Evolutionary Placement of Short Sequence Reads

Performance, Accuracy and Web-Server for Evolutionary Placement of Short Sequence Reads under maximum-likelihood

Simon A. Berger, Denis Krompaß, Alexandros Stamatakis

The Exelixis Lab, Scientific Computing Group, Heidelberg Institute for Theoretical Studies, Schloss-Wolfsbrunnenweg 35, D-69118, Heidelberg, Germany

Abstract

We present an Evolutionary Placement Algorithm (EPA) and a Web-Server for the rapid assignment of sequence fragments (short reads) to edges of a given phylogenetic tree under the maximum-likelihood (ML) model. The accuracy of the algorithm is evaluated on several real-world data sets and compared to placement by pair-wise sequence comparison, using edit distances and BLAST.

We introduce a slow and accurate as well as a fast and less accurate placement algorithm. For the slow algorithm, we develop additional heuristic techniques that yield almost the same run times as the fast version with only a small loss of accuracy. When those additional heuristics are employed, the run time of the more accurate algorithm is comparable to that of a simple BLAST search for data sets with a high number of short query sequences. Moreover, the accuracy of the EPA is significantly higher, in particular when the sample of taxa in the reference topology is sparse or inadequate. Our algorithm, which has been integrated into RAxML, therefore provides an equally fast, but more accurate alternative to BLAST for tree-based inference of the evolutionary origin and composition of short sequence reads. We are also actively developing a web-server that offers a freely available service for computing read placements on trees using the EPA.

Keywords: maximum-likelihood, short sequence reads, phylogenetic placement, RAxML, metagenomics

Identification of organisms from, for example, microbial communities increasingly relies on analysis of DNA extracted from soil or water samples containing many, often unknown, organisms rather than from the individual organisms. Recently, the advent of new DNA sequencing techniques (e.g., pyrosequencing; Ronaghi, 2001) has increased the amount of sequence data available for identification and analysis of microbial communities by several orders of magnitude. This rapid increase in the amount of sequence data available poses new challenges for short-read sequence identification tools. We can no longer expect that the steady increase in computing power, according to Moore's law, is fast enough to handle this *flood of sequence data*.

Depending on the DNA sequencing method used, a single sequencing run can already generate more than 100,000 short read sequences, which comprise sequence fragments with a length of approximately 30 to 450 nucleotides (base pairs). Such sequencing runs can be carried out within about an hour. Besides rapid full-genome assembly, another important application is the sampling of microbial communities from, for example, permafrost-affected soils (Ganzert et al., 2007), vertebrate guts (Ley et al., 2005; Turnbaugh et al., 2008; Ley et al., 2008), hypersaline mats (Ley et al., 2006), or on hands of humans (Fierer et al., 2008).

Given the large amount of short-read sequences that meta-genomic studies of microbial communities often yield and the fact that the provenance of these often is unknown, the first step in studies of

meta-genomic data is to identify the biological origin of these reads. This assignment of short reads to known organisms then allows us to examine and compare microbial samples and communities (see Turnbaugh et al., 2008). For instance, 20% of the reads in one sample might be most closely related to a specific taxonomic group of bacteria, while in a different sample only 5% may be associated to this group.

Here we present a novel algorithm, the Evolutionary Placement Algorithm (EPA), for rapid phylogenetic identification of anonymous query sequences (QS), using a set of full-length reference sequences (RS). The most straight-forward approach to identify the origin of a QS is to use tools that are based on sequence similarity (e.g., BLAST; Altschul et al., 1997). However, the BLAST-based approach has an important limitation: It can yield misleading assignments of QS to RS if the sample of RS does not contain sequences that are sufficiently closely related to the QS (i.e., if the taxon sampling is sparse or inappropriate). Any approach based on pair-wise sequence similarity, like BLAST, will not unravel, but silently ignore, potential problems in the taxon sampling of the RS. For instance, given two RS a and b, a QS q may be identified by BLAST as being most closely related to a. In reality, q might be most closely related to a RS c, which is not included in the set of RS. Since this is a known problem (Koski and Golding, 2001), recent studies of microbial communities have begun to employ phylogenetic methods for QS identification (von Mering et al., 2007), despite the significantly higher computational cost. This treatment of short sequence reads is related to phylogenetic tree reconstruction methods that employ stepwise addition of sequences (Kluge and Farris,

1969), with the difference that each QS is individually placed in the phylogenetic reference tree (RT). If a QS connects to an internal edge of a RT comprising the RS (i.e., it is not located near a leaf of the tree), then this indicates that the sampling of the RS is insufficient to identify and characterize the diversity of the QS. This can be used as a means to identify parts of the tree in which taxon sampling is sparse and to guide sequencing efforts to improve the sampling.

To date, phylogeny-based identification of the provenance of anonymous reads is conducted as follows: the QS are aligned with respect to a reference alignment (RA) for the RS, and then inserted into the reference tree either via a complete *de novo* tree reconstruction, a constrained tree search, using the RT as a constraint or backbone, or a fast and/or approximate QS addition algorithm, such as ARB (Ludwig et al., 2004), which uses maximum parsimony (MP). For DNA barcoding, phylogeny-based Bayesian analysis methods have recently been proposed by Munch et al. (2008) as well as Nielsen and Matz (2006); these, however, are applied to trees with significantly less taxa. Recently, Brady and Salzberg (2009) proposed the Phymm and PhymmBL algorithms for metagenomic phylogenetic classification, and report an improved classification accuracy for simulated QS compared to BLAST for PhymmBL. Classification by Phymm is based on oligonucleotide composition, whereas PhymmBL uses a weighted combination of scores from Phymm and BLAST. These algorithms classify QS relative to a given database of un-aligned bacterial genomes (the NCBI RefSeq database; Pruitt et al., 2007) and their phylogenetic labels as provided in the RefSeq database (i.e., from Phylum

level down to Genus level). This is different to the EPA, which can be used with any set of aligned RS and places QS into a fully resolved bifurcating RT. Because of the different focus of Phymm and the EPA, it is currently impossible to directly compare their accuracy on the same data sets (e.g., multiple sequence alignments and fully resolved phylogenies are not provided by the NCBI RefSeq database). This also hinders direct comparison to other previous phylogenetic classification methods like PhyloPythia (McHardy et al., 2007), which is the only phylogenetic classifier that has been compared to Phymm (Brady and Salzberg (2009) show that Phymm and BLAST substantially outperform PhyloPythia) and MEGAN (Hudson et al., 2007). However, it is possible to compare the relative performance of the different methods (Phymm, PhymmBL, and EPA) to placements/classifications obtained by using BLAST, on data sets that are suitable for the respective methods. Therefore, we mainly compare the accuracy of the EPA to basic BLAST searches and discuss the analogous accuracy evaluation performed for Phymm/PhymmBL.

A very similar algorithm to the EPA, called pplacer, has been developed independently by Matsen et al. (2010). The execution times of pplacer are comparable to those of the EPA according to a joint performance study conducted by F. Matsen, A. Stamatakis, and S.A. Berger. A comparative study is included in Matsen et al. (2010).

