degree polytomies are much harder to refine than lower degree polytomies. Therefore, higher degree polytomies are the perfect spot where intra-polytomy parallelism should be exploited.

C. Hybrid-Polytomy Parallelism

To exploit hybrid-polytomy parallelism, it is required to quantify the amount of threads to be used when exploiting intra-polytomy parallelism. This is a key decision since the exploitation of intra-polytomy parallelism affects the way that inter-polytomy parallelism get exploited, and vice versa. Therefore, it becomes fundamental to establish a metric that enables to set the weight of each polytomy, which then can be used to determine the amount of threads to be used while running the inference operation (i.e., while exploiting intra-polytomy parallelism). Thus, for each polytomy, the amount of threads is given by the polytomy's weight times the total number of cores used. Depending on the inference tool to be

[2]From RAxML manual: "-f E: This option will execute a very fast tree search algorithm that will not try as hard to optimize the likelihood. It is intended for very large trees and follows a similar logic as the FastTree program.".

used, this number—the amount of threads used to refine a polytomy—may be bounded (for instance, the parallelization of FastTree is limited to 3 threads).

In this study, we used the time complexity provided by FastTree, which is $O(N^{1.5}log(N)La)$ time, where N is the number of unique sequences (i.e., the number of taxa), L is the width of the alignment, and a is the size of the alphabet (in practice, for each of the aforementioned matrices N is the number of rows and L the number of columns of that matrix, and a is the amount of different symbols of the same matrix). Unfortunately, RAxML does not provide such time complexity analysis, and its manual provide "only" the following rule of thumb: "As a rule of thumb I'd use one core/thread per 500 DNA site patterns...". The best that one can take from that information is that RAxML seems tailored to exploit parallelism on very large alignments. Since each RAxML command use the `-f E2` option (see Table IV) that enables RAxML to follow a similar logic to that of FastTree program, we used the same time complexity for RAxML (i.e., the time complexity of FastTree), though such metric is not accurate for RAxML.

D. Hybrid Parallelization

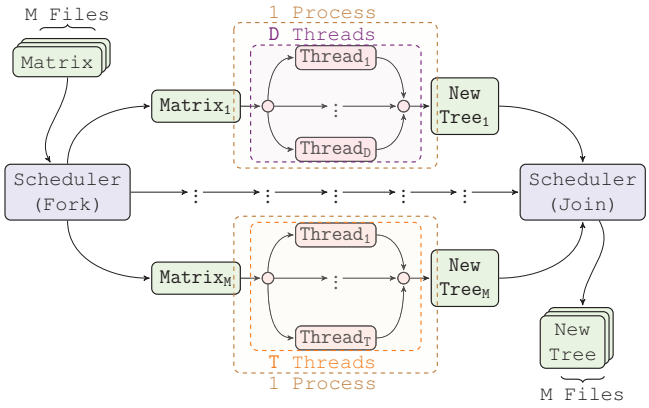

Fig. 3. Workflow to exploit hybrid-polytomy parallelism (a detailed version of inference phase shown in Figure 2).

Figure 3 depicts an abstract overview of our approach to exploit hybrid-polytomy parallelism in this kind of applications (deeper insights are given in [25]). The scheduler starts by applying a metric, accordingly to the inference tool to be used, to determine the amount of threads to be used while refining one element—matrix file—of its worklist. The worklist is sorted to ensure that larger tasks get executed in first place. Then, per each element of its worklist, the scheduler chooses the proper command accordingly to the selected inference tool, the model of nucleotide substitution, the model of rate heterogeneity or CAT approximation (see Table IV) and sets, if necessary, the number of threads. After, the scheduler will launch one process per each element of its worklist, maintaining each available core busy. If all cores are occupied, the scheduler waits for the completion of a process. While there is work to be done, when one or more cores are available the scheduler launch always the process that requires more threads, occupying the necessary cores.

