Supplementary Materials for "Phylogeny Estimation Given Sequence Length Heterogeneity"

February 28, 2023

1 Supplementary Material

The information provided here is supplementary to the paper by Smirnov, V. and Warnow, T., 2021. "Phylogeny estimation given sequence length heterogeneity." Systematic Biology, 70(2), pp.268-282.

1.1 Additional Tables

Method	1000M1	1000M2	1000M3	1000M4	RNASim	RNASim2	16S.M	23S.M
				SPFN				
PASTA	0.230	0.171	**0.049**	**0.010**	**0.109**	**0.067**	0.203	0.249
SEPP(F)	**0.197**	**0.156**	**0.052**	**0.012**	**0.104**	**0.062**	**0.174**	**0.238**
UPP(F)	**0.192**	**0.153**	**0.050**	**0.011**	**0.104**	**0.062**	**0.174**	**0.238**
SEPP(R)	**0.197**	**0.156**	**0.052**	**0.012**	**0.104**	**0.062**	**0.175**	**0.239**
UPP(R)	**0.192**	**0.152**	**0.050**	**0.011**	**0.104**	**0.062**	**0.174**	**0.238**
				SPFP				
PASTA	0.221	0.165	**0.049**	**0.011**	**0.109**	**0.068**	0.220	**0.322**
SEPP(F)	**0.176**	**0.141**	**0.045**	**0.011**	**0.104**	**0.063**	**0.193**	**0.312**
UPP(F)	**0.177**	**0.142**	**0.046**	**0.011**	**0.104**	**0.063**	**0.195**	**0.312**
SEPP(R)	**0.176**	**0.141**	**0.045**	**0.011**	**0.104**	**0.063**	**0.192**	**0.312**
UPP(R)	**0.177**	**0.141**	**0.046**	**0.011**	**0.104**	**0.063**	**0.195**	**0.312**

Table 1: **Alignment error under low fragmentation.** Each condition has 75% full-length sequences and 25% fragmentary sequences with an average 50% length. SEPP(X) and UPP(X) operate in four steps: first they compute a PASTA alignment on the full-length sequences, then they compute a backbone tree on the backbone alignment using ML heuristic "X" (RAxML or FastTree), then they build an ensemble of profile HMMs on the backbone tree, and finally they add the fragmentary sequences into the backbone alignment. The best results for each model condition (within 1%) are shown in boldface. The error rates are averaged over 20 replicates for the simulated datasets.

Method	1000M1	1000M2	1000M3	1000M4	RNASim	RNASim2	16S.M	23S.M
				SPFN				
PASTA	0.640	0.461	0.090	**0.016**	0.288	0.140	0.396	0.277
SEPP(F)	0.238	0.191	**0.070**	**0.014**	**0.112**	**0.069**	**0.179**	**0.239**
UPP(F)	**0.226**	**0.180**	**0.062**	**0.013**	**0.112**	**0.068**	**0.179**	**0.237**
SEPP(R)	0.239	0.190	**0.070**	**0.014**	**0.112**	**0.069**	**0.179**	**0.238**
! UPP(R)	**0.226**	**0.180**	**0.062**	**0.013**	**0.112**	**0.068**	**0.178**	**0.237**
				SPFP				
PASTA	0.613	0.444	0.088	**0.015**	0.185	0.113	**0.141**	0.338
SEPP(F)	**0.188**	**0.153**	**0.050**	**0.011**	**0.110**	**0.067**	0.197	**0.307**
UPP(F)	**0.188**	**0.153**	**0.049**	**0.011**	**0.110**	**0.067**	0.199	**0.306**
SEPP(R)	**0.188**	**0.153**	**0.050**	**0.011**	**0.110**	**0.067**	0.196	**0.301**
UPP(R)	**0.188**	**0.153**	**0.049**	**0.011**	**0.110**	**0.067**	0.199	**0.306**

Table 2: **Alignment error under high fragmentation.** PASTA is run in default mode. SEPP(X) and UPP(X) operate in four steps: first they compute a PASTA alignment on the full-length sequences, then they compute a backbone tree on the backbone alignment using ML heuristic "X" (RAxML or FastTree), then they build an ensemble of profile HMMs on the backbone tree, and finally they add the fragmentary sequences into the backbone alignment. Each condition has 50% full-length sequences and 50% fragmentary sequences with an average 25% length. The error rates are averaged over 20 replicates for the simulated datasets. The best results for each model condition (within 1%) are shown in boldface.

