V. EXPERIMENTS

We tested both algorithms using a selection of non-recombining DNA sequences. These include mitochondrial DNA samples from two human populations [28] and a chimpanzee population [27], Y chromosome samples from human [19] and chimpanzee populations [27], and a bacterial DNA sample [21]. Such non-recombining data sources provide a good test of the algorithms' ability to perform inferences in situations where recurrent mutation is the probable source of any deviation from the perfect phylogeny assumption.

We implemented variants of both algorithms. The simple algorithm was derandomized and used along with a standard implementation of the Dreyfus-Wagner routine. For the FPT algorithm, we implemented the randomized variant described above using an optimized Dreyfus-Wagner routine. The randomized algorithm takes two parameters, q and p, where q is the imperfectness and p is the maximum probability that the algorithm has failed to find an optimal solution of imperfectness q. On each random trial, the algorithm tallies the probability of failure of each random guess, allowing it to calculate an upper bound on the probability that that trial failed to find an optimal solution. It repeats random trials until the accumulated failure probability across all trials is below the threshold p. An error threshold of 1% was used for the present study.

The results are summarized in Table I. Successive columns of the table list the source of the data, the input size, the optimal penalty q, the parsimony score of the resulting tree, the run times of both of our algorithms in seconds, and the number of trials the randomized FPT algorithm needed to reach a 1% error bound. All run times reported are based on execution on a 2.4 GHz Intel P4 computer with 1 Gb of RAM. One data point, the human mtDNA sample from the Buddhist population, was omitted from the results of the simple algorithm because it failed to terminate after 20 minutes of execution. All other instances were solved optimally by the simple algorithm and all were solved by the randomized FPT algorithm. The randomized variant of the FPT algorithm in all but one case significantly outperformed the derandomized simple algorithm in run time. This result that may reflect the superior asymptotic performance of the FPT algorithm in general, the performance advantage of the randomized versus the deterministic variants, and the advantage of a more highly optimized Dreyfus-Wagner subroutine. The randomized algorithm also generally needed far fewer trials to reach a high