A rapid phylogeny-based method for accurate community profiling of large-scale metabarcoding datasets

Lenore Pipes[1] & Rasmus Nielsen[1,2]

[1]*Department of Integrative Biology, University of California-Berkeley, Berkeley, California, USA*

[2]*GLOBE Institute, University of Copenhagen, Copenhagen, Denmark*

Environmental DNA (eDNA) is becoming an increasingly important tool in diverse scientific fields from ecological biomonitoring to wastewater surveillance of viruses. The fundamental challenge in eDNA analyses has been the bioinformatical assignment of reads to taxonomic groups. It has long been known that full probabilistic methods for phylogenetic assignment are preferable, but unfortunately, such methods are computationally intensive and are typically inapplicable to modern Next-Generation Sequencing data. We here present a fast approximate likelihood method for phylogenetic assignment of DNA sequences. Applying the new method to several mock communities and simulated datasets, we show that it identifies more reads at both high and low taxonomic levels more accurately than other leading methods. The advantage of the method is particularly apparent in the presence of polymorphisms and/or sequencing errors and when the true species is not represented in the reference database.

In the past ten years, metabarcoding and metagenomics based on DNA sequencing and subsequent taxonomic assignment, have become an important approach for understanding diversity and community organization at many taxonomic levels. This has led to the publication of over

80 taxonomic classification methods[1]. There are three major strategies in classification methods: composition-based, which do not align sequences but extract compositional features (e.g., kmers) to build models of probabilistic taxonomic inclusion, alignment-based, which rely on alignments to directly compare query sequences to reference sequences but do not use trees, and phylogenetic-based, which rely on a phylogenetic tree reconstruction method, in addition to alignments, to perform a placement of the query onto the tree. As a trade-off between speed and precision for processing Next-Generation Sequencing (NGS) data, the vast majority of recent classification methods have either relied on alignment-based or composition-based strategies.

Composition-based tools reduce the reference database by indexing compositional features such as kmers for a rapid search of the database. These methods require an exact match between the kmer in the query sequence and the kmer in the reference database. As a result of hash indexing of kmers, kraken2[2], for example, can classify $>$1 million reads within 1 minute using the entire Geengenes or SILVA databases[3]. Alignment-based tools use a fast local aligner such as BLAST[4] to pairwise align queries to the reference database, and define a score based on sequence similarity in the alignment between the read and reference sequence. However, alignment-based methods can be many orders of magnitude slower than composition-based tools since datasets with $>$10 million reads require weeks of BLASTN running time[5]. In both composition-based tools and alignment-based tools, a lowest common ancestor (LCA) algorithm is then typically used to assign at different taxonomic levels (**Figure 1A**). LCA works by assigning to the smallest possible clade that include all matches with a similarity less than the specified cut-off.

Phylogenetic placement methods place a query sequence onto a phylogenetic tree of refer-

ence sequences. This placement requires a full multiple sequence alignment (MSA) of the reference sequences and a subsequent estimation of a phylogenetic tree. However, large datasets with high rates of evolution are hard to align accurately[6] and phylogeny estimation methods produce poor trees when MSAs are not of high quality[7]. Futhermore, phylogentic placement tends to be computationally demanding as both running time and memory usage scale linearly with the size of the reference database [8]. Even for reference databases that contain sequences as few as 1,600 sequences, assignment for a single query using the most cited phylogenetic placement method, *pplacer*[9], takes more than 7 minutes and requires over 10GB of RAM. At this rate, a reference database that contains a metabarcode such as Cytochrome oxidase 1 (COI) that has at least 1.5 million reference sequences, assigning just a single query would require 20.9 hours and 2.37TB RAM. Scaling the query size to millions of queries would therefore be computationally intractable.

To address these challenges, the most recent implementations of phylogeneny-based methods[10] rely on reference database reduction techniques (i.e., using only representative taxa or consensus sequences for a sparse backbone tree) to handle the large amount of data that is routinely produced. Often a single species is selected to represent an entire clade[11]. While this reduces the computational cost, it also reduces the granularity, and potentially the accuracy, of the assignments. As a trade-off between speed and precision, the vast majority of recent classification methods are either alignment-based or composition-based approaches[12] since phylogeny-based methods have not scaled to handle the entirety of the rapidly growing reference databases of genome markers and the increasingly large amounts of NGS data.