The current standard approach for analysis of environmental reads yields a fully resolved bifurcating tree that often comprises more than 10,000 sequences (Fierer et al., 2008; Turnbaugh et al., 2008). The alignments used to reconstruct these trees mostly comprise only a single

gene, typically 16S or 18S rRNA. The reconstruction of such large trees with thousands of taxa, based on data from a single gene, is time-consuming and hard because of the weak historical signal in the alignment, which results in a decreased reconstruction accuracy for trees with many, but relatively short sequences (see Bininda-Emonds et al., 2000; Moret et al., 2002). Moreover, in metagenomic data sets a large number of QS will only have a length of approximately 200-450 base pairs if a 454 sequencer is used. Thus, for identification of the provenance of short read QS, the lack of historical signal becomes even more prevalent and critical if a comprehensive tree is reconstructed. In order to solve the problems associated with the lack of signal and to significantly accelerate the analysis, we advocate a different approach that only computes the optimal insertion position for every QS in the RT with respect to its ML score.

We introduce a new algorithm for the phylogenetic placement of QS and thoroughly test the placement accuracy on eight published data sets. We assess the impact of QS length on placement accuracy and conduct tests on short reads derived from original full length sequences of the test data sets. Because phylogenetic placement is inherently more computationally intensive than BLAST-based placement, performance optimization is an important factor in the development of such an algorithm if it is to become a useful and fast alternative to BLAST. Therefore, we have devised several evolutionary placement algorithms and heuristics with varying degrees of computational complexity.

The algorithm, which has been developed and tested in cooperation with microbial biologists, is already available in the latest open-source code

release of RAxML (Stamatakis, 2006b) (version 7.2.7, released in August 2010, `http://wwwkramer.in.tum.de/exelixis/software.html`). The alpha version of the respective web-service is available at `http://i12k-exelixis3.informatik.tu-muenchen.de/raxml`. Our new approach represents a useful, scalable and fast tool for evolutionarily sound identification of the provinance of environmental QS. The EPA and pplacer are currently the only algorithms that can perform the task described here. The parallelization of the EPA (Stamatakis et al., 2010) and the ability to conduct placements under all time-reversible substitution models and data-types offered by RAxML is a unique feature of the EPA that helps it to scale well on large and diverse data sets. Pplacer can infer placements using either ML (like the EPA) or Bayesian posterior probabilities.

Evolutionary Placement Algorithm

The input for our evolutionary placement algorithm consists of a RT comprising the r RS, and a large comprehensive alignment that contains the r RS *and* the q QS. The task of aligning several QS with respect to a given RS alignment can be accomplished with ARB (Ludwig et al., 2004), NAST (DeSantis et al., 2006), MUSCLE (Edgar, 2004), MAFFT (Katoh et al., 2005) or, as tested here, with HMMER (Eddy, 1998). One key assumption is, that the RT is biologically well-established or that it has been obtained via a preceding thorough phylogenetic analysis. In a typical usage scenario, the EPA could be used for mapping distinct microbial samples, for instance, from time series experiments or different

locations/individuals, to the same, for example, microbial reference tree in order to compare communities. Because the same RT can be used in multiple placement runs, the inference of the RT is not part of the EPA.

Initially, the algorithm will read the RT and reference alignment and mark all sequences *not* contained in the RT as QS. Thereafter, the ML model parameters and edge lengths on the RT will be optimized using the standard procedures implemented in RAxML (a description of the maximum-likelihood model can be found in the supplementary material).

Once the model parameters and edge lengths have been optimized on the RT, the actual identification algorithm is invoked. It will visit the $2r - 3$ edges of the RT by a preorder tree traversal, starting at an arbitrary edge of the tree leading to a tip (i.e., visit the current edge first and then recursively visit the two unvisited neighboring edges). At each edge, initially the probability vectors of the RT to the left and the right will be re-computed (if they are not already oriented towards the current edge). Thereafter, the program will successively insert, and subsequently remove, one QS at a time into the current edge and compute the likelihood (henceforth denoted as the insertion score) of the respective tree containing $r + 1$ taxa. The insertion score will then be stored in a $q \times (2r - 3)$ table that keeps track of the insertion scores for all q QS into all $2r - 3$ edges of the RT. In order to more rapidly compute the per-edge insertions of the QS, we use an approximation that is comparable to the Lazy Subtree Rearrangement (LSR) moves that are deployed in the standard RAxML search algorithm (see Stamatakis et al., 2005). After inserting a QS into an edge of the RT, we would normally need to re-optimize all edge lengths of

the $r+1$ tipped tree in order to obtain the corresponding insertion score. Instead, we only optimize the three edges adjacent to the insertion node of the QS (see Figure 1) before computing the likelihood of the insertion. This approach rests on the same rationale that was used to justify the LSR moves. Our experimental results justify this approximation because it yields a high placement accuracy. We use two methods, like those used for the LSR moves, to re-estimate the three edges adjacent to the insertion edge: a *fast* method and a *slow* method that uses the Newton-Raphson method. The *fast* method simply splits the insertion edge, b_r, in the RT into two parts, b_{r1} and b_{r2}, by setting $b_{r1} = b_{r2} = b_r/2$, and $b_q = 0.9$ (i.e. the edge leading to QS), where 0.9 is the default RAxML value to initialize edge lengths. These values were chosen empirically for good placement accuracy over varying input data. Note that, we used the *slow* method, that is, the most accurate placement method available in RAxML without the above approximations, for the main accuracy evaluation presented here. Thereafter, we also conducted a separate comparison of the *slow* and *fast* methods. The *slow* method repeatedly applies the Newton-Raphson method to all three edges (b_{r1}, b_{r2} and b_q) until no further application of the Newton-Raphson method is needed (i.e., when $\epsilon \leq 0.00001$, where ϵ is the edge length change between two invocations of the Newton-Rhapson method). Alternatively, our algorithm can also use MP to pre-score and order promising candidate insertion edges in order to further accelerate the placement process.

The output of this procedure consists of the RT, enhanced by assignments of the QS to edges of the RT. Each QS is attached to the edge

that yielded the best insertion score for the specific QS. Hence, the algorithm will return a multi-furcating tree if two or more QS are assigned to the same edge. An example is depicted in Figure 2.

The EPA algorithm can optionally use the non-parametric bootstrap (Felsenstein, 1985) to account for uncertainty in the placement of the QS. An example for this is shown in Figure 3. Thus, a QS might be placed repeatedly onto different edges of the RT with various levels of support. For the bootstrap procedure, we introduce additional heuristics to accelerate the insertion process. During the insertions onto the RT using the original alignment we keep track of the insertion scores for *all* QS into *all* edges of the RT. For every QS we can then sort the insertion edges by their scores and for each bootstrap replicate only conduct insertions for a specific QS into 10% of the best-scoring insertion edges on the RT. This reduces the number of insertion scores to be computed per QS on each bootstrap replicate by 90% and therefore approximately yields a ten-fold speedup for the bootstrapping procedure. In a typical application scenario, one may determine the diversity of the environmental sample for every replicate using, for instance, UniFrac (Lozupone and Knight, 2005), and then compute an average diversity over all replicates.