Method	1000M1	1000M2	1000M3	1000M4	RNASim	RNASim2	16S.M	23S.M
FN Rate:								
PASTA-FastTree	0.252	0.185	**0.103**	**0.063**	0.199	0.180	0.143	**0.143**
PASTA-RAXML	0.246	0.181	**0.095**	**0.061**	**0.186**	**0.163**	0.135	**0.137**
UPP(F)-FastTree	0.213	0.172	0.129	**0.065**	0.247	0.218	0.150	0.185
UPP(F)-RAXML	**0.159**	**0.127**	**0.095**	**0.061**	**0.185**	**0.163**	**0.121**	0.167
UPP(R)-FastTree	0.210	0.171	0.125	**0.066**	0.248	0.217	**0.112**	0.167
UPP(R)-RAXML	**0.157**	**0.128**	**0.094**	**0.061**	**0.185**	**0.163**	**0.112**	**0.143**
FP Rate:								
PASTA-FastTree	0.255	0.189	**0.108**	**0.085**	0.199	0.180	0.598	**0.476**
! PASTA-RAXML	0.248	0.185	**0.100**	**0.083**	**0.186**	**0.163**	0.595	**0.473**
UPP(F)-FastTree	0.215	0.176	0.134	**0.087**	0.247	0.218	0.598	0.502
UPP(F)-RAXML	**0.162**	**0.131**	**0.100**	**0.083**	**0.185**	**0.163**	**0.588**	0.491
UPP(R)-FastTree	0.213	0.175	0.130	**0.088**	0.248	0.217	**0.580**	0.491
UPP(R)-RAXML	**0.160**	**0.132**	**0.099**	**0.083**	**0.185**	**0.163**	**0.584**	**0.476**
Resolution:								
PASTA-FastTree	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
PASTA-RAXML	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
UPP(F)-FastTree	1.000	1.000	1.000	1.000	1.000	1.000	0.991	1.000
UPP(F)-RAXML	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
UPP(R)-FastTree	1.000	1.000	1.000	1.000	1.000	1.000	0.991	1.000
UPP(R)-RAXML	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000

Table 3: **Tree error rates and resolution under low fragmentation for MSA-ML methods.** We show FN rates (top), FP rates (middle), and resolution (bottom); each method is given by a pair U-V where U is the MSA method and V is the tree estimation method. Each condition has 75% full-length sequences and 25% fragmentary sequences (which have an average 50% length). The best results for each model condition (within 1%) are shown in boldface. The error rates are averaged over 20 replicates for the simulated datasets.

Method	1000M1	1000M2	1000M3	1000M4	RNASim	RNASim2	16S.M	23S.M
FN Rate:								
PASTA-FastTree	0.760	0.615	0.377	0.205	0.541	0.495	0.485	0.381
PASTA-RAXML	0.765	0.616	0.355	**0.164**	0.436	0.362	0.409	**0.321**
UPP(F)-FastTree	0.664	0.622	0.580	0.389	0.715	0.696	0.580	0.470
UPP(F)-RAXML	**0.373**	**0.307**	**0.236**	**0.168**	**0.377**	**0.336**	0.373	0.363
UPP(R)-FastTree	0.667	0.628	0.573	0.395	0.714	0.696	0.591	0.512
UPP(R)-RAXML	**0.370**	**0.304**	**0.237**	**0.167**	**0.377**	**0.338**	**0.340**	0.363
FP Rate:								
PASTA-FastTree	0.761	0.617	0.381	0.224	0.541	0.495	0.758	0.622
! PASTA-RAXML	0.766	0.618	0.359	**0.184**	0.436	0.362	0.723	**0.585**
UPP(F)-FastTree	0.666	0.624	0.582	0.404	0.715	0.696	0.802	0.676
UPP(F)-RAXML	**0.375**	**0.310**	**0.240**	**0.187**	**0.377**	**0.336**	0.706	0.611
UPP(R)-FastTree	0.668	0.629	0.576	0.409	0.714	0.696	0.808	0.702
UPP(R)-RAXML	**0.372**	**0.307**	**0.241**	**0.187**	**0.377**	**0.338**	**0.690**	0.611
Resolution:								
PASTA-FastTree	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
PASTA-RAXML	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
UPP(F)-FastTree	1.000	1.000	1.000	1.000	1.000	1.000	0.998	1.000
UPP(F)-RAXML	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
UPP(R)-FastTree	1.000	1.000	1.000	1.000	1.000	1.000	0.998	1.000
UPP(R)-RAXML	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000

Table 4: **Tree error rates and resolution under high fragmentation for MSA-ML methods.** We show FN rates (top), FP rates (middle), and resolution (bottom); each method is given by a pair U-V where U is the MSA method and V is the tree estimation method. Each condition has 50% full-length sequences and 50% fragmentary sequences with an average 25% length. The best results for each model condition (within 1%) are shown in boldface. The error rates are averaged over 20 replicates for the simulated datasets.

!