Here we describe a new method for phylogenetic placement, implemented in the program

'Tronko' (https://github.com/lpipes/tronko, **Supplementary Software**), the first phylogeny-based taxonomic classification method designed to truly enable the use of modern-day reference databases and NGS data. The method is based on approximating the phylogenetic likelihood calculation by (1) only allowing the edge connecting the reference sequence to the tree to join at existing nodes in the tree and then (2) approximating the likelihood using a probabilistically weighted mismatch score based on pre-calculated fractional likelihoods stored in each node (Online methods). We argue that (2) approximates the full maximized likelihood assignment without requiring any numerical maximization under the approximating assumption that the read joins the tree in an existing node with a zero length branch. The approximation is equivalent to calculating the expected average mismatch to each node in the phylogeny. The assignment method in Tronko uses the LCA criteria but, unlike composition-based and alignment-based approaches (**Figure 1A**), takes advantage of fractional likelihoods stored in all nodes of the tree with a cut-off that can be adjusted from conservative to aggressive (**Figure 1B**). In the simplest case, when the reference sequences form a single tree, Tronko uses a pre-calculated MSA, the phylogenetic tree based on the MSA, and pre-calculated posterior probabilities, which are proportional to the fractional likelihoods. However, in more typical cases, when a single tree/MSA is unsuitable for analyses, as the reference sequences encompass increasingly divergent species as well as an increasing volume of sequences, we present a fully customizable divide-and-conquer method for reference database construction that is based on dividing reference sequences into phylogenetic subsets that are re-aligned and with local trees re-estimated.

The construction of the database, MSAs, and trees, facilitates fast phylogenetic assignment.

The assignment algorithm then proceeds by (1) A BWA-MEM [13] search on all sequences in the database, (2) a pairwise sequence alignment between the query and the top hit in each alignment-subset containing a BWA-MEM hit using either the Needleman-Wunsch algorithm[14] or the Wave-front Alignment algorithm[15], and (3) a calculation of a score based on the approximate likelihood for each node in subsets with a BWA-MEM hit. An additional LCA assignment for all subsets can then be applied to summarize the results. For full details, please see ONLINE METHODS.

RESULTS

To compare the new method (Tronko) to previous methods, we constructed reference databases for COI and 16S for common amplicon primer sets. We first compared Tronko to pplacer for reference databases containing a reduced amount of sequences ($<$1,600 sequences) to compare the speed and memory requirements with a comparable phylogenetic-based assignment method. Tronko shows speed-ups $>$20 times, with a vastly reduced memory requirement illustrating the computational advantage of the approximations in Tronko (**Supplementary Fig. S4**).

Next, we evaluated Tronko's performance to kmer-based kraken2[2] which previously has been argued to have the lowest false-positive rate[3], and two other popular alignment-based methods: MEGAN[16] and metaphlan2[17]. We did not compare to pplacer because it would require too many computational resources to do so. We used two types of cross validation tests: leave-one-species-out and leave-one-individual-out analyses. The leave-one-species-out test involves removing an entire species from the reference database, simulating next generation sequencing reads from that species, and then attempting to assign those reads with that species missing from the database. The leave-one-individual-out test involves removing a single individual from the reference database,

simulating next generation sequencing reads from that individual, and then attempting to assign those reads with that individual missing from the database. In both tests, singletons (i.e., cases in which only one species was present in a genera or cases in which only one individual represented a species) were exempt from the tests.

We performed a leave-one-species-out test comparing Tronko (with LCA cut-offs for the score of 0, 5, 10, 15, and 20 with both Needleman-Wunsch alignment and Wavefront alignment) to kraken2, metaphlan2, and MEGAN for 1,467 COI sequences from 253 species from the order Charadriiformes using 150bp x 2 paired-end sequences and 150bp and 300bp single-end sequences using 0, 1, and 2% error/polymorphism (**Figures 2 and 3**). See **Figure S2** for results with Wavefront alignment.