As a faster alternative to the non-parametric bootstrap, the insertion scores can also be directly used to compute placement uncertainty. von Mering et al. (2007) used expected likelihood weights (ELW; Strimmer and Rambaut, 2002) to assign QS to an area of a tree with a certain confidence. Methods for calculating a placement uncertainty using ELW are already implemented in the EPA and pplacer.

In order to improve the runtime of the *slow* insertion method, we developed two heuristics that rely on the *fast* scoring approach (described above) or MP scores, respectively. Given those pre-scoring techniques, the number of insertion positions considered for the thorough, but slow insertion process, can be reduced to a fraction of promising candidate edges. The proportion of insertion edges suggested by the rapid pre-scoring heuristics for analysis under the slow insertion method is determined by a user-defined parameter fh. As part of our performance evaluation, we tested the ML- and MP-based heuristics with regard to this parameter setting.

It is a known problem (Jermiin et al., 2004) that compositional heterogeneity can bias phylogenetic inference because it implies that the sequences cannot have evolved under the same stationary, reversible and homogeneous conditions (assumed by all the available time-reversible substitution models). In the case of the EPA, this can be problematic for the placement of QS onto potentially very short edges of the RT. Different approaches exist to resolve those problems by using more sophisticated models (e.g. Galtier and Gouy, 1998; Jayaswal et al., 2007). Nonetheless, time-reversibility is required to accommodate the high computational demands of large scale tree and EPA inferences, in particular with respect to computing the likelihood on trees (the "pulley principle"; Felsenstein, 1981). Therefore, we recommend that users of EPA test their alignment of RS for evidence that the sequences have not evolved under time-reversible conditions (e.g., using methods published by Ababneh et al. (2006) and Ho et al. (2006)) before using the EPA. If the data have not evolved under

time-reversible conditions, then the conclusions drawn from using the EPA should be given with this caveat in mind. These considerations do not affect our accuracy assessment, because trees of full length sequences and placements have been inferred under the same model and the potential errors that may occur because of inappropriate model selection affect both tree construction and placement.

Experimental Setup

Data Sets

To test the accuracy of the EPA and competing approaches, we used eight reference alignments (RA) of nucleotides or amino acids from 140 up to 1,604 sequences. The experimental data span a broad range of organisms and include rbcL genes (D500), small subunit rRNA (D150, D218, D714, D855, D1604), fungal DNA (D628) and amino acid sequences from *Papillomaviridae* (D140). For each set we computed the most likely tree and obtained bootstrap support (BS) (Stamatakis et al., 2008) values for the internal edges; this ML tree was denoted the RT. The data sets are available at `http://wwwkramer.in.tum.de/exelixis/epaData.tar.bz2`. The selection of the data sets and data types per se is not important, as long as QS with well supported positions can be extracted from them. It should be noted, that the number of data sets that could be assessed was limited by the excessive computational requirements of the leave-one-out experiments described in the following sections. The question we intend to answer by these experiments is this: can the EPA place a QS of reduced

length onto approximately the same position from which the full-length QS was pruned? We specifically did not include real metagenomic data sets in our study because the phylogenetic positions of the QS are unknown. Moreover, real metagenomic data sets do not allow for comparing the placement accuracy, or inferred placement position, between full-length and short-read sequences, whereas using a full-length sequence alignment and emulating short-reads allows for such a comparison. Therefore, we emulated metagenomic data sets using real-world alignments with ML trees and BS support, which is as close as one can get to reality for assessing placement accuracy using real sequence data. To analyze real metagenomic data we used the parallel version of the EPA, which can be applied to very large real metagenomic data sets (4,874 RS and 100,627 QS; Stamatakis et al., 2010).

Generation of QS

To evaluate the accuracy of our algorithm, we pruned one candidate QS at a time from the existing ML trees before reinserting the QS into the tree. We only pruned and reinserted those QS that were associated with high BS scores in the RT in order to assess placement accuracy for taxa whose position in the original tree is reliable. A candidate QS is considered to have high BS, when the BS of either one of the two edges to which the taxon is attached is $\geq 75\%$ and the other one leads to a neighboring tip (Fig. 4a), or if both of these edges have a BS $\geq 75\%$ (Fig. 4b). The 75% threshold reflects the typical empirical cutoff that is widely used in phylogenetics (Hillis and Bull, 1993). For each QS, a reduced RT is derived by pruning the respective tip from the original tree. The QS associated to

that taxon is then placed onto the reduced tree (Fig. 4c) with our EPA algorithm.

In our test data sets, the QS were always derived from the full-length sequences in the RA. In a typical application scenario, however, the placement algorithm will have to cope with QS that are significantly shorter than the full-length sequences in the RA. Hence, we carry out a systematic assessment of the placement accuracy depending on the length of the QS by artificially shortening the full length sequence in question via insertion of gaps. We deployed three distinct methods to produce a QS that are ordered according to increasing biological realism:

A first method to produce QS involved randomly replacing existing characters by gaps. While this method arguably does not reflect a real usage scenario, it provides a means to systematically assess the placement of QS over a wide variation of "virtual read lengths", while minimizing the influence of the position specific placement accuracy variation. Position specific effects on placement accuracy have previously been identified as a problem by Chakravorty et al. (2007). Multiple placement runs were conducted for QS with the relative proportion of non-gap characters set to 10%, 20%,..., 90%, up to the full sequence length. Because the sequences have been extracted from the original multiple alignment, the remaining non-gap characters are still aligned to the RA. Because the proportion of gaps is calculated relative to the length of the RA, the maximum proportion of available non-gap characters is alignment dependent. We emphasize that, mathematically, the introduction of random gaps does not influence the calculation of the likelihood function, because the used models

assume independence between sites. The results from these experiments show the qualitative relationship between QS length and placement accuracy, in a way that is not feasible with the more realistic QS generation method described below, because of the high computational requirements of these evaluations.

The second method to artificially shorten candidate QS involved randomly sampling contiguous subsequences from the full length sequence in question. This method to produce QS closely reflects the main EPA application scenario. Typically, a large number of short sequence reads generated by next generation sequencing methods from unknown positions, will need to be placed onto a RT. For every full-length sequence in question we sampled 20 QS with uniformly distributed positions and normally distributed lengths (mean length: 200 ± 60 bp for nucleotides; 70 ± 20 for amino acids). This roughly corresponds to the read lengths generated by current high throughput sequencing technologies. We sampled 20 QS per full-length sequence to minimize the aforementioned influence of position-specific bias on the placement of QS. Because of the high computational requirements of placing 20 QS per full-length sequence, we did not repeat this evaluation for different mean read-lengths. To assess the relationship between read length and placement accuracy we conducted the random-gap evaluation. Because a method is needed to align the short reads to the RA in a typical analysis using EPA, we also assessed placement accuracy using re-aligned QS and compared it to the QS placement accuracy induced by using the original alignment. We used HMMER to re-align the QS to a profile Hidden Markov Model (HMM) of the RA

(sequence-to-profile alignments with MUSCLE and MAFFT resulted in a slightly inferior placement accuracy; data not shown). Because the re-alignment procedure is not an integral part of the EPA, and future developments could potentially improve the re-alignment quality, we present results for the EPA with and without HMMER re-alignment.