Method	1000M1	1000M2	1000M3	1000M4	RNASim	RNASim2	16S.M	23S.M
FN Rate:								
UPP(F)-pplacer	**0.217**	**0.181**	**0.153**	**0.114**	**0.244**	**0.220**	**0.173**	**0.208**
SEPP(F)-pplacer	**0.222**	**0.186**	**0.156**	**0.113**	**0.243**	**0.221**	0.202	0.220
SEPP(F)-pplacer(c)	0.227	**0.189**	0.161	**0.118**	0.280	0.235	0.247	0.274
UPP(F)-APPLES	0.279	0.230	0.198	0.150	0.379	0.334	0.240	0.298
UPP(R)-pplacer	**0.215**	**0.183**	**0.150**	**0.112**	**0.243**	**0.215**	0.192	0.244
SEPP(R)-pplacer	**0.218**	**0.186**	**0.152**	**0.112**	**0.243**	**0.216**	0.230	0.274
SEPP(R)-pplacer(c)	**0.225**	**0.188**	**0.160**	**0.117**	0.280	0.231	0.249	0.321
UPP(R)-APPLES	0.270	0.229	0.195	0.148	0.380	0.329	0.245	0.333
FP Rate:								
UPP(F)-pplacer	**0.181**	**0.144**	**0.114**	**0.092**	**0.207**	**0.184**	**0.594**	**0.492**
SEPP(F)-pplacer	**0.187**	**0.149**	**0.116**	**0.092**	**0.206**	**0.184**	0.608	**0.500**
SEPP(F)-pplacer(c)	0.193	**0.153**	0.122	**0.098**	0.243	0.200	0.632	0.534
UPP(F)-APPLES	0.247	0.198	0.164	0.132	0.345	0.301	0.620	0.546
UPP(R)-pplacer	**0.179**	**0.146**	**0.111**	**0.091**	**0.208**	**0.178**	0.604	0.517
SEPP(R)-pplacer	**0.184**	**0.149**	**0.113**	**0.091**	**0.207**	**0.179**	0.624	0.534
SEPP(R)-pplacer(c)	0.190	**0.153**	0.121	**0.096**	0.244	0.195	0.632	0.567
UPP(R)-APPLES	0.238	0.197	0.161	0.129	0.345	0.295	0.622	0.576
Resolution:								
UPP(F)-pplacer	0.952	0.952	0.950	0.953	0.953	0.956	0.954	0.953
SEPP(F)-pplacer	0.953	0.952	0.950	0.953	0.953	0.956	0.954	0.953
SEPP(F)-pplacer(c)	0.954	0.953	0.950	0.954	0.951	0.956	0.959	0.953
UPP(F)-APPLES	0.954	0.955	0.954	0.956	0.947	0.953	0.937	0.945
UPP(R)-pplacer	0.953	0.952	0.950	0.953	0.955	0.955	0.957	0.956
SEPP(R)-pplacer	0.954	0.952	0.950	0.953	0.954	0.955	0.959	0.953
SEPP(R)-pplacer(c)	0.954	0.954	0.951	0.954	0.951	0.956	0.957	0.956
UPP(R)-APPLES	0.955	0.956	0.954	0.956	0.946	0.952	0.937	0.960

Table 5: **Tree error rates and resolution for placement-based methods under low fragmentation.** We show FN rates (top), FP rates (middle), and degree of resolution (bottom). Each method is given by a pair U(B)-V where U is the extended alignment method, B is the backbone tree method, and V is the placement method. All methods run PASTA in default mode on the full-length sequences, and differ only in how they perform the remaining steps. Each condition has 75% full-length sequences and 25% fragmentary sequences (which have an average 50% length). The best results for each model condition (within 1%) are shown in boldface. The error rates are averaged over 20 replicates for the simulated datasets.

!