Using leave-one-species-out and simulating reads (both paired-end and single-end) with a 0-2% error (or polymorphism), Tronko detected the correct genus more accurately than the other methods even when using an aggressive cut-off (i.e., when cut-off=0) (**Figure 3D and G**). Using 150bp paired-end reads with 1% error, Tronko had a misclassification rate of only 9.045% with a recall rate of 71.381% at the genus level using a cut-off set to 15 while kraken2, MEGAN, and metaphlan2 had misclassification rates of 33.475%, 10.046%, and 27.731%, respectively, with recall rates of 90.555%, 52.136%, and 95.027% (see **Figure 2B**). Tronko had a lower misclassification rate relative to the recall rate out of all methods for 150bp x 2 paired-end reads with 0% error/polymorphism (**Figure 2A**), 1% error/polymorphism (**Figure 2B**), and 2% error/polymorphism (**Figure 2** and **Figure 3D-I**), for 150bp reads with 0% error/polymorphism (**Figure 2D**), 1% error/polymorphism (**Figure 2E**), and 2% error/polymorphism (**Figure 2F**), and for 300bp reads

with 0% error/polymorphism (**Figure 2G**), 1% error/polymorphism (**Figure 2H**), and 2% error/polymorphism (**Figure 2I**). See Methods for definitions of recall and misclassification rates. Tronko also accurately assigned genera from the Scolopacidae family (top left of matrices in **Figure 3**) using Needleman-Wunsch with a cut-off of 10 compared to kraken2 and metaphlan2.

Next, we performed a leave-one-individual-out test for the same COI sequences (**Figures 4 and 3G-L**). See **Figure S3** for results with Wavefront alignment. Using single-end reads of lengths 150bp and 300bp, Tronko has a lower misclassification rate and higher recall rate than kraken2, metaphlan2, and MEGAN. Using 150bp paired-end reads with 0% error (**Figure 4D**), Tronko had a misclassification rate at only 0.258% with a recall rate of 65.110% at the species level using a cut-off set to 15 while kraken2, MEGAN, and metaphlan2 had misclassification rates of 1.240%, 0.313%, and 11.181%, respectively, with recall rates of 81.717%, 65.668%, and 99.839%. Both metaphlan and kraken2 have a number of mis-assignments within the family of Laridae (see blue points across the diagonal in **Figure 3G** and **H**) and Tronko is able to accurately assign species within this family or assign at the genus or family level.

We then compared Tronko's performance to kraken2, MEGAN, and metaphlan2 using mock communities for both 16S[18,19] and COI markers[20] (**Figure 5**). For 16S, we used two different mock community datasets. We used 2 x 300bp Illumina MiSeq sequencing data from a mock community consisting of 49 bacteria and 10 archaea species from Schirmer *et al*. (2015)[18] and 2 x 300bp Illumina MiSeq sequencing data from a mock community of 20 evenly distributed bacterial species from Gohl *et al* (2016)[19]. For the data from Schirmer *et al.* (2015), at the species level, Tronko had a less than 0.6% misclassification rate at every cut-off with a recall rate of 11.020%

at cut-off 0 (**Figure 5A**; See **Figure S6** for plot without outliers). kraken2 had a misclassification rate of 1.242% with a recall rate of 10.569% when using its default database, and a misclassification rate of 3.542% and a recall rate of 35.110% when using the same reference sequences as Tronko. metaphlan2 did not have any assignments at the species, genus, or family level using the default database, and it had an 8.334% misclassification and 8.943% recall rate at the species level when using the same reference sequences as Tronko. MEGAN had a recall rate of 0.206% and a misclassification rate of 0% at the species level.

For the data from Gohl *et al.* (2016), at the species level, Tronko had a less than 2.6% misclassification rate at every cut-off with a recall rate of 12.815% at cut-off 0 (**Figure 5B**; See **Figure S7** for plot without outliers). kraken2 had a misclassification rate of 26.812% and recall rate of 33.694% when using its default database, and a misclassification rate of 21.409% and recall rate of 25.405% when using the same reference sequences as Tronko. metaphlan2 did not have any assignments at the species, genus, or family level using the default database, and it had an 8.470% misclassification and 2.073% recall rate at the species level when using the same reference sequences as Tronko. MEGAN had a misclassification rate of 0.00629% and a recall rate of 4.439% at the species level.