A third method to produce QS involved generating these such that they correspond to paired-end reads (see Fig. 4c; in this experiment we excluded data set D140 because it is a multi-gene alignment of amino acid sequences). Thus, in contrast to the previous method, the position of the extracted subsequences within the gene remains fixed. This modification reflects another real-world scenario of the EPA, since paired-end sequencing is a widely used technique. Once again, we conducted our placement accuracy assessment on paired-end reads that were artificially generated from the full-length sequence in question by replacing all characters in the middle of the sequence by gaps. The artificial paired-end reads were 2x50 bp and 2x100 bp in length.

Comparison to Placements based on pair-wise Sequence Similarity

We conducted our evaluation of the accuracy of the EPA by comparison to a typical application scenario, in which appropriate sequence-based search tools such as BLAST are used to assign a QS to the most similar RS. In this setting, a QS will always be assigned to one of the terminal edges of the RT. As mentioned above, for the EPA tests we can choose to re-use the alignment information from the original multiple sequence alignment from which the QS were generated. With BLAST we

do not have this option, so all QS will effectively be re-aligned against the RS. For this reason we compare BLAST against the EPA with and without previous QS re-alignment using HMMER.

For all tests involving BLAST, we removed all gaps from the multiple alignment and built a BLAST database for each data set. We also removed all gaps from the candidate QS and concatenated the two ends of the artificial paired-end reads into one sequence. Searches with those sequences were conducted against the corresponding BLAST database. The default parameters of the BLAST program from the NCBI C Toolkit were used for character match/mismatch (scores 1 and -3) and gaps (non-affine gap penalty of -1). The default values from the NCBI BLAST web site with affine gap penalties were also tested, but produced slightly worse placement results and higher run times than the default settings. Using BLAST has the disadvantage that the information stemming from the RA that is present in the QS cannot be used, so for fairness the BLAST placements should be mainly compared to EPA placements with previous QS re-alignment.

For the tests on random gap QS, we did not use BLAST for comparison, because it is not well suited for aligning such QS. The gap model as well as the local alignment algorithm rely on contiguous sequence stretches of certain lengths, which were not present in the random gap QS. For the random gap evaluation we used a custom distance measure to calculate the pair-wise sequence similarities as an alternative to BLAST. It is defined as follows: For the two aligned sequences in question, we count the number of positions, where two different, non-gap characters are aligned

with each other. This measure corresponds to the Hamming (1950) distance, where a gap is a place-holder for any character. Placements derived from this distance measure will be referred to as sequence-based nearest-neighbor (SEQ-NN) placements.

Accuracy Measures

To quantify placement accuracy, we used two distance measures based on the topology and edge lengths of the original ML tree. In all cases, we considered an original edge and an insertion edge. The original edge is the one corresponding to the sequence used to generate the QS and into which it should ideally be re-inserted. The insertion edge is computed by the EPA. To quantify the distance between the original edge and the insertion edge we use the following two distance measures: The Node Distance (ND), is the unweighted path length in the original RT between the two edges. This corresponds to the number of nodes located on the path that connects the two edges (Fig. 5a) and represents an absolute distance measure. The second measure is the sum of edge lengths on the path connecting the two edges. This measure also includes 50% of the length of the insertion edge and 50% of the length of the original edge (Fig. 5a). For comparability between different trees and in order to obtain a relative measure, we normalize the edge path length by dividing it by the maximum tree diameter (Fig. 5b). The maximum diameter is the edge path of maximum length between two taxa in the RT. This distance measure is henceforth denoted as normalized edge distance (NED%).

Results and Discussion

Placement Accuracy for Random Gap QS

To assess the influence of QS length and at the same time reduce the impact of positional variability on placement accuracy, we tested the accuracy of the EPA on random-gap sequences of various lengths. Placements were carried out on the 8 data sets for varying artificial read lengths. Figure 6 provides a detailed plot of the accuracy as a function of the proportion of gaps, averaged over all candidate QS from all data sets (respective plots for the individual data sets can be found in the supplementary material). As a measure of accuracy we use the distance between the placement position and the true position from which the respective QS was originally pruned. Therefore, a lower distance indicates higher accuracy. For each placement method, we show the distances for three subsets of candidate QS: the disjunct sets of outer QS and inner QS (see Figure 4), as well as the complete set of QS (all QS).

As expected, the general trend is that placement accuracy increases with the QS lengths. For all three QS subsets, the EPA achieves higher placement accuracies than SEQ-NN. Generally, the distances of the EPA placements are at least two times lower than for SEQ-NN. For SEQ-NN, the placement accuracy is considerably lower for the inner QS subset compared to the other QS subsets. Placement based on pair-wise sequence similarity is harder for inner QS than for the outer QS because the candidate QS do not have direct neighbors in the RT. This decrease of placement accuracy is independent of the QS lengths. For the EPA, the

accuracy is more uniform across the three QS subsets. Only for short QS there is a slight accuracy decrease for inner QS (this effect is more pronounced for the NED% measure). With increasing QS length, the EPA placements become almost equally accurate for the three QS subsets. It is worth noting that, on average, the EPA correctly places almost all QS from all three subsets, when they contain less than 50% gaps. The results suggest that there is a steady increase in accuracy for increased QS lengths up to the 'perfect' placement on our test sets. This is particularly promising because read lengths will further increase.

The EPA placements on the inner QS compared to the outer QS are especially encouraging because this subset represents a worst-case scenario with respect to taxon sampling in the RT. In contrast to SEQ-NN, the original QS position has negligible impact on the placement accuracy. The results on this subset are indicative for the performance on data sets with a sparse or inadequate taxon sampling. Since it is hard to determine an adequate taxon sampling *a priori* for an unknown microbial community, our approach therefore can be used to appropriately adapt the taxon sampling.

Placement Accuracy for randomly selected Subsequences

Table 2 provides the placement accuracy (according to the ND measure) for the uniformly sampled contiguous subsequences of normally distributed lengths (mean: 200 ± 60bp and 70 ± 20 amino acid residues). The table values represent distances between the placement positions and the true positions from which the QS have been pruned. A ND of 0 represents a perfect placement, larger values indicate larger placement

errors. As in the previous section, we show separate results for outer QS, inner QS and all QS. In the second column, we show the placement accuracy of the EPA in terms of ND. For data set D140, this means that, averaged over all QS, the placements calculated by the EPA are within 0.51 nodes of the original placements. In the next column, we provide the average ND of the EPA for the case when the QS have been re-aligned to the RA using HMMER prior to the placement run. For D140, the average ND is 0.59, so the additional re-alignment step results in EPA placements that are on average 1.16 times further away from their original position than EPA placements without QS re-alignment. The third column gives the average ND for a BLAST-based approach. For D140, the value of 0.91 corresponds to placements that are on average 1.78 and 1.53 times further away from the true position than the EPA and EPA-RA placements. A corresponding table using the NED% measure is provided in the supplementary material. In general, the results are comparable to the ND results.