V. Results

A. Experimental Design

We used in our evaluations one computing node at Stampede [26] supercomputer. A Stampede's computing node has two eight-core Xeon E5-2680 (2.27 GHz) processors, is configured with 32GB of memory, and runs CentOS release 6.5 (Final). RAxML and FastTree were compiled using gcc 4.7.1 with -O3 optimization flag. RAxML was compiled with support for AVX, and with support for AVX and Pthreads. FastTree was compiled with support for SSE3, and with support for SSE3 and OpenMP. We used Python 2.7.3 EPD 7.3-2 (64-bit) to run SuperFine. Finally, we took the average running time of six runs for each program/thread-count/data set combination. By program, we mean a setup that uses SuperFine (including all possible variations: Seq(uential); Intra, exploits intra-polytomy parallelism; Inter, exploits inter-polytomy parallelism, and Hybrid, exploits hybrid-polytomy parallelism) and an inference tool (including the possible variations: GAMMA model of rate heterogeneity; or CAT approximation). As an example, the Hybrid_SuperFine+RAxML(GTR + CAT) program is characterized by a setup that uses the hybrid parallel version of SuperFine which in turns uses RAxML under the GTR model using the CAT approximation. It is important to notice that the SuperFine+PAUP* program is the baseline implementation (described in [13]), and the Inter_SuperFine+PAUP* program is the parallel implementation of SuperFine described in [15].

B. Tree Accuracy

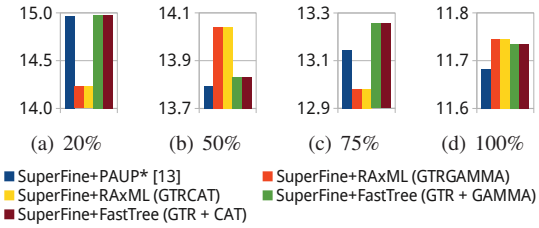

Fig. 4. Average topological accuracy, given by Robinson-Foulds (RF) error rates (%), of the inferred trees compared with the model trees of the 1000-taxon simulated data set, accordingly to each scaffold factor (20—100%).

With the 1000-taxon simulated data set, we examined topological accuracy using false-positive (FP), false-negative (FN), and Robinson-Foulds (RF) [27] error rates of the inferred trees compared with the model trees. Figure 4 shows the RF error rates in percentage. The scaffold factor is the proportion of taxa from the model tree that is sampled in the scaffold tree, known as the scaffold density (for further details see Section I and [13]). As it is possible to observe, the RF error rates are roughly the same no matter the inference tool, or the model used. Moreover, the RF error rates are roughly the same no matter the version—Sequetial, Intra-Parallel, Inter-Parallel, or Hybrid-Parallel—of the program used (we decided to elide those results due to space limitations). However, it is important to mention that while PAUP* and FastTree exhibit the same level of FP and FN error rates that is not the case of RAxML. RAxML exhibits very low FP error rates—less than 6%—but

relatively high FN error rates—between 19% and 23%—(we decided to elide those results due to space limitations). The FP, FN, and RF error rates have the same level of the counterpart error rates reported in [13][15], when using the same data set.

C. Performance of Sequential SuperFine

We decided to show only results when the CAT approximation was used by RAxML and FastTree since the results when using the GAMMA model are barely the same. This decision was also based on the calibration made (see Section III-A) and on tree accuracy results (see Section V-B). Figure 5 shows the running times (in seconds) of three sequential programs, being the Seq_SuperFine+PAUP* program the baseline implementation [13].

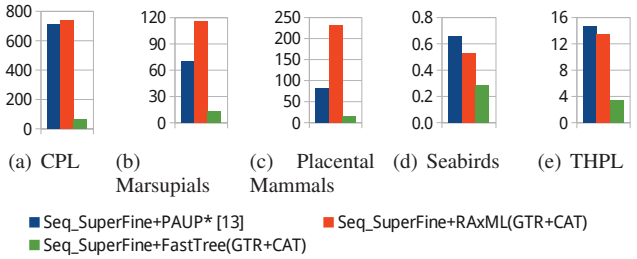

Fig. 5. Running times (in seconds) of three sequential programs, the Seq_SuperFine+PAUP* program is the baseline implementation [13].