For COI, we used a dataset from Braukmann et al. (2019)[20] which consists of 2 x 300bp Illumina MiSeq sequencing data from 374 species of terrestrial arthropods, which is the most expansive mock community dataset that we used. At the genus level, Tronko had a misclassification of less than 0.6% with a recall rate of 91.306% at the cut-off of 0 (**Figure 5C**; see **Figure S8** for plot without outliers). With the default database, kraken2 had a misclassification rate of 40.537%

with a recall rate of 6.504%. With the same reference sequences as Tronko, kraken2 still had a misclassification of 13.998% with a recall rate of 83.141%. metaphlan2 had a misclassification rate of 3.538% with a recall of 86.487% with the same reference sequences as Tronko while the default database failed to assign any reads. MEGAN did not assign any reads at the species or genus level.

We compared Tronko with kraken2, metaphlan2, and MEGAN (using BLAST as the aligner) for running time (**Figure 6**A) and peak memory (**Figure 6B**) using 100, 1,000, 10,000, 100,000, and 1,000,000 sequences using the COI reference database. Unsurprisingly, kraken2 had the fastest running time followed by metaphlan2, but MEGAN had a substantially slower running time than all methods. Tronko was able to assign 1,000,000 queries in $\sim$8 hours with the choice of aligner being negligible. Tronko had the highest peak memory ($\sim$50GBs) as it stores all reference sequences, their trees, and their posterior probabilities in memory. We note that for very large databases, the memory requirements can, in theory, be reduced by processing different alignment subsets sequentially.

Discussion

Both leave-one-species out and leave-one-individual-out simulations show that Tronko recovers the correct taxonomy with higher probability than competing methods and represents a substantial improvement over current assignment methods. The advantage of Tronko comes from the use of limited full sequence alignments and the use of phylogenetic assignment based on a fast approximation to the likelihood.

We evaluate Tronko using diferent cut-offs representing different trade-offs between recall

and misclassification rate, thereby providing some guidance to users for choice of cut-off. We note that in most cases, the other methods evaluated here fall within the convex hull of Tronko, showing that Tronko dominates those methods, and in no cases do other methods fall above the convex hull of Tronko. However, in some cases other methods are so conservative, or anti-conservative, that a direct comparison is difficult. For example, when using single-end 300bp reads (**Figure 4G-I**), MEGAN has assignment rates that are so low that a direct comparison is difficult.

Among the methods compared here, kraken2 is clearly the fastest (**Figure 6A**). However, it generally also has the worst performance with a higher misclassification rate than other methods, especially in the leave-one-species out simulations (**Figure 2**).

Both metaphlan2 and MEGAN tend fall within the convex hull of Tronko. Typically, metaphlan2 assigns much more aggressively, and therefore, has both a recall and misclassification rate that is much higher than MEGAN, which assigns very conservatively. We also note that the computational speed of MEGAN is so low that it, in some applications, may be prohibitive (**Figure 6A**).

We evaluated Tronko using two different alignment methods, Needleman-Wunsch and Wavefront Alignment. In many cases, the two alignment algorithms perform similarly. However, in the case, where short, single-end reads are used (i.e., 150bp single-end reads), the Wavefront Alignment performs worse than the Needleman-Wunsch Alignment (see Figures S2D-F and S3D-F). The Wavefront Alignment algorithm implements heuristic modes to accelerate the alignment, which performs similar to Needleman-Wunsch when the two sequences being aligned are similar in length. However, when there is a large difference between the two sequences being aligned, we notice that the Wavefront Alignment forces an end-to-end alignment which contains large gaps at the begin-

ning and end of the alignment. Hence, based on current implementations, we cannot recommend the use of the Wavefront Alignment for assignment purposes of short reads, although this conclusion could change with future improvements of the implementation of the wavefront alignment algorithm.