The values for D140 reflect the general trends which apply to most of the data sets. The EPA placements are on average more accurate than the BLAST placements, by factors between 1.12 and 2.06. Except for the two smallest (in terms of number of taxa) nucleotide data sets, the advantage of the EPA over alternative methods improves for the inner QS. For the smallest data sets the number of inner QS (see Table 1, remember that we only select well supported leaves as candidate QS) is very small; thus, this variation may be attributed to random effects. For the outer QS, the advantage over BLAST is less pronounced. For two of the larger data sets (D1604 and D755), the absolute accuracy of the EPA is approximately

equal for outer QS and inner QS, while there is a larger accuracy decrease for BLAST. The re-alignment using HMMER (see Table 2 columns EPA and EPA-RA) before placement by the EPA has only a small negative impact on placement accuracy. The re-alignment step decreases the accuracy of the EPA with respect to the two distance measures by factor 1.03–1.2. For one data set only (D628), the combination of EPA and HMMER was found to be slightly less accurate than placement with BLAST. We conclude that profile-HMMs as implemented in HMMER offer a useful method for the addition of short reads to a RA in this scenario, while there is still room to improve the QS alignment on certain data sets.

In Table 3 we show the placement precision of the EPA and BLAST, in terms of percentage of QS placed within a certain ND of their true position. Up to a ND threshold of 10, the EPA with HMMER re-alignment (column EPA-RA) outperforms BLAST by 2.2–8%. As before, the EPA without re-alignment has slightly higher accuracy compared to EPA-RA. For the inner QS subset, the difference between BLAST and EPA-RA is larger, reaching 19.7% for a ND threshold of 1 (by definition, BLAST cannot correctly place inner QS, thus ND will be at least 1). Brady and Salzberg (2009) evaluated Phymm/PhymmBL in a similar experimental setup, by measuring the classification accuracy of Phymm, PhymmBL and BLAST at different taxonomic levels. PhymmBL achieved an accuracy improvement of approximately 6% for PhymmBL over BLAST on simulated QS.

As previously noted here, there can be position specific effects that can influence the placement accuracy depending on which part of the gene

is used to generate QS. To minimize the influence of this position-specific bias, we sampled multiple QS (at different gene regions) from every input sequence and report averages in this study. This broad sample of QS along the gene can also be used to plot the site-specific, mean placement accuracy over the length of the RA. Accuracy plots for the data sets in this study are available in the supplementary material. This approach can also be used, a priori, on a full length sequence alignment to determine appropriate gene regions for short-read generation.

Placement Accuracy for paired-end Reads

Table 4 provides the overall results of the experiments with virtual paired-end reads of length 2x100bp (the results for 2x50bp reads are provided in the supplementary material). Similar to the previous section the placement accuracies are given in terms of ND and NED%, averaged over all QS. For D150, the EPA places the QS within 1.26 nodes of their true position while for BLAST, the average ND of 3.67 corresponds to placements that are 2.91 times further away from the original position, relative to the EPA. The table also contains corresponding values for the NED% measure, which indicate similar results as the ND measure. As in the previous section the general trend is similar for all data sets. The accuracy of the EPA placements are on average 1.58–5.87 times better than for BLAST.

Figure 7 provides histograms for the accuracy distribution of individual placements computed by the EPA and BLAST for 2x100bp paired-end reads on data set D855. Respective histograms for all data sets

on 2x100bp and 2x50bp reads are available in the supplementary material. The plots suggest that the placement error for both methods follows an approximate power law distribution. The placements obtained via the EPA are, on average, closer to the true position and yield smaller maximum placement errors than BLAST.

Table 4 highlights that the EPA consistently outperforms BLAST-based placements for paired-end reads and that placements are approximately twice as accurate on average. Generally, the placement accuracy for paired-end reads of lengths 2x100bp is better than was expected from the test with randomly selected contiguous subsequences of mean length 200bp in the previous section. One contributing factor is that many of the subsequences in the previous experiment were significantly shorter than 200bp, because we use normally distributed lengths. There also appears to exist a positive effect, generated by selecting subsequences from two distinct regions of the gene (the start and the end) from the original QS; in combination, phylogenetic information from both ends, may contain a stronger historical signal than a single, randomly selected subsequence.

Impact of Placement Algorithms and Substitution Models on Accuracy

All preceding computational experiments were carried out using the most thorough (*slow*) version of the EPA under the GTR+Γ and WAG+Γ (AA) models. In this mode, the EPA optimizes edge lengths via the Newton-Rhapson method for every possible insertion edge on the RT. As previously mentioned, we also devised a *fast* version of the EPA where this optimization is deactivated for QS insertions. These heuristics can speed up

the EPA by one order of magnitude, when a large amount of QS is being placed onto a RT. An additional speedup by a factor of 3 to 4 can be achieved by using the GTR+CAT or PROT+CAT approximations (Stamatakis, 2006a) of rate heterogeneity.

Figure 8 shows the impact of EPA heuristics and rate heterogeneity models on placement accuracy for all QS over all data sets (analogous plots for the individual data sets are available in the supplementary material). For the *slow* insertion method, there is practically no difference in placement accuracy between the Γ model and the CAT approximation. For the *fast* insertion method, there is a notable decrease in placement accuracy for the CAT as well as the Γ models. The decrease is more pronounced for the outer QS while for the inner QS the effect of using the *fast* insertion method is less pronounced. As already mentioned, correct placement of the inner QS is harder than placement of the outer QS, which have direct neighbors in the RT. The results of this experiment show that the *slow* version of the EPA which includes edge length optimization can produce better placement results than the *fast* version, especially when QS are placed on inner edges of the RT.

Heuristics for Slow Insertions

As shown in the previous section, the loss of accuracy induced by the *fast* insertion method is minimal. Nonetheless, a slight accuracy improvement can be attained by the *slow* insertion method, especially regarding the more precise edge length estimate at the insertion position that can be used for post-analysis purposes. Using the rapid insertion edge

pre-scoring heuristics already described, it is possible to accelerate the *slow* insertion algorithm with little to no impact on placement accuracy. Here, we evaluate the accuracy trade-offs associated with these heuristics. We also provide run-time measurements for the EPA and BLAST.

In contrast to the previous accuracy assessments, we do not test the placement of one QS at a time onto an existing RT from which the QS has been previously pruned. Instead, we randomly split the alignments into two subsets that each comprise 50% of the taxa. The first subset is used to infer a best-known ML tree with RAxML onto which the remaining taxa (of the second subset) are placed via the EPA. Here, we assume the *slow* EPA placements to be the true placements. In this experiment, we reduce the length of the QS to 50% non-gap characters. The non-gap characters are a contiguous sequence fragment that starts at the beginning of the respective sequence; that is, the QS represent roughly the first half of the gene.