The first point to notice is that our new version of sequential SuperFine—the Seq_SuperFine+FastTree program—outperforms, by far, the other sequential versions. The second point to notice is that the Seq_SuperFine+FastTree program exhibits good scalability since its performance does not get affected with vary problem sizes (see Table III). The third point to notice is that the Seq_SuperFine+RAxML program takes longer than the baseline implementation—the Seq_SuperFine+PAUP* program—to complete on the CPL, the Marsupials, and the Placental Mammals data sets. It is also important to notice that the performance of the Seq_SuperFine+FastTree program outperforms any setup that was used in [23], for the same data sets.

D. Performance of Parallel SuperFine

We decided to show speedups instead of running times since speedups show in a clear way the improvements in performance that it is possible to achieve when using the hybrid parallelization and FastTree. We decided also to show only results when the CAT approximation was used by RAxML and FastTree since the results when using the GAMMA model are barely the same. This decision was also based on the calibration made (see Section IV-A) and on tree accuracy results (see Section V-B). Moreover, we also decided to not show results of any program that exploits only intra-polytomy parallelism since, as expected, those results are better than the ones of the counterpart sequential version, but are worse than the ones obtained when exploiting inter-polytomy or hybrid-polytomy parallelism. Figure 6 shows the achieved speedups of several parallel programs relatively to the performance of the Seq_SuperFine+PAUP* program [13] (i.e., the baseline implementation).

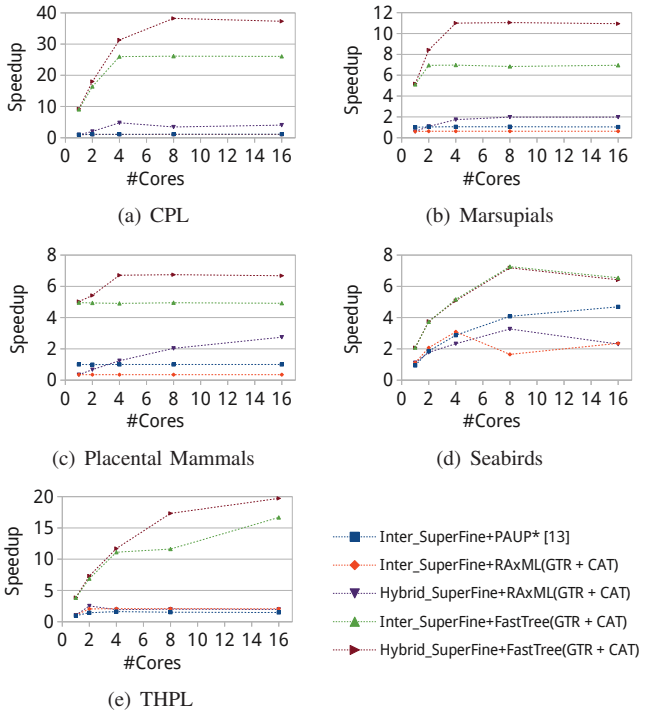

Fig. 6. Speedups of several programs relatively to Seq_SuperFine+PAUP* program [13].

The first point to notice is that the program that combines SuperFine with FastTree and exploits hybrid-polytomy parallelism—Hybrid_SuperFine+FastTree—outperforms any other, no matter the data set. The second point to notice is that the Hybrid_SuperFine+FastTree program exhibits a good scalability, it is important to recall that the data sets used in this study have very different characteristics (see Section I and Table III). On the contrary, programs using PAUP* or RAxML do not scale well, or not scale at all. The third point to notice, and probably the most important, are the magnitude of achieved speedups when using the Hybrid_SuperFine+FastTree program. As an example, on the CPL data set the Hybrid_SuperFine+FastTree program is roughly 38X faster than the baseline implementation [13] when using 8 cores, and more than 35X faster than the Inter_SuperFine+PAUP* program (i.e., the parallel implementation described in [15]). In other words, on the CPL data set the hybrid parallelization enables to go from more than 700 seconds to less than 20 seconds.