Tronko is currently not applicable to eukaryotic genomic data as it requires well-curated alignments of markers and associated phylogenetic trees, although we note that whole-genome phylogenetic reference databases for such data could potentially be constructed. Such extensions of the use of Tronko would require heuristics for addressing the memory requirements. Tronko currently has larger memory requirements than methods that are not phylogeny-based. Nonetheless, for assignment to viruses, amplicon sequencing and other forms of non-genomic barcoding, Tronko provides a substantial improvement over existing assignment methods and is the first full phylogenetic assignment method applicable to modern large data sets generated using Next Generation Sequencing.

The methods presented in this paper are implemented in the Tronko software package that includes Tronko-build and Tronko-assign for reference database building and species assignment, respectfully. Tronko can be downloaded at http://www.github.com/lpipes/Tronko and is available under an open-software license.

Methods

Tronko-build reference database construction with a single tree

The algorithm used for assignment takes advantage of pre-calculated posterior probabilities of nu-

cleotides at internal nodes of a phylogeny. We first estimate the topology and branch-lenghts of the tree using RAxML[21], although users of the method could use any tree estimation algorithm. We then calculate and store the posterior probabilities of each nucleotide in each node of the tree. For computational efficiency, this is done under a Jukes and Cantor (1969) model[22], but the method can easily be extended to other models of molecular evolution. The calculations are achieved using an algorithm that traverses the tree only twice to calculate posterior probabilities simultaneously for all nodes in the tree. In brief, fractional likelihoods are first calculated in each node using a standard postorder traversal (e.g. Felsenstein 1981[23]). This directly provides the posterior probabilities in the root after appropriate standardization. An inorder traversal of the tree is then used to pull fractional likelihoods from the child nodes of the root down the tree and, at each node, simultaneously calculate fractional likelihoods by multiplying appropriate products of transition probabilities and fractional likelihoods from the child nodes with products of transition probabilities and fractional likelihoods from the parent node. While naive application of standard algorithms for calculating posterior probabilities in a node, to all nodes of a tree, have computational complexity that is quadratic in the number of nodes, the algorithm used here is linear in the number of nodes. The algorithm is implemented in the program 'Tronko-build'.

Each node in the tree is subsequently provided a taxonomy assignment. This is done by first making taxonomic assignments of the leaf nodes using the taxonomy provided by the taxid of the associated NCBI accession. We then make taxonomic assignments for internal nodes, at all taxonomic levels (species, genus, etc), using a postorder traversal of the tree that assigns a taxonomic descriptor to node i if both children of node i have the same taxonomic assignment. Otherwise,

node i does not have a taxonomic assignment at this taxonomic level. In other words, node i only gets a taxonomic assignment if the taxonomic assignments of both child nodes agree.

Tronko-build reference database construction with multiple trees

MSAs for a large number of sequences can become unreliable, and computationally challenging to work with, due to the large number of insertions and deletions. For that reason, we devise an algorithm for partitioning of sequence sets into smaller subsets based on the accuracy of the alignment and using the inferred phylogenetic tree to guide the partitioning (**Figure S1**).

To measure the integrity of the MSA we calculate an average quality score, sum-of-pairs, ASP, which is a sum of pairwise alignment scores in the MSA. Assume a multiple sequence alignment of length l with K sequences, $A = \{a_{i,j}\}$, where $a_{i,j}$ is the jth nucleotide in sequence i, $1 \leq i \leq K$, $1 \leq j \leq l$, $a_{i,j} \in M = \{-, A, C, T, G, N\}$. Define the penalty function, p:

$$p(I,V) = \begin{cases} 3 & \text{if } I = V \text{ and } I \neq - \text{ (match)} \\ -2 & \text{if } I \neq V,\ I, V \notin \{N, -\} \text{ (mismatch)} \\ -1 & \text{otherwise} \end{cases} \tag{1}$$

where $I, V \in M$. ASP is then calculated as

$$ASP = \frac{\sum_{j=1}^{l} \sum_{i=1}^{K} \sum_{k=i+1}^{K} p(a_{i,j}, a_{k,j})}{\binom{K}{2}} \tag{2}$$