Figure 9 shows the accuracy on the largest data set D1604 (placement of 802 QS onto a RT with 802 RS). The fraction of insertion edges considered for the slow insertion phase is controlled by the parameter fh. In the plot the accuracy of the heuristics for values of $fh = \frac{1}{4}, \frac{1}{8}, \frac{1}{16}, \frac{1}{32}, \frac{1}{64}, \frac{1}{128}, \frac{1}{256}$ is shown (i.e., on this dataset approx. 400, 200, 100, 50, 25, 12, 6 out of 1601 possible insertion edges are considered). For the lowest fh values, there is a visible decrease in placement accuracy (sharp rise of the curves on the left side of the plot). For 50 or more insertion edges, the accuracy remains virtually constant. The results suggest that, on this data set, it is sufficient to more thoroughly analyze only 50 out of 1601 ($fh = \frac{1}{32}$) candidate insertion edges proposed by the

heuristics to obtain the best possible accuracy (even for $fh = \frac{1}{64}$ there is only a very small deviation from the reference placements). Another important result is that the MP heuristics produce equally accurate placements as the ML heuristics, for all, except the smallest values of fh. We conclude that the MP heuristics with a parameter setting of $fh = 1/32$ (using the Γ model for *thorough* insertions) are sufficient for achieving placement accuracy comparable to the reference placement, but with computational requirements (290 seconds) that are in the same order of magnitude as a simple BLAST search (216 seconds) of the QS against the sequences in the RS. Re-alignment of the QS to the RA takes 224 seconds using HMMER on this data-set. Thus, the combined run time of HMMER and the EPA is approximately 2.5 times higher than a simple BLAST search, but still within the same order of magnitude.

The lowest run time (113 seconds) was achieved by using the CAT model for *slow* insertions, at the expense of a slight loss in accuracy (i.e., on average the ND increased by approximately 0.1). Based on the results in the previous section, we expect the accuracy difference between the CAT approximation and the Γ model of rate heterogeneity to be negligible in a real-world scenario.

The differences in accuracy between the *fast* and *slow* insertion methods as well as between the Γ and CAT models are generally larger than in the previous sections. This is not surprising, given that the setup of this experiment was not designed to measure the insertion accuracy relative to an assumed correct position, but the deviation between our best, yet slowest, method and less accurate, accelerated methods. Here, we do not

constrain the experiment to QS with high support values in the RT, but chose QS at random, which may introduce a certain loss of precision to this evaluation. In addition, the RT (comprising 50% of the taxa in the original RA) is smaller than in the previous evaluations and thus more sparsely sampled. Nonetheless, the deviation between the *fast* and *slow* EPA versions amounts to less than half a node on average and the general finding that *slow* insertions under CAT are more accurate than *fast* insertions under Γ is consistent with previous experiments.

EPA Web-Service

An alpha version of a Web-Server that offers the EPA algorithm is freely available at `http://i12k-exelixis3.informatik.tu-muenchen.de/raxml` and will be continuously developed and improved. The server runs on a dedicated machine with 24 AMD cores and 128GB of main memory.

Users can upload RTs and RAs and chose to align QS to the RA with HMMER or upload an alignment that already contains the QS. When the QS are to be aligned by the server, they can also be clustered using UCLUST (`http://www.drive5.com/usearch/`) prior to alignment with HMMER. The UCLUST option can be used to reduce the number of reads that will subsequently be placed and aligned. Finally, the Web-Server also offers a JAVA-based result visualization tool that uses the Archeopteryx framework (Han and Zmasek, 2009) and provides a simple visualization of the read distribution over the RT.

Conclusion

We have presented an accurate and scalable approach for tree-dependent sequence comparison and compared its accuracy to sequence comparison based methods. A tree-dependent approach has methodological advantages over standard, pair-wise, similarity-based comparative approaches and the EPA is freely available for download as open source code and as a web-service. We demonstrate that our approach may be substantially more accurate than standard techniques used to analyze, for example, microbial communities. More importantly, we demonstrate that achieving improved accuracy does not require longer inference times and that our approach is as fast as a simple BLAST-based search when using additional heuristics.

The EPA also is relatively straight-forward to parallelize by applying a multi-grain parallelization technique (Stamatakis et al., 2010). On a multi-core system with 32 cores and 64GB of main memory, we were able to place 100,627 QS in parallel into a RT with 4,874 taxa within 1.5 hours. The application of the EPA is not limited to molecular data only. Berger and Stamatakis (2010) have used the EPA for the placement of fossil taxa onto a (molecular) RT of extant species.

A major challenge that remains to be solved consists in aligning the QS to a given RA. Throughout this paper, we have assumed that such an alignment was given or that such an alignment can conveniently be generated by aligning the individual QS to a profile-HMM of the RA. Ideally, one would like to simultaneously place and align the QS to the

respective insertion edge. We are currently working on a tree-dependent alignment procedure for short read QS, which can be closely integrated with the placement process. Initial tests show that with tree-dependent re-alignment it is possible to achieve higher EPA placement accuracy than with re-alignment using profile-HMMs. Another challenge consists of dealing with the matter of assigning short read sequences to a reference alignment when there is compositional heterogeneity across the sequences. This is a particularily important issue if sequences have aquired the same nucleotide or amino acid composition independently. A final challenge consists in developing appropriate visualization tools and metrics for analyzing distributions of reads on trees that have been computed by the EPA or pplacer programs.

Acknowledgments

The authors would like to thank Rob Knight, Steven Kembel, Micah Hamady, Christian von Mering and Manuel Stark for useful discussions on algorithm design and for providing test data sets. We also thank F. Matsen for fruitful discussions on pplacer and EPA. We wish to acknowledge the constructive suggestions by an anonymous reviewer.

Bibliography

Ababneh, F., L. S. Jermiin, C. Ma, and J. Robinson. 2006. Matched-pairs tests of homogeneity with applications to homologous nucleotide sequences. Bioinf. 22:1225–1231.

Altschul, S., T. Madden, A. Schaffer, J. Zhang, Z. Zhang, W. Miller, and D. Lipman. 1997. Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. Nucl. Acids Res. 25:3389.

Berger, S. A. and A. Stamatakis. 2010. Accuracy of morphology-based phylogenetic fossil placement under maximum likelihood. *in* Proceedings of IEEE/ACS International Conference on Computer Systems and Applications (AICCSA-10); Hammamet, Tunisia, 2010.

Bininda-Emonds, O. R. P., S. G. Brady, M. J. Sanderson, and J. Kim. 2000. Scaling of accuracy in extremely large phylogenetic trees. Pages 547–558 *in* Pacific Symposium on Biocomputing 2001 (R. B. Altman, A. K. Dunker, L. Hunter, K. Lauderdale, and T. E. Klein eds.). World Scientific, River Edge, New Jersey.