The results of the versions that use FastTree, despite being excellent, are restricted due to the intrinsic limitation of FastTree parallelization (three OpenMP parallel sections). This limitation is evident on the CPL, the Marsupials, and the Placental Mammals data sets. On the Seabirds data set, when moving from 8 to 16 cores, the downgrade in performance is due to the size of the data set, there are fewer polytomies—10 (see Table III)—than cores—16. Nevertheless, these are results that were never achieved before and corroborate that the hybrid parallelization represents a step forward towards a faster and accurate supertree inference.

Finally, the results of the programs that use RAxML were

somehow disappointing. Despite some data sets used in this study being relatively large, their polytomies do not represent extremely large alignments and, as aforementioned, RAxML seems tailored to exploit parallelism on that kind of alignments.

VI. Conclusions and Future Work

In this paper, we have identified several types of parallelism that are available while refining a supertree. We also have presented several improvements that we made to a state-of-the-art supertree (meta)method—SuperFine. As shown, those improvements yield significant speedups, both on the sequential and parallel versions of SuperFine. Additionally, we gave several useful insights about this kind of applications, mainly on how and when to exploit inter-polytomy and intra-polytomy parallelism, which together can be used to efficiently exploit hybrid-polytomy parallelism. We have shown that the hybrid-parallel strategy allows to achieve groundbreaking results, even when using real-world data (i.e., biological data sets). We also have shown that it is possible to achieve a radical performance improvement without sacrificing tree accuracy. Moreover, the performance results shown in this paper exceed by far the counterpart results shown in [15] and in [23].

We still have shown that the parallelization strategy of FastTree should be redesigned, otherwise with some data sets it would not be possible to harness the performance potential of nowadays parallel platforms.

The parallelization of SuperFine should be extended to the SCM phase. However, this should be extremely difficult, if not impossible, with the current implementation of the SCM phase of SuperFine, since the SCM algorithm is an agglomerative clustering that imposes order on how each pair of source trees get amalgamated. Most likely, a new algorithm to amalgamate source trees would be required.

The use of computational tools to help scientists in their research is a reality in science nowadays. Thus, we plan to turn our implementations publicly available, as soon as possible. We are certain that this work is valuable to others, such as computational biologists, since it allows to accelerate time consuming analyses, without impairing tree accuracy.

We are currently developing support for MPI to enable the use of distributed memory systems. Thus, the levels of supported parallelism will increase and it will be possible to cope with the growth of data sets. We are also planning to add support for other features, such as checkpointing.

VII. Acknowledgments

This research was partially supported by Fundação para a Ciência e a Tecnologia (grant SFRH/BD/42634/2007).

We thank Rui Gonçalves, Rui Silva, and Tandy Warnow for fruitful discussions and valuable feedback. We thank Keshav Pingali for his valuable support and sponsorship to let us execute jobs on TACC machines. We are deeply grateful to Rui Oliveira, without whom it would not be possible to present this work. We are very grateful to the anonymous reviewers for the evaluation of our paper and for the constructive critics.

References

[1] Z. Yang and B. Rannala, "Molecular phylogenetics: principles and practice," *Nature Reviews Genetics*, vol. 13, no. 5, pp. 303–314, 2012.

[2] D. L. Swofford, "PAUP*: phylogenetic analysis using parsimony, version 4.0b10," 2011.

[3] J. Felsenstein, *PHYLIP (Phylogeny Inference Package) version 3.6a3*. Distributed by the author. Department of Genome Sciences, University of Washington, Seattle, 2002.

[4] M. N. Price, P. S. Dehal, and A. P. Arkin, "FastTree 2 Approximately Maximum-Likelihood Trees for Large Alignments," *PloS One*, 2010.

[5] S. Guindon, J. Dufayard, V. Lefort, M. Anisimova, W. Hordijk, and O. Gascuel, "New algorithms and methods to estimate maximum-likelihood phylogenies: assessing the performance of PhyML 3.0." *Syst. Biol.*, 2010.

[6] A. Stamatakis, "Raxml version 8: A tool for phylogenetic analysis and post-analysis of large phylogenies," *Bioinformatics*, 2014.

[7] J. P. Huelsenbeck and F. Ronquist, "MRBAYES: Bayesian inference of phylogenetic trees," *Bioinformatics*, 2001.