If the ASP is lower than the ASP threshold (a threshold of 0.1 was used in our analyses in this manuscript), the corresponding tree is split in three partitions at the node with the minimum variance, calculated as:

$$v = \underset{i \in T}{\operatorname{argmin}} \left\{ ((L_1(i) - K/3)^2 + (L_2(i) - K/3)^2 + (K - L_1(i) - L_2(i) - K/3)^2) \right\} \tag{3}$$

where T is a tree, i.e. a set of nodes, $L_1(i)$ and $L_2(i)$ is the number of leaf nodes descending from the left and right child node, respectively, of node i, and K is the total number of leaf nodes in the tree. We then split the tree into 3 subtrees by eliminating node v. Each partition is re-aligned with `FAMSA`[24] and new trees are constructed using `RAxML`[21] using default parameters and the GTR+Gamma model. The sequences are recursively partitioned until the ASP score is above the threshold. Finally, the trees, multiple sequence alignments, taxonomic information, and posterior probabilities are printed to one reference file which can be loaded for subsequent assignment of reads. Notice, that the procedure for phylogeny estimating and calculation of posterior probabilities only has to be done once for a marker and then can be used repeatedly for assignment using different data sets of query sequences.

Taxonomic classification of query sequences

First, `BWA-MEM`[25] is used with default options to align the query sequences to the reference sequences, thereby identifying a list of the highest scoring reference sequences (which we designate as BWA-MEM hits) from the reference database. Second, a global alignment, either using the Needleman-Wunsch algorithm[14] or the Wavefront alignment algorithm[15], is performed only on the sequence with the highest score from each subtree (reference sequence set) identified using the previously described partitioning algorithm.

Once aligned to the reference sequence, a score, $S(i)$ is calculated for all nodes, i, in the tree(s) that the reference sequence is located to. For a given read, let b_j be the observed nucleotide in the position of the read mapping to position j in the alignment. We also assume an error rate, c. For example, if the true base is G and the error rate is c, then the probability of observing A in

the read is $c/3$. We note that this error rate can be consider to include both true sequencing errors and polymorphisms/sequence divergence. In an ungapped alignment, the score for site j in node i is then the negative log of a function that depends on the posterior probability of the observed nucleotide in the query sequence, $PP_{ij}(b_j)$, and the error rate:

$$-log(c/3 + (1 - 4c/3)PP_{ij}(b_j)) \tag{4}$$

Assuming symmetric error rates, the probability of observing the base by error is $(1-PP_{ij}(b_j))c/3$ and the probability of observing the base with no error is $(1 - c)PP_{ij}(b_j)$. The sum of these two expressions equals the expression in the logarithm above. The score for all s sites in the read is defined as $-\sum_{j=1}^{s} log(c/3 + (1 - 4c/3)PP_{ij}(b_j))$.

Notice that the full phylogenetic likelihood for the entire tree, under standard models of molecular evolution with equal base frequencies and not accounting for errors is $\ell = \sum_{j=1}^{s} log(\sum_{v\in\{A,C,T,G\}} PP_{ij}(v)P_{vb_j}(t))$ where $P_{vb_j}(t)$ is the time dependent transition probability from base v to base b_j in time t. This statement takes advantage of the fact that, under time-reversibility, the posterior for a base in an node is proportional to the fractional likelihood of that base in the node, if the tree is rooted in the node. For small values of t, ℓ converges to $log(PP_{ij}(b_j)$. Minimizing the score function, therefore, corresponds to maximizing the full phylogenetic likelihood function assuming that the branch leading to the query sequence is infinitesimally short and connects with the tree in an existing node. An alternative interpretation is that the score maximizes the probability of observing the query sequence if it is placed exactly in a node or, equivalently, minimizes the expected mismatch between the query and a predicted sequence sampled form the node.