Brady, A. and S. L. Salzberg. 2009. Phymm and PhymmBL: metagenomic

phylogenetic classification with interpolated Markov models. Nature Methods 6:673–676.

Chakravorty, S., D. Helb, M. Burday, N. Connell, and D. Alland. 2007. A detailed analysis of 16S ribosomal RNA gene segments for the diagnosis of pathogenic bacteria. J. Microbiol. Meth. 69:330–339.

DeSantis, T., P. Hugenholtz, K. Keller, E. Brodie, N. Larsen, Y. Piceno, R. Phan, and G. Andersen. 2006. NAST: a multiple sequence alignment server for comparative analysis of 16S rRNA genes. Nucleic Acids. Res. 34:W394–399.

Eddy, S. 1998. Profile hidden markov models. Bioinformatics 14:755–763.

Edgar, R. C. 2004. Muscle: multiple sequence alignment with high accuracy and high throughput. Nucl. Acids Res. 32:1792–1797.

Felsenstein, J. 1981. Evolutionary trees from DNA sequences: a maximum likelihood approach. J. Mol. Evol. 17:368–376.

Felsenstein, J. 1985. Confidence limits on phylogenies: An approach using the bootstrap. Evolution 39:783–791.

Fierer, N., M. Hamady, C. Lauber, and R. Knight. 2008. The influence of sex, handedness, and washing on the diversity of hand surface bacteria. Proc. Natl. Acad. Sci. USA 105:17994–17999.

Galtier, N. and M. Gouy. 1998. Inferring pattern and process: Maximum-likelihood implementation of a nonhomogeneous model of dna sequence evolution for phylogenetic analysis. Mol. Biol. Evol. 15:871–879.

Ganzert, L., G. Jurgens, U. Munster, and D. Wagner. 2007. Methanogenic communities in permafrost-affected soils of the laptev sea coast, siberian arctic, characterized by 16s rrna gene fingerprints. FEMS Microbiol. Ecol. 59:476–488.

Hamming, R. W. 1950. Error detecting and error correcting codes. Bell System Technical Journal 26:147–160.

Han, M. and C. Zmasek. 2009. phyloXML: XML for evolutionary biology and comparative genomics. BMC Bioinformatics 10:356.

Hillis, D. and J. Bull. 1993. An empirical test of bootstrapping as a method for assessing confidence in phylogenetic analysis. Syst. Biol. 42:182.

Ho, J. W. K., C. E. Adams, J. B. Lew, T. J. Matthews, C. C. Ng, A. Shahabi-Sirjani, L. H. Tan, Y. Zhao, S. Easteal, S. R. Wilson, and L. S. Jermiin. 2006. SeqVis: Visualization of compositional heterogeneity in large alignments of nucleotides. Bioinf. 22:2162–2163.

Hudson, D., A. Auch, J. Qi, and S. Schuster. 2007. MEGAN analysis of metagenomic data. Genome Res. 17:377–386.

Jayaswal, V., J. Robinson, and L. S. Jermiin. 2007. Estimation of phylogeny and invariant sites under the general markov model of nucleotide sequence evolution. Syst. Biol. 56:155–162.

Jermiin, L. S., S. Y. W. Ho, F. Ababneh, J. Robinson, and A. W. D. Larkum. 2004. The biasing effect of compositional heterogeneity on phylogenetic estimates may be underestimated. Syst. Biol. 53:638–643.

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

Kluge, A. and J. Farris. 1969. Quantitative phyletics and the evolution of anurans. Syst. Zool. 18:1–32.

Koski, L. and G. Golding. 2001. The closest BLAST hit is often not the nearest neighbor. J. Mol. Evol. 52:540–542.

Ley, R. E., F. Bäckhed, P. Turnbaugh, C. A. Lozupone, R. D. Knight, and J. I. Gordon. 2005. Obesity alters gut microbial ecology. Proc. Natl. Acad. Sci. USA 102:11070–11075.

Ley, R. E., J. K. Harris, J. Wilcox, J. R. Spear, S. R. Miller, B. M. Bebout, J. A. Maresca, D. A. Bryant, M. L. Sogin, and N. R. Pace. 2006. Unexpected diversity and complexity of the guerrero negro hypersaline microbial mat. Appl. Environ. Microbiol. 72:3685–3695.

Ley, R. E., C. A. Lozupone, M. Hamady, R. D. Knight, and J. I. Gordon. 2008. Worlds within worlds: evolution of the vertebrate gut microbiota. Nat. Rev. Microbiol. 6:776–788.

Lozupone, C. and R. Knight. 2005. UniFrac: a new phylogenetic method for comparing microbial communities. Appl. Environ. Microbiol. 71:8228–8235.

Ludwig, W., O. Strunk, R. Westram, L. Richter, H. Meier, Yadhukumar, A. Buchner, T. Lai, S. Steppi, G. Jobb, W. Forster, I. Brettske,

S. Gerber, A. W. Ginhart, O. Gross, S. Grumann, S. Hermann, R. Jost, A. Konig, T. Liss, R. Lussmann, M. May, B. Nonhoff, B. Reichel, R. Strehlow, A. Stamatakis, N. Stuckmann, A. Vilbig, M. Lenke, T. Ludwig, A. Bode, and K.-H. Schleifer. 2004. Arb: a software environment for sequence data. Nucl. Acids Res. 32:1363–1371.

Matsen, F., R. Kodner, and E. V. Armbrust. 2010. pplacer: linear time maximum-likelihood and bayesian phylogenetic placement of sequences onto a fixed reference tree. BMC Bioinformatics 11:538.

McHardy, A. C., H. G. Martin, T. Aristotelis, P. Hugenholtz, and I. Rigoutsos. 2007. Accurate phylogenetic classification of variable-length DNA fragments. Nature Methods 4:63–72.

Moret, B. M. E., U. Roshan, and T. Warnow. 2002. Sequence-length requirements for phylogenetic methods. Pages 343–356 *in* Proceedings of 2nd Intl Workshop on Algorithms in Bioinformatics (WABI 02) 2002 (R. Guigo and D. Gusfield eds.). Lecture Notes in Computer Science 2452, Springer.

Munch, K., W. Boomsma, J. P. Huelsenbeck, E. Willerslev, and R. Nielsen. 2008. Statistical assignment of dna sequences using bayesian phylogenetics. Syst Biol 57:750–757.

Nielsen, R. and M. Matz. 2006. Statistical approaches for DNA barcoding. Syst. Biol. 55:162–169.

Pruitt, K., T. Tatusova, and D. Maglott. 2007. NCBI reference sequences

(RefSeq): a curated non-redundant sequence database of genomes, tanscrips and proteins. Nucleic Acids. Res. 35 Database issue:D61–D65.

Ronaghi, M. 2001. Pyrosequencing sheds light on DNA sequencing. Genome Res. 11:3–11.