[8] K. Liu, T. J. Warnow, M. T. Holder, S. Nelesen, J. Yu, A. Stamatakis, and C. R. Linder, "SATé-II: Very Fast and Accurate Simultaneous Estimation of Multiple Sequence Alignments and Phylogenetic Trees," *Syst. Biol.*, 2011.

[9] K. Liu, C. Linder, R. Suri, and T. Warnow, "Multiple sequence alignment: a major challenge to large-scale phylogenetics," *PLoS Currents: Tree of Life*, 2010.

[10] B. R. Baum, "Combining trees as a way of combining data sets for phylogenetic inference, and the desirability of combining gene trees," *Taxon*, 1992.

[11] M. A. Ragan, "Phylogenetic inference based on matrix representation of trees," *Molecular Phylogenetics and Evolution*, vol. 1, pp. 53–58, 1992.

[12] A. D. Gordon, "Consensus supertrees: The synthesis of rooted trees containing overlapping sets of labeled leaves," *Journal of Class.*, 1986.

[13] M. S. Swenson, R. Suri, C. R. Linder, and T. Warnow, "Superfine: Fast and accurate supertree estimation," *Systematic Biology*, 2011.

[14] U. Roshan, B. Moret, T. Williams, and T. Warnow, "Performance of Supertree Methods on Various Data Set Decompositions," in *Phylogenetic Supertrees*, 2004.

[15] D. T. Neves, T. Warnow, J. L. Sobral, and K. Pingali, "Parallelizing SuperFine," in *Proceedings of the 27th Annual ACM Symposium on Applied Computing, SAC '12*. ACM, 2012, pp. 1361–1367.

[16] A. Stamatakis, "An efficient program for phylogenetic inference using simulated annealing," in *Par. and Dist. Proc. Symposium, IPDPS*, 2005.

[17] M. McMahon and M. Sanderson, "Phylogenetic supermatrix analysis of GenBank sequences from 2228 papilionoid legumes," *Syst. Biol.*, vol. 55, no. 5, pp. 818–836, 2006.

[18] M. Cardillo, O. R. P. Bininda-Emonds, E. Boakes, and A. Purvis, "A species-level phylogenetic supertree of marsupials," *J. Zool.*, vol. 264, pp. 11–31, 2004.

[19] O. R. P. Bininda-Emonds, M. Cardillo, K. E. Jones, R. D. E. MacPhee, R. M. D. Beck, R. Grenyer, S. A. Price, R. A. Vos, J. L. Gittleman, and A. Purvis, "The delayed rise of present-day mammals," *Nature*, vol. 446, pp. 507–512, 2007.

[20] M. Kennedy and R. Page, "Seabird supertrees: combining partial estimates of procellariiform phylogeny," *The Auk*, 2002.

[21] M. Wojciechowski, M. Sanderson, K. Steele, and A. Liston, "Molecular phylogeny of the "temperate herbaceous tribes" of papilionoid legumes: a supertree approach," *Adv. Legume Syst.*, vol. 9, pp. 277–298, 2000.

[22] L. Dagum and R. Menon, "Openmp: an industry standard api for shared-memory programming," *Computational Science & Engineering, IEEE*, vol. 5, no. 1, pp. 46–55, 1998.

[23] N. Nguyen, S. Mirarab, and T. Warnow, "MRL and SuperFine+ MRL: new supertree methods." *Algorithms for Molecular Biology*, 2012.

[24] A. Stamatakis, "Phylogenetic models of rate heterogeneity: a high performance computing perspective," in *Par. and Dist. Proc. Symposium, IPDPS*, 2006.

[25] D. T. Neves, "Multilevel Task Parallelism Exploitation on Asymmetric Sets of Tasks and When Using Third-Party Tools," in *Proceedings of the 14th International Symposium on Parallel and Distributed Computing, ISPDC '15*. IEEE, 2015.

[26] Texas Advanced Computing Center (TACC), The University of Texas at Austin. [Online]. Available: http://www.tacc.utexas.edu/

[27] D. Robinson and L. Foulds, "Comparison of phylogenetic trees," *Mathematical Biosciences*, 1981.