To address insertions and deletions, we define scores of γ and λ for a gap or insertion, respec-

tively, in the query sequence relative to the reference sequence. We also entertain the possibility of a gap in the reference sequence in node i in read position j, r_{ij}, which occurs when the reference is a leaf node with a gap in the position or if it is an internal node with all descendent nodes having gaps in the position. We use the notation $M_g = \{-, N\}$ for gaps and $M_n = \{A, C, T, G\}$ for nucleotides (no gap). Then, the score for node i in site j of the read, with observed base b_j, is

$$S_j(i) = \begin{cases} c/3 + (1 - 4c/3)PP_i(b_j) & \text{if } b_j \in M_n \text{ and } r_{ij} \in M_n \\ \gamma & \text{if } b_j \in M_g \text{ and } r_{ij} \in M_n \\ 1 & \text{if } b_j \in M_g \text{ and } r_{ij} \in M_g \\ \lambda & \text{if } b_j \in M_n \text{ and } r_{ij} \in M_g \end{cases} \tag{5}$$

The total score for the entire read is

$$S(i) = \sum_{j=m(1)}^{l} log(S_j(i)) \tag{6}$$

For paired reads, the scores for each node in the tree is calculated as the sum of the scores for the forward read and the scores for the reverse read. Scores are calculated for all nodes in each tree that contain a best hits from the `bwa mem` alignment. For all analyses in this paper we use values of $c = 0.01$, $\lambda = 0.01$, and $\gamma = 0.25$.

After calculation of scores, the LCA of all of the lowest scoring nodes, using a user-defined cut-off parameter, is calculated. For example, if the cut-off parameter is 0, only the highest scoring node (or nodes with the same score as the highest scoring node) is used to calculate the LCA. If the cut-off parameter is 5, the highest scoring node along with all other nodes within a score of 5

of the highest scoring node are used to calculate the LCA. Once the LCA node is identified, the classification of the single read (or paired-reads) will be assigned to the taxonomy assigned to that node. The classification of query sequences is parallelized.

Classification metrics used for accuracy evaluations.

We used the taxonomic identification metrics from Siegwald *et al.* 2017[26] and Sczyrba *et al.*2017[27]. A true positive (TP) read at a certain taxonomic rank has the same taxonomy as the sequence it was simulated from. A misclassification (FP) read at a certain taxonomic rank has a taxonomy different from the sequence it was simulated from. A false negative (FN) read, at a certain taxonomic rank, is defined as a read that received no assignment at that rank. For accuracy, we use the following measures for recall and misclassification rate.

$$\text{Recall} = \frac{TP}{TP + FN} \tag{7}$$

$$\text{Misclassification rate} = \frac{FP}{TP + FP + FN} \tag{8}$$

Classification of mock community reads

For Schirmer *et al.* (2015) we used the ERR777705 sample, for Gohl *et al.* (2016) we used the SRR3163887 sample, and for Braukmann *et al.* (2019) we used the SRR8082172 sample. All sample raw reads used for assignment were first filtered through the Anacapa Quality Control pipeline[28] with default parameters up until before the amplicon sequence variant (ASV) construction step. Only paired reads were retained for assignment. For mock datasets where the true species were only defined with "sp.", species assignment were excluded for all methods.

Leave-species-out and leave-one-individual-out analyses

We used 1,467 COI reference sequences from 253 species from the order Charadriiformes. For

the leave-species-out analyses we removed each of the species one at a time (excluding singletons, i.e. species only represented by a single sequence), yielding 252 different reference databases. For each database, we then simulated reads from the species that had been removed, and assigned to taxonomy using all methods tested (Tronko, kraken2, metaphlan2, and MEGAN), using the same reference databases and same simulated reads for all methods. For the leave-individual-out analysis, we removed a single individual from each species (excluding singletons) yielding 1,423 different reference databases. Assignments for all method were performed with default parameters and where a paired read mode was applicable, that mode was used when analyzing paired reads. For paired-end read assignments with MEGAN, the assignment is the LCA of the forward and reverse read assignments as described in the MEGAN manual v6.12.3. For metaphlan, the results from the forward reads and reverse reads were combined.