Stamatakis, A. 2006a. Phylogenetic models of rate heterogeneity: A high performance computing perspective. *in* Proceedings of 20th IEEE/ACM International Parallel and Distributed Processing Symposium (IPDPS2006); Rhodos, Greece. 2006.

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

Stamatakis, A., P. Hoover, and J. Rougemont. 2008. A Rapid Bootstrap Algorithm for the RAxML Web Servers. Syst. Biol. 57:758–771.

Stamatakis, A., Z. Komornik, and S. A. Berger. 2010. Evolutionary placement of short sequence reads on multi-core architectures. *in* Proceedings of IEEE/ACS International Conference on Computer Systems and Applications (AICCSA-10); Hammamet, Tunisia, 2010.

Stamatakis, A., T. Ludwig, and H. Meier. 2005. RAxML-III: A fast program for maximum likelihood-based inference of large phylogenetic trees. Bioinformatics 21(4):456–463.

Strimmer, K. and A. Rambaut. 2002. Inferring confidence sets of possibly

misspecified gene trees. Proceedings of the Royal Society B: Biological Sciences 269:137–142.

Turnbaugh, P., M. Hamady, T. Yatsunenko, B. Cantarel, A. Duncan, R. Ley, M. Sogin, W. Jones, B. Roe, J. Affourtit, et al. 2008. A core gut microbiome in obese and lean twins. Nature 457:480–484.

von Mering, C., P. Hugenholtz, J. Raes, S. G. Tringe, T. Doerks, L. J. Jensen, N. Ward, and P. Bork. 2007. Quantitative phylogenetic assessment of microbial communities in diverse environments. Science 315:1126–1130.

Table 1: Data sets used for evaluation of the Evolutionary Placement Algorithm (EPA). The columns contain (from left to right): the name of the data; the type of the data (N: nucleotides; AA: amino acids); the length of the data (i.e., number of sites in the alignment); the number of taxa (# taxa); the number of query sequences (# QS); and the number of inner query sequences (# inner OS) (for definitions of QS and inner QS, see the main text).

Data	type	length	# taxa	# QS	# inner QS
D140	AA	1104	140	95	9
D150	N	1269	150	66	10
D218	N	2294	218	80	14
D500	N	1398	500	205	29
D628	N	1228	628	210	20
D714	N	1241	714	293	61
D855	N	1436	855	344	48
D1604	N	1276	1604	541	83

Table 2: Accuracy on randomly sampled short subsequences in terms of ND from the original position. The second column (EPA) shows the average ND of the EPA placements (using *slow* insertions under the $GTR + \Gamma$ model) for the data sets in question. The third column (EPA-RA) shows the average ND for the EPA with previous re-alignment using HMMER. The last column (BLAST) shows the average ND for a BLAST-based approach.

	Data	EPA	EPA-RA	BLAST
outer QS	D140	0.49	0.58	0.82
	D150	1.14	1.2	1.96
	D218	1.7	2.01	3.66
	D500	1.31	1.37	2.36
	D628	2.44	2.95	2.69
	D714	1.71	1.82	2.36
	D855	2.87	2.97	3.53
	D1604	2.26	2.45	2.87
inner QS	D140	0.74	0.71	1.81
	D150	3.09	3.1	4.62
	D218	2.24	2.66	3.81
	D500	2.05	2.37	4.16
	D628	3.28	3.65	4.04
	D714	1.78	1.93	3.51
	D855	3.68	3.74	4.91
	D1604	2.23	2.38	3.86

Table 3: Percentage of QS placed correctly within a certain node distance (max(ND)) of the original position over all data sets.

	max(ND)	EPA	EPA-RA	BLAST
outer QS	0	58.4%	56.4%	56.3%
	1	71.1%	69.5%	64.1%
	2	77.6%	76.0%	69.5%
	5	87.9%	86.8%	82.6%
	10	94.6%	93.9%	91.9%
inner QS	0	37.0%	35.1%	0.0%
	1	64.3%	63.4%	43.7%
	2	74.7%	73.1%	56.9%
	5	86.5%	85.8%	77.9%
	10	94.1%	93.6%	90.5%

Table 4: Accuracy of the placement of 2x100 bp paired-end reads. The values given are the node-distance (ND) and the normalized edge distance (NED %). The methods used are the Evolutionary Placement Algorithm (EPA) (*slow* insertions under the $GTR + \Gamma$ model) and BLAST-based nearest neighbor.

		ND		NED %	
	Data	EPA	BLAST	EPA	BLAST
outer QS	D150	1.14	3.38	0.63	1.76
	D218	1.44	4.77	3.6	8.45
	D500	0.84	4.88	1.75	7.35
	D628	0.56	1.83	0.83	1.83
	D714	1.31	3.59	1.77	4.78
	D855	2.0	5.98	1.11	3.35
	D1604	1.58	3.67	0.79	1.65
inner QS	D150	1.9	5.3	2.23	5.64
	D218	4.43	5.21	8.16	9.38
	D500	1.59	9.41	3.15	12.24
	D628	0.85	2.95	0.55	1.13
	D714	1.66	4.9	2.9	7.76
	D855	2.9	7.54	2.04	4.61
	D1604	1.57	5.3	1.69	4.84

Figure 1: Local optimization of edge lengths for the insertion of a query sequences (QS) into the reference tree (RT).

Figure 2: Evolutionary identification of 3 query sequences (QS_0, QS_1, QS_2) using a 4-taxon reference tree.

Figure 3: Phylogenetic placement of 3 query sequences (QS_0, QS_1, QS_2) onto a 4-taxon reference tree with insertion support (IS) score.

Figure 4: Illustration of the criterion for the query sequence (QS) selection and experimental setup. (a) Candidate QS belongs to sub tree of size 2 that is connected to the tree by a well supported edge. It has one other tip as direct neighbor. QS with this property will be referred to as outer QS. (b) Candidate QS is connected to the tree by two well supported edges. QS with this property will be referred to as inner QS. (c) Experimental setting: re-insert shortened candidate QS in to pruned reference tree. We use three different ways of generating QS with desirable features: contiguous subsequences, paired-end reads and random gaps.

Figure 5: Illustration of the tree-based distance measures. (a) Example tree with two edges (original and insertion edge) highlighted. There are two nodes on the path, so the node distance is 2. The edge distance corresponds to the length of the connecting path, where of the two end edges only half of the edge length is used. (b) Tree diameter which is used to normalize the edge distance.

Figure 6: Placement accuracy for QS with artificially introduced random gaps. (a) Average node distance and (b) normalized edge distance (between insertion positions and real positions).

Figure 7: Histogram showing the placement accuracy, based on node distances, for the placement of 2x100 bp paired-end reads from D855, using outer (a) and inner (b) QS.

Figure 8: Average node distance for different versions of the EPA (*fast*/*slow* insertions) algorithm and model types (GTR+Γ, GTR+CAT) on inner QS and outer QS from all data sets.

Figure 9: Accuracy of the EPA as a function of the number of edges considered for *slow* insertion after heuristic filtering.