Method	1000M1	1000M2	1000M3	1000M4	RNASim	RNASim2	16S.M	23S.M
FN Rate:								
UPP(F)-pplacer	**0.487**	**0.437**	**0.379**	**0.320**	**0.507**	**0.476**	**0.499**	**0.429**
SEPP(F)-pplacer	0.514	0.459	0.395	**0.320**	**0.509**	**0.475**	**0.496**	**0.423**
SEPP(F)-pplacer(c)	0.534	0.483	0.420	0.334	0.560	0.510	0.549	0.506
UPP(F)-APPLES	0.559	0.517	0.472	0.394	0.716	0.674	0.591	0.542
UPP(R)-pplacer	**0.488**	**0.437**	**0.380**	**0.320**	**0.507**	**0.477**	**0.496**	0.458
SEPP(R)-pplacer	0.520	0.452	0.392	**0.319**	**0.508**	**0.475**	0.508	0.542
SEPP(R)-pplacer(c)	0.539	0.477	0.417	0.334	0.561	0.510	0.615	0.613
UPP(R)-APPLES	0.560	0.515	0.471	0.394	0.718	0.675	0.620	0.530
FP Rate:								
UPP(F)-pplacer	**0.369**	**0.307**	**0.233**	**0.172**	**0.397**	**0.356**	**0.716**	**0.586**
SEPP(F)-pplacer	0.403	0.335	0.255	**0.172**	**0.399**	**0.356**	**0.717**	**0.578**
SEPP(F)-pplacer(c)	0.431	0.367	0.288	0.189	0.459	0.401	0.743	0.629
UPP(F)-APPLES	0.455	0.408	0.356	0.270	0.641	0.595	0.759	0.659
UPP(R)-pplacer	**0.370**	**0.307**	**0.234**	**0.170**	**0.397**	**0.362**	**0.712**	0.613
SEPP(R)-pplacer	0.410	0.327	0.251	**0.170**	**0.400**	**0.359**	**0.720**	0.668
SEPP(R)-pplacer(c)	0.435	0.360	0.284	0.189	0.460	0.402	0.780	0.714
UPP(R)-APPLES	0.454	0.406	0.355	0.270	0.643	0.595	0.771	0.646
Resolution:								
UPP(F)-pplacer	0.810	0.808	0.805	0.801	0.817	0.815	0.827	0.844
SEPP(F)-pplacer	0.811	0.809	0.807	0.801	0.817	0.815	0.833	0.836
SEPP(F)-pplacer(c)	0.816	0.813	0.811	0.801	0.812	0.818	0.822	0.815
UPP(F)-APPLES	0.806	0.812	0.816	0.811	0.790	0.804	0.796	0.822
UPP(R)-pplacer	0.810	0.809	0.805	0.800	0.818	0.819	0.821	0.855
SEPP(R)-pplacer	0.812	0.810	0.808	0.800	0.819	0.819	0.824	0.844
SEPP(R)-pplacer(c)	0.814	0.813	0.811	0.801	0.814	0.820	0.821	0.825
UPP(R)-APPLES	0.803	0.813	0.815	0.810	0.791	0.804	0.777	0.811

Table 6: **Tree error rates for placement-based methods under high fragmentation.** We show FN rates(top), FP rates (middle), and resolution (bottom). Each method is given by a pair U(B)-V where U is the extended alignment method, B is the backbone tree method, and V is the placement method. All methods run PASTA in default mode on the full-length sequences, and differ only in how they perform the remaining steps. Each condition has 50% full-length sequences and 50% fragmentary sequences (which have an average 25% length). The best results for each model condition (within 1%) are shown in boldface. The error rates are averaged over 20 replicates for the simulated datasets.

The full-length datasets are available at:

ROSE: https://sites.google.com/eng.ucsd.edu/datasets/alignment/sate-i
RNASim: https://sites.google.com/eng.ucsd.edu/datasets/alignment/pastaupp
16S/23S: https://sites.google.com/eng.ucsd.edu/datasets/alignment/16s23s

2 Commands

2.1 PASTA 1.8.5

```
   python3 run_pasta.py -i backbone_sequences.txt -o ouput_dir
--temporaries temp_dir
```

2.2 RAxML-NG 0.9.0

```
   raxml-ng --msa sequences.txt --prefix name --threads 8
--seed 242234 --model GTR+G --tree pars{5}
```

2.3 FastTree 2.1

```
   FastTree -nt -gtr sequences.txt > output.tre
```

2.4 UPP 4.3.10

```
   python3 run_upp.py -s fragment_sequences.txt
-a backbone_alignment.txt -t backbone_tree.tre
```

2.5 SEPP 4.3.10

```
   python3 run_sepp.py -t backbone_tree.tre -a backbone_alignment.txt
-f fragment_sequences.txt -r raxml_info.txt
```

2.6 pplacer 1.1

```
   pplacer -t backbone_tree.tre -r backbone_alignment.fasta
-s raxml_info.txt -o output.json fragment_alignment.fasta
```

2.7 APPLES 1.2.0

APPLES requires the reference tree branch lengths to be estimated, which was done with FastTree as follows:

```
FastTree -nosupport -nt -nome -noml -intree backbone_tree.tre
< backbone_alignment.txt > backbone_tree_me.tre
```

Then, APPLES was run as follows:

```
python3 run_apples.py -t backbone_tree_me.tre -o output.json
-s backbone_alignment.txt -q fragment_alignment.txt
```

2.8 Alignment Error

We used FastSP 1.6.0fastsp to estimate alignment error:

```
java -jar FastSP_1.6.0.jar -r true_alignment.txt
-e estimated_alignment.txt
```

2.9 Tree Error

We used the compare_trees.py script (courtesy Erin Molloy), found in tools.zip at https://databank.illinois.edu/datasets/IDB-1424746, using the following command:

```
compare_trees.py <true_tree> <estimated_tree>
```

The script returns the number of false negative and positive edges, which were divided by the number of internal edges in the true tree and estimated tree, respectively.