JOURNAL OF COMPUTATIONAL BIOLOGY
Volume 22, Number 5, 2015
© Mary Ann Liebert, Inc.
Pp. 377–386
DOI: 10.1089/cmb.2014.0156

PASTA: Ultra-Large Multiple Sequence Alignment for Nucleotide and Amino-Acid Sequences

SIAVASH MIRARAB,[1] NAM NGUYEN,[1] SHENG GUO,[2] LI-SAN WANG,[3]
JUNHYONG KIM,[3] and TANDY WARNOW[1,4]

ABSTRACT

We introduce PASTA, a new multiple sequence alignment algorithm. PASTA uses a new technique to produce an alignment given a guide tree that enables it to be both highly scalable and very accurate. We present a study on biological and simulated data with up to 200,000 sequences, showing that PASTA produces highly accurate alignments, improving on the accuracy and scalability of the leading alignment methods (including SATé). We also show that trees estimated on PASTA alignments are highly accurate—slightly better than SATé trees, but with substantial improvements relative to other methods. Finally, PASTA is faster than SATé, highly parallelizable, and requires relatively little memory.

Key words: algorithms, metagenomics, molecular evolution, multiple alignment, phylogenetic trees.

1. INTRODUCTION AND MOTIVATION

Multiple sequence alignment (MSA) is a basic step in many bioinformatics analyses, including predicting the structure and function of RNAs and proteins and estimating phylogenies. Performance studies have shown that some MSA methods can produce highly accurate alignments for large slowly evolving datasets (e.g., Sievers et al., (2013). However, studies focusing on phylogeny estimation with up to 28,000 sequences have shown that only SATé-I (Liu et al., 2009) and SATé-II (Liu et al., 2011) produced sufficiently accurate analyses of sequence datasets that are large and evolve under high rates of evolution. Yet, phylogenetic analyses of sequence datasets containing more than 100,000 sequences are being attempted by at least two groups that we are aware of: the iPTOL project (iPlant Collaborative, 2013) and the Thousand Transcriptome project (1KP) (Wong, 2013), and little is known about how well alignment methods perform on such ultra-large datasets.

We present PASTA, practical alignments using SATé and TrAnsitivity. PASTA begins with an alignment and tree estimated using a very simple profile HMM-based technique and then realigns the sequences using the tree. If desired, a new tree can be estimated on the new alignment, and the algorithm can iterate. We demonstrate PASTA's speed and accuracy on a collection of biological and simulated datasets, including a 200K-sequence RNASim dataset (Guo et al., 2009), which we align in less than 24 hours using PASTA on a 12-core machine.

[1] Department of Computer Science, University of Texas at Austin, Austin, Texas.
[2] Genomics and Computational Biology Graduate Group, University of Pennsylvania, Philadelphia, Pennsylvania.
[3] Institute for Biomedical Informatics, University of Pennsylvania, Philadelphia, Pennsylvania.
[4] Department of Computer Science, University of Illinois at Urbana-Champaign, Urbana, Illinois.

2. PASTA

PASTA uses an iterative strategy, and each iteration involves six steps (Fig. 1). The first iteration begins with the starting tree, and subsequent iterations begin with the tree estimated in the previous iteration. In each step, the guide tree is used to divide the set of sequences into smaller subsets, and to build a spanning tree with these subsets as nodes. We independently estimate MSAs for all the sequence subsets. Then, pairs of MSAs corresponding to subset that are adjacent in the spanning tree are aligned together using existing tools that can align MSAs. The resulting collection of MSAs overlap each other and are compatible where they overlap. These properties enable us to merge these overlapping MSAs using transitivity and generate an MSA on the entire set of sequences. Finally, a maximum likelihood tree is estimated on the final alignment. We provide a description of these steps in their default settings, but see Appendix C.1 in the Supplementary Material (available online at www.liebertpub.com/cmb) for full details.

Default starting tree: We compute an alignment A of a random subset X of 100 sequences from S; this is called the ''backbone alignment.'' We use HMMER (Eddy, 2009; Finn et al., 2011) to compute an HMM on A, and to align all sequences in S-X one by one to A. We then construct an ML tree on this alignment using FastTree-2 (Price et al., 2010).

Step 1: We divide the sequence set into disjoint sets, $S_1, \ldots, S_m$, each with at most 200 sequences, using the current guide tree and the centroid decomposition technique in SATé-II, which works as follows. If the tree has at most 200 leaves, we return the set of sequences; otherwise, we find an edge in the tree that splits the set of leaves into roughly equal sizes, remove it from the tree, and then recurse on each subtree.

Step 2: We compute a spanning tree T^* on the subsets, $S_1, S_2, \ldots, S_m$, as follows. First we label all leaves by their subset. For every node v in the guide tree that is on a path between two leaves that both belong to S_i we label it by S_i. If some nodes are not yet labeled, we propagate labels from nodes to unlabeled neighbors (breaking ties by using the closest neighbor according to branch lengths in the guide tree) until all nodes are labeled. We then collapse edges that have the same label at the endpoints.