Custom 16S and COI Tronko-build reference database construction

For the construction of the reference databases in this manuscript, we use custom built reference sequences that were generated using common primers[29–32] for 16S and COI amplicons that have been used in previous studies[33–35] using the CRUX module of the Anacapa Toolkit[28]. For the COI reference database, we use the following forward primer: GGWACWGGWTGAACWGTWTAYCCYCC, and reverse primer: TANACYTCnGGRTGNCCRAARAAYCA from Leray *et al.* (2013) and Geller *et al.* (2013)[30,31], respectively, as input into the CRUX pipeline[28] to obtain a fasta and taxonomy file of reference sequences. For the 16S database, we use forward primer: GTGCCAGCMGCCGCGGTAA, and reverse primer: GACTACHVGGGTATCTAATCC from Caporaso *et al.* (2012)[29]. We set the length of the minimum amplicon expected to 0bp, the length

of the maximum amplicon expected to 2000bp, and the maximum number of primer mismatches to 3 (parameters `-s 0`, `-m 2000`, `-e 3`, respectively). Since all of the custom built libraries contain $\geq$500,000 reference sequences and MSAs, we first used Ancestralclust [36] to do an initial partition of the data, using parameters of 1000 seed sequences in 30 inital clusters (parameters `-r 1000` and `-b 30`, respectively). For the COI database, we obtain 76 clusters and for the 16S database we obtain 228 clusters. For each cluster, we use FAMSA[24] with default parameters to construct the MSAs and RAxML[21] with the model GTR+Γ of nucleotide substitution to obtain the starting trees for Tronko-build. The identified reference databases, MSAs, phylogeentic trees, and posterior probabilites of nucleotides in nodes for COI and 16S, are available for download at https://doi.org/10.5281/zenodo.7407318.

1. Gardner, P. P. *et al.* Identifying accurate metagenome and amplicon software via a meta-analysis of sequence to taxonomy benchmarking studies. *PeerJ* **7**, e6160 (2019).

2. Wood, D. E., Lu, J. & Langmead, B. Improved metagenomic analysis with kraken 2. *Genome biology* **20**, 257 (2019).

3. Lu, J. & Salzberg, S. Ultrafast and accurate 16s microbial community analysis using kraken 2. *bioRxiv* (2020).

4. Altschul, S. F., Gish, W., Miller, W., Myers, E. W. & Lipman, D. J. Basic local alignment search tool. *Journal of molecular biology* **215**, 403–410 (1990).

5. Ainsworth, D., Sternberg, M. J., Raczy, C. & Butcher, S. A. k-slam: accurate and ultra-fast taxonomic classification and gene identification for large metagenomic data sets. *Nucleic acids*

research **45**, 1649–1656 (2017).

6. Sievers, F. *et al.* Fast, scalable generation of high-quality protein multiple sequence alignments using clustal omega. *Molecular systems biology* **7**, 539 (2011).

7. Kapli, P., Yang, Z. & Telford, M. J. Phylogenetic tree building in the genomic age. *Nature Reviews Genetics* **21**, 428–444 (2020).

8. Balaban, M., Sarmashghi, S. & Mirarab, S. APPLES: Scalable Distance-Based Phylogenetic Placement with or without Alignments. *Systematic Biology* **69**, 566–578 (2019). URL https://doi.org/10.1093/sysbio/syz063. https://academic.oup.com/sysbio/article-pdf/69/3/566/33097067/syz063.pdf.

9. Matsen, F. A., Kodner, R. B. & Armbrust, E. V. pplacer: linear time maximum-likelihood and bayesian phylogenetic placement of sequences onto a fixed reference tree. *BMC bioinformatics* **11**, 538 (2010).

10. Barbera, P. *et al.* Epa-ng: massively parallel evolutionary placement of genetic sequences. *Systematic biology* **68**, 365–369 (2019).

11. Czech, L., Stamatakis, A., Dunthorn, M. & Barbera, P. Metagenomic analysis using phylogenetic placement–a review of the first decade. *arXiv preprint arXiv:2202.03534* (2022).

12. Hleap, J. S., Littlefair, J. E., Steinke, D., Hebert, P. D. N. & Cristescu, M. E. Assessment of