Step 3: We compute MSAs on each S_i using an existing MSA tool and refer to each such alignment as a *type 1 subalignment*. By default, we use Mafft (Katoh et al., 2005) with the L-INS-i settings, which is based on the iterative refinement method incorporating local pairwise alignment information.

Step 4: Every node in T^* is labeled by an alignment subset for which we have a type 1 subalignment from Step 3. For every edge (v, w) in T^*, we use OPAL (Wheeler and Kececioglu, 2007) to align the type 1 subalignments at v and w; this produces a new set of alignments, each containing at most $2k$ sequences, which are called *type 2 subalignments*. We require that the merger technique used to compute type 2 subalignments not change the alignments on the type 1 subalignments; therefore, type 2 subalignments induce the type 1 subalignments computed in Step 2 and are all compatible with each other.

Step 5: We compute the final alignment through a sequence of pairwise mergers using transitivity. Each MSA defines an equivalence relation on the letters (i.e., nongap positions) within its sequences, whereby two letters are in the same equivalence class *if and only if* they are in the same column (Fig. 1; last row, middle box). Hence, given two alignments A and B that induce identical alignments on their shared sequences (called *overlapping compatible* alignments henceforth), we can define an equivalence relation on the union of the letters from their sequence subsets as follows: a and b are in the same equivalence class for the merged alignment if and only if at least one of the following is true: (1) they are in the same equivalence class in A or B, or (2) there is some letter c such that a and c are in the same equivalence class in one alignment, and b and c are in the same equivalence class in the other alignment (Fig. 1, bottom right corner). The resulting equivalence relation defines an MSA on the union of sequences in A and B, and we refer to it as the transitivity merger of A and B. Using this definition, we use the spanning tree to merge all the type-2 subalignments through a sequence of pairwise transitivity mergers into a multiple sequence alignment on the entire set of sequences. Note that each subset is part of at least one type-2 alignment and each type-2 alignment overlaps with at least one more type-2 alignment (the adjacent edge in the spanning tree); thus, the final transitivity merger produces an alignment that includes all the sequences.

Step 6: If an additional iteration (or a tree on the alignment) is desired, we run FastTree-2 to estimate a maximum likelihood tree on the MSA produced in the previous step. We mask all columns that have more

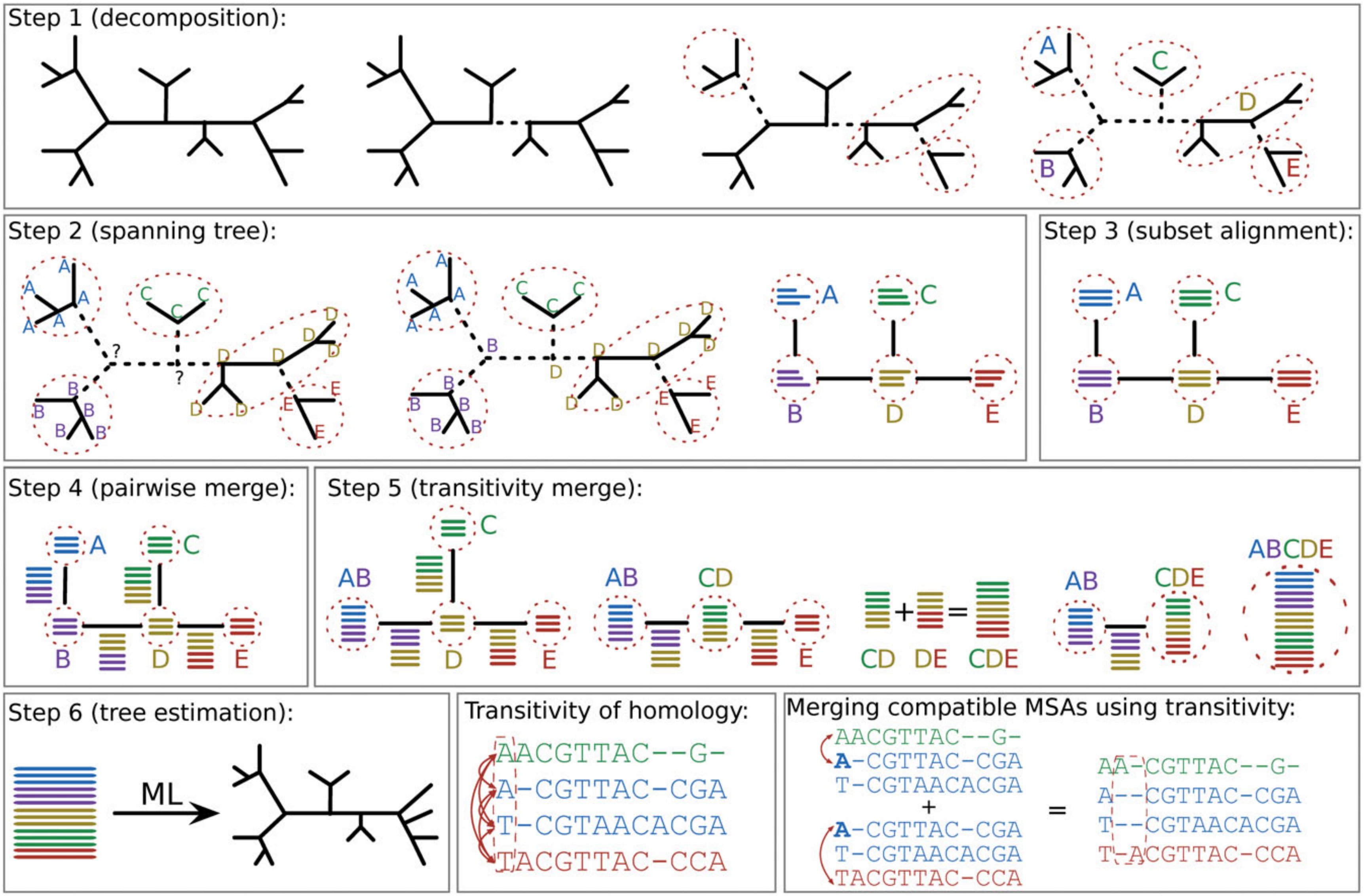

FIG. 1. Algorithmic design of PASTA. The first six boxes show the steps involved in one iteration of PASTA. The last two boxes show the meaning of transitivity for homologies defined by a column of an MSA, and how the concept of transitivity can be used to merge two compatible and overlapping alignments. MSA, multiple sequence alignment.

than 99.9% gaps in the alignment obtained in Step 5; this has no significant impact on the tree estimation, but reduces the running time (sometimes dramatically).

Running Time. The final output of Step 5 (transitivity merge) does not depend on the order in which edges of the spanning tree are processed, but the order can impact the running time. An *arbitrary* order of edge contractions can result in a worst case $O(qm^2 + Lm)$ running time. However, if we merge subalignments using the reverse order of the centroid edge deletions, then the running time can be bounded, as follows.

Theorem 1. *Given m type 1 alignments and $m - 1$ type 2 alignments, the algorithm to compute the transitivity merge of these alignments uses $O(qm \log m + Lm)$ time, where q is the maximum length of any sequence (not counting gaps) in any type 1 alignment, and L is the length of the output alignment.*

Proofs have been omitted due to space constraints (see Appendix C.2 in the Supplementary Material).

3. EXPERIMENTAL SETUP

3.1. Datasets

All the datasets used in this study are available online (See Appendix F in the Supplementary Material for download links).

Nucleotide: To explore performance on moderate-sized datasets, we used the 1000-sequence nucleotide datasets with average length 1000–1023 from Liu et al. (2009), which were generated using ROSE (Stoye et al., 1998).

To explore performance on larger datasets, we simulated 10,000-sequence datasets using Indelible v. 1.03 (Fletcher and Yang, 2009) under three different rates of evolution (10 replicates each), with average sequence length 1000.

To explore performance on ultra-large datasets (up to 200,000 sequences), we subsampled from the million-sequence RNASim (Guo et al., 2009) dataset (available online), with average sequence length 1556. RNASim is a simulator for RNA sequence evolution that we present here, and that was designed to simulate a complex molecular evolution process using a nonparametric population genetic model that generates long-range statistical dependence and heterogeneous rates. Briefly, the simulation is of an RNA molecule with both stabilizing and directional selection on the secondary structure of the molecule. RNA molecules are assumed to mutate both by substitutions and indels. The fixation probability of any potential mutation is computed using a fitness model that is a function of the folded free energy of the mutated RNA. The resulting distribution of evolved molecules display complex statistical characteristics that do not follow the standard parametric models. The simulated dataset displays many of the properties of naturally observed RNA molecules in terms of both morphological variation and optimization difficulty. See Appendix A in the Supplementary Material for more details about RNASim. We subsampled the million-sequence RNASim dataset to create datasets with 10,000, 50,000, 100,000, and 200,000 sequences. For the 10K RNASim dataset we made 10 replicates, but we used only one replicate of these datasets for the larger datasets.

We include three large biological datasets from the comparative ribosomal website (CRW) (Cannone et al., 2002): the 16S.3 dataset (6,323 sequences of average length 1557, spanning three phylogenetic domains), the 16S.T dataset (7,350 sequences of average length 1492, spanning three phylogenetic domains), and the 16S.B.ALL dataset (27,643 sequences of average length 1371.9, spanning the bacteria domain). These datasets have curated reference alignments based on secondary and tertiary structures. Reference trees for the biological datasets were computed using RAxML (Stamatakis, 2006) on the reference alignments and all edges with bootstrap support less than 75% were contracted; using other thresholds produces similar results (see Appendix E.3 in the Supplementary Material). The reference alignments and trees for the simulated datasets are the true alignment and true (model) trees.

Amino-acid: We used 10 large datasets (AA-10) with curated MSAs [the eight largest BAliBASE datasets from Thompson et al. (2011) and IGADBL_100 and coli_epi_100 from Gloor et al. (2005)]; ranging in size between 320 to 807 sequences. We also include 19 of the largest HomFam datasets that have between 10,099 to 93,681 sequences (the ‘‘rhv’’ dataset was omitted due to the warning on the Pfam website that the alignment is very weak), which were used by Sievers et al. (2011) to evaluate protein MSA methods on large datasets, and have Homstrad (Mizuguchi et al., 1998) reference alignments on very small subsets (5–20 sequences, median 7) of their sequences.

Methods. We compare PASTA to SATé-II version 2.2.7, Muscle version 3.8.31, Mafft version 7.143b (Katoh et al., 2005), Clustal-Omega version 1.2.0 (Sievers et al., 2011), and also to our approach for obtaining the starting alignment and tree. PASTA results are based on the default settings with three iterations. Mafft was run in its default settings wherever it could run, and otherwise we used Mafft-PartTree. We ran SATé-II for three iterations and with identical starting trees as PASTA. Due to the high computational costs of running OPAL on large datasets, we used Muscle for merging alignments inside SATé-II for datasets with 5,000 sequences or more, and otherwise we used the default settings in SATé-II. Finally, we used FastTree-2 version 2.1.5 to compute ML trees on each alignment. See Appendix D in the Supplementary Material for the commands.

Criteria. We measure the alignment accuracy, tree error, and running time. Alignment accuracy is measured using FastSP (Mirarab and Warnow, 2011) with two different metrics: the SP-score (the percentage of homologies in the reference alignment recovered in the estimated alignment) and the modeler score (the percentage of homologies in the estimated alignment that are correct), averaged together to get one measure, which we call ‘‘pairs score’’ for short (averaging these two scores amounts to penalizing false positive and false negative homologies equally). We also report the TC score (the number of columns that are recovered entirely correctly in the estimated alignment). For HomFam datasets, we measure the error with respect to a very small number of reference ‘‘seed’’ sequences for which a reliable alignment is provided. To measure tree error, we report the false negative (FN) rate, which is the percentage of true tree edges missing in the estimated trees. For AA datasets, since the seed alignments include only a handful of sequences, we measure only alignment accuracy and not tree error.

Computational Platform. We ran all methods on the Lonestar Linux cluster at TACC (Boisseau and Stanzione, 2013), and each run was given one node with 12 cores and 24 GB of memory. Since running time on Lonestar is limited to 24 hours, we were only able to run techniques that could finish in 24 hours (see Section 4). However, PASTA and SATé-II are iterative techniques, and we allowed them to perform as many iterations (but no more than three) as they could complete within 24 hours. We report the wall clock time in all cases.

4. RESULTS

Ability to complete analyses. We report which methods completed analyses within 24 hr using 12 cores and 24 GB of memory. All methods were completed on all datasets with at most 30,000 sequences, with the exception of Clustal-Omega, which was not able to run on the Indelible 10,000 M2 dataset. Clustal-Omega, Muscle, and SATé-2 failed to complete on the RNASim datasets with 50,000 sequences or more, and Mafft failed to complete on the RNASim dataset with 200K sequences. On 100k RNAsim, PASTA finished two iterations in 24 hr, and on 200k, PASTA was able to complete one iteration and was the only method that could run.

Nucleotide datasets. Results (alignment and tree error) on the 1000-sequence datasets are shown in the Supplementary Material, Appendix E.2; tree error results of ML trees on reference and estimated alignments for the other nucleotide datasets are shown in Figure 2. Unsurprisingly, ML trees computed on the true or reference alignments had the best accuracy, but PASTA trees matched or improved on the accuracy of SATé-II.

Table 1 compares methods with respect to the TC and pairs scores. On the Indelible datasets, PASTA had the most accurate alignments according to both measures of accuracy, and the difference between PASTA and other methods increased as the rate of evolution increased. On the RNASim data, PASTA had by far the most accurate alignments of all methods tested according to TC, and its pairs scores were better than all other methods except for the starting alignment.

On the 16S.T dataset, the starting alignment did not return an alignment with all the sequences, because HMMER considered one of the sequences unalignable; in general, though, the starting alignment had good alignment scores. Of the remaining methods, PASTA had the best pairs scores, and the other methods were substantially less accurate. With respect to TC scores, on 16S.B.ALL and 16S.T, PASTA had the highest accuracy, but on 16S.3, SATé-II had the highest accuracy (followed by Mafft and PASTA).

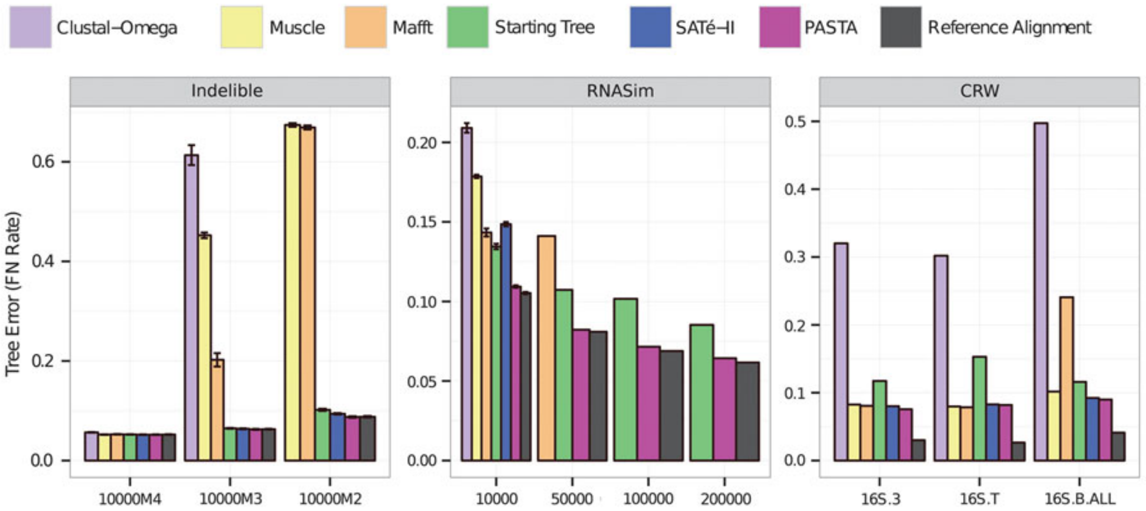

FIG. 2. Tree error rates on nucleotide datasets. We show missing branch (also known as false negative or FN) rates for maximum likelihood trees estimated on the reference alignment as well as alignments computed using PASTA and other methods; results not shown indicate failure to complete within 24 hr using 12 cores on the datasets. Error bars show standard error over 10 replicates for all model conditions of the Indelible and the 10,000-sequence RNASim datasets.

Alignment accuracy on AA datasets. Table 2 shows alignment accuracy on the AA datasets. Due to dataset sizes, Muscle and SATé-II failed to complete on two of the HomFam datasets, so we separate out the results for these two datasets from the remaining 17 HomFam datasets.

PASTA had the best pairs score or was tied for the best pairs score for both HomFam and AA-10 datasets. Mafft had the best TC score for HomFam(17), but PASTA was very close. For HomFam(2), PASTA had the best TC score and Mafft was a close second. On AA-10 datasets, SATé-II had the best TC score and was closely trailed by Mafft and PASTA.

Comparison to SATé-II on 50,000-taxon dataset. SATé-II could not finish even one iteration on the RNASim with 50,000 sequences running for 24 hr and given 12 CPUs on TACC. However, we were able to

Table 1. Alignment Accuracy on Nucleotide Datasets

	Indelible - 10,000			*RNASim*				*CRW (16S)*		
	M4	*M3*	*M2*	*10k*	*50k*	*100k*	*200k*	*16S.3*	*16S.T*	*16S.B.ALL*
Column (TC) score										
Clustal-O	160	10	X	13	X	X	X	12	0	1
Muscle	803	7	0	0	X	X	X	34	21	81
Mafft	337	13	0	28	30	26	X	75	85	15
Initial	422	106	18	11	15	5	4	33	X	24
SATé-II	977	758	792	35	X	X	X	**89**	60	87
PASTA	**987**	**920**	**1151**	**152**	**311**	**492**	**823**	71	**121**	**102**
Pairs score (mean of SP score and modeler score)										
Clustal-O	0.97	0.34	X	0.65	X	X	X	0.57	0.53	0.60
Muscle	**1.00**	0.12	0.01	0.35	X	X	X	0.74	0.67	0.66
Mafft	**1.00**	0.76	0.02	0.72	0.73	0.72	X	0.75	0.70	0.71
Initial	0.99	0.98	0.91	**0.87**	**0.88**	**0.87**	**0.88**	0.86	X	**0.95**
SATé-II	1.00	0.93	0.72	0.56	X	X	X	0.76	0.65	0.66
PASTA	**1.00**	**1.00**	**0.99**	0.85	0.85	**0.87**	0.86	**0.87**	0.83	0.94

We show the number of correctly aligned sites (top) and the average of the SP-score and modeler score (bottom). X indicates that a method failed to run on a particular dataset given the computational constraints. "Initial" corresponds to the alignment approach used to obtain the starting tree of PASTA (HMMER failed to align one sequence in the 16S.T dataset) and Clustal-O stands for Clustal-Omega.

Boldface indicates the best values for each model condition.

Table 2. Alignment Accuracy on AA Datasets

	Column (TC) score			*Pairs score*		
Method	*AA-10*	*HomFam(17)*	*HomFam(2)*	*AA-10*	*HomFam(17)*	*HomFam(2)*
Clustal-O	78	88	29	**0.76**	0.72	0.71
Muscle	48	51	X	0.70	0.52	X
Mafft	81	**103**	32	**0.76**	0.75	0.79
Initial	54	95	16	0.75	0.71	0.81
SATé-II	**83**	73	X	0.75	0.64	X
PASTA	80	102	**36**	**0.76**	**0.78**	**0.83**

We show TC (the number of correctly aligned sites, left) and the pairs score (the average of the SP-score and modeler score, right). X indicates that a method failed to run on a particular dataset given the computational constraints. ''Initial'' corresponds to the alignment approach used to obtain the starting tree of PASTA (HMMER failed to align one sequence in the 16S.T dataset). All values shown are averages over all datasets in each category.

Boldface indicates the best values for each model condition.

run two iterations of SATé-II on a separate machine with no running time limits (12 Quad-Core AMD Opteron processors, 256GB of RAM memory). Given 12 CPUs, two iterations of SATé-II took 137 hr, compared to 10 hr for PASTA. However, the resulting SATé-II alignment recovered only 30 columns entirely correctly while PASTA recovered 311 columns. The pairs score of SATé-II was extremely poor (38.2%), while PASTA was quite accurate (81.0%). The tree produced by SATé-II had higher error than PASTA (12.6% versus 8.2% FN rate).

Impact of varying algorithmic parameters. We compared results obtained using four different starting trees: a random tree, the ML tree on the Mafft-PartTree alignment, PASTA's default starting tree, and the true (model) tree (see Table 3). After one iteration, PASTA alignments and trees based on our starting tree or true tree had roughly the same accuracy, and the starting tree based on Mafft-PartTree resulted in only a slightly worse tree (1% higher FN rate). However, using a random tree resulted in much higher tree error rates (52.3% error), and alignments that were also less accurate. Interestingly, after three iterations of PASTA, no noticeable difference could be detected between results from various starting trees. Thus, PASTA is robust to the choice of the starting tree.

We also evaluated the impact of changing the alignment subset size (Table 4); these analyses showed that using alignment subsets of only 50 sequences improved the TC score and running time substantially, and only slightly changed the pairs score or tree error score. Although these analyses were performed only for two datasets, they suggest the possibility that improved results might be obtained through smaller alignment subsets.

Running Time. Figure 3 compares the running time (in hours) of different alignment methods. Note that PASTA was faster than SATé-II in all cases and could analyze datasets that SATé-II could not (i.e., the

Table 3. Effect of the Starting Tree on Final PASTA Alignment and Tree

Initial tree		*Alignment accuracy*		
method	*Error (FN)*	*Pairs score*	*TC*	*Tree error FN*
One iteration				
Random	100.0%	79.9%	2	52.3%
Mafft-parttree	28.7%	87.0%	126	11.7%
Starting tree	12.4%	86.8%	138	10.5%
True tree	0%	86.1%	133	10.5%
Three iterations				
Random	100.0%	90.4%	138	11.0%
Mafft-parttree	28.7%	83.9%	144	10.7%
Starting tree	12.4%	88.8%	145	10.7%
True tree	0%	90.8%	150	10.5%

Alignment accuracy and tree error is shown for PASTA with various starting trees, after one iteration (top) and three iterations (bottom) on one replicate of the 10k RNASim dataset.

Table 4. Impact of Alignment Subset Size

Dataset	*Subset size*	*Tree error FN*	*Alignment accuracy*		*Running times*
			Pairs score	*TC*	
RNASim 10K	200	10.7%	**88.8%**	145	13,478
RNASim 10K	100	**10.4%**	87.4%	185	8,235
RNASim 10K	50	10.7%	88.6%	**210**	6,015
16S.T	200	8.2%	**82.7%**	121	9,120
16S.T	100	8.1%	82.0%	125	7,086
16S.T	50	**7.9%**	79.0%	**129**	5,780

We report tree error and alignment accuracy on one replicate of the 10K RNASim dataset and also on the 16S.T dataset, using three iterations of PASTA in which we explore the impact of changing the subset size from 200 (the default) to 100 and 50; all other algorithmic parameters use default values. Boldface indicates the best performance on the data.

RNASim datasets with 50K or more sequences). PASTA was not always faster than other methods, but was able to complete its analyses of all datasets within the 24-hr time limit, whereas other methods (except the starting tree) were unable to complete analyses on the largest datasets.

Figure 4a presents a detailed running time comparison of PASTA and SATé-II on two specific model conditions of RNASim dataset. Note that merging subset alignments (and the last pairwise merge, shown in the dotted area) was the majority of the time used by SATé-II to analyze the 50K RNASim dataset, but a very small fraction of the time used by PASTA. PASTA uses transitivity for all but the initial pairwise mergers, and therefore scales well with increased dataset size, as shown in Figure 4b (the sub-linear scaling is due to a better use of parallelism with increased number of sequences). Finally, Figure 4c shows that PASTA is highly parallelizable and has a much better speed-up with increasing number of threads than SATé does.

5. SUMMARY

The key algorithmic contribution in PASTA is the use of transitivity to align sequences on a guide tree, which addresses computational limitations in SATé and also improves the alignment of very distantly related sequences and remote homology detection. PASTA is fast and scales well with the number of processors, so that datasets with even 200,000 sequences can be analyzed in less than a day with a small

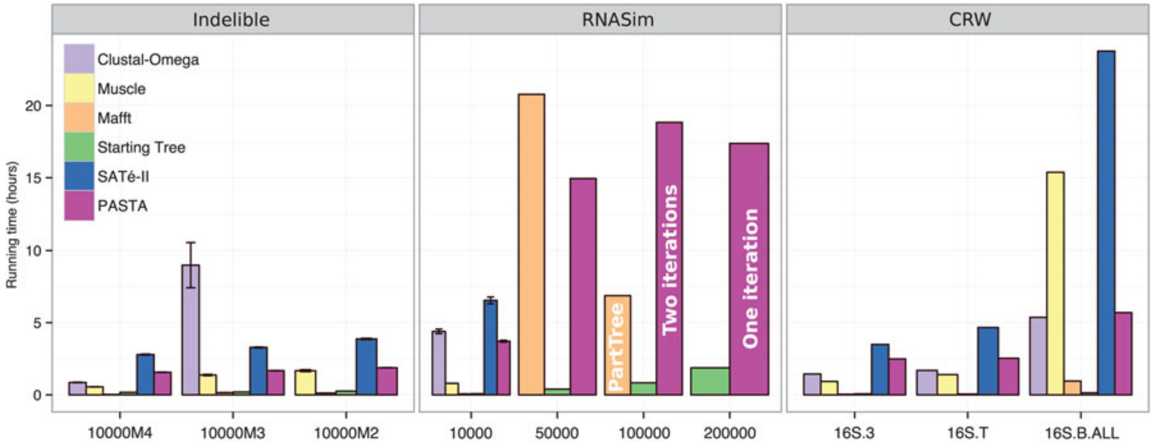

FIG. 3. Alignment running time (hours). Note that PASTA was run for three iterations everywhere, except on the 100,000-sequence RNASim dataset where it was run for two iterations, and on the 200,000-sequence RNASim dataset where it was run for one iteration. Mafft was run in default mode, except for the 100,000-sequences where PartTree was used.

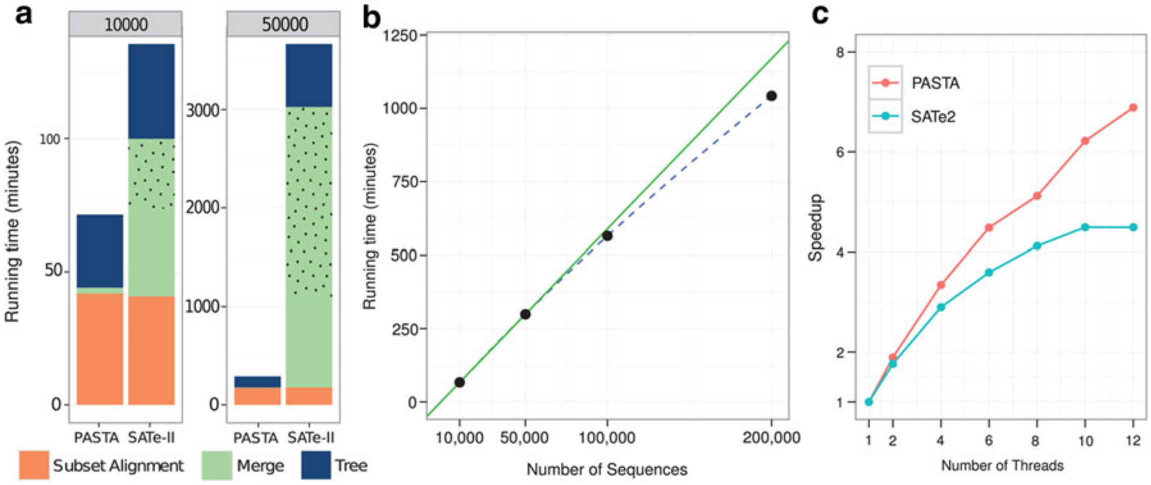

FIG. 4. Running time comparison of PASTA and SATé-II. **(a)** Running time profiling on one iteration for RNASim datasets with 10K and 50K sequences (the dotted region indicates the last pairwise merge); **(b)** running time for one iteration of PASTA with 12 CPUs as a function of the number of sequences (the solid line is fitted to the first two points); and **(c)** scalability for PASTA and SATé-II with increased number of CPUs.

number of processors. Thus, highly accurate alignment and phylogeny estimation is possible, even on hundreds of thousands of sequences, without supercomputers. (PASTA software is publicly available in open source form online).

ACKNOWLEDGMENTS

This research was supported in part by National Science Foundation (NSF) grant DBI 0733029 to T.W., by the Howard Hughes Medical Institute (HHMI) International Predoctoral Fellowship to S.M., and by a subgrant from the University of Alberta to T.W., made possible through a donation from Musea Ventures, which is held by Professor Gane Ka-Shu Wong. This work was also supported by a generous allocation from TACC and by the iPlant Collaborative, NSF grant DBI-1265383. L.W. and S.G. were supported by NSF grant 0331453; L.W. was supported in part by the National Institutes of Health (NIH) through grants T32-HG000046, NIA U24-AG041689, and R01-GM099962. J.K.'s participation in this project was funded, in part, by a Health Research Formula Fund from the Pennsylvania Department of Health, which disclaims responsibility for any analyses, interpretations, or conclusions. The authors thank the anonymous reviewers for their helpful comments.

AUTHOR DISCLOSURE STATEMENT

No competing financial interests exist.

REFERENCES

Boisseau, J., and Stanzione, D. 2013. TACC: Texas Advanced Computing Center. Available at: www.tacc.utexas.edu

Cannone, J.J., Subramanian, S., Schnare, M.N., et al. 2002. The comparative RNA web (CRW) site: an online database of comparative sequence and structure information for ribosomal, intron, and other RNAs. *BMC Bioinformatics* 3, 2.

Eddy, S. 2009. A new generation of homology search tools based on probabilistic inference. *Genome Informatics* 23, 205211.

Finn, R., Clements, J., and Eddy, S. 2011. HMMER web server: interactive sequence similarity searching. *Nucleic Acids Research* 39, W29–W37.

Fletcher, W., and Yang, Z. 2009. Indelible: A flexible simulator of biological sequence evolution. *Mol. Bio. Evol.* 26, 1879–1888.

Gloor, G.B., Martin, L.C., Wahl, L.M., and Dunn, S.D. 2005. Mutual information in protein multiple sequence alignments reveals two classes of coevolving positions. *Biochemistry* 44, 7156–7165.

Guo, S., Wang, L.-S., and Kim, J. 2009. Large-scale simulating of RNA macroevolution by an energy-dependent fitness model. arXiv:0912.2326.

iPlant Collaborative. 2013. iPTOL, assembling the tree of life for the plant sciences. Available at: https://pods.iplantcollaborative.org/wiki/display/iptol/Home

Katoh, K., Kuma, K., Toh, H., and Miyata, T. 2005. MAFFT version 5: Improvement in accuracy of multiple sequence alignment. *Nucleic Acids Res.* 33, 511–518.

Liu, K., Raghavan, S., Nelesen, S., et al. 2009. Rapid and accurate large-scale coestimation of sequence alignments and phylogenetic trees. *Science* 324, 1561–1564.

Liu, K., Warnow, T., Holder, M., et al. 2011. SATé-II: Very fast and accurate simultaneous estimation of multiple sequence alignments and phylogenetic trees. *Systematic Biology* 61, 90–106.

Mirarab, S., and Warnow, T. 2011. FastSP: Linear-time calculation of alignment accuracy. *Bioinformatics* 27, 3250–3258.

Mizuguchi, K., Deane, C., Blundell, T., and Overington, J. 1998. HOMSTRAD: a database of protein structure alignments for homologous families. *Protein Science* 7, 2469–2471.

Price, M., Dehal, P., and Arkin, A. 2010. FastTree-2 approximately maximum-likelihood trees for large alignments. *PLoS ONE* 5, e9490.

Sievers, F., Dineen, D., Wilm, A., and Higgins, D.G., 2013. Making automated multiple alignments of very large numbers of protein sequences. *Bioinformatics* 29, 989–995.

Sievers, F., Wilm, A., Dineen, D., et al. 2011. Fast, scalable generation of high-quality protein multiple sequence alignments using Clustal Omega. *Mol. Sys. Bio.* 7, 539.

Stamatakis, A. 2006. RAxML-VI-HPC: Maximum likelihood-based phylogenetic analyses with thousands of taxa and mixed models. *Bioinformatics* 22, 2688–2690.

Stoye, J., Evers, D., and Meyer, F. 1998. ROSE: generating sequence families. *Bioinformatics* 14, 157–163.

Thompson, J.D., Linard, B., Lecompte, O. and Poch, O. 2011. A comprehensive benchmark study of multiple sequence alignment methods: current challenges and future perspectives. *PloS ONE* 6, e18093.

Wheeler, T., and Kececioglu, J. 2007. Multiple alignment by aligning alignments. *Intelligent Systems for Molecular Biology*, 559–568.

Wong, G.K.-S. 2013. The thousand transcriptome (1KP) project. Available at: www.onekp.com/project.html

Address correspondence to:
Prof. Tandy Warnow
Department of Computer Science
University of Illinois at Urbana-Champaign
201 North Goodwin Avenue
Urbana, IL 61801

E-mail: warnow@illinois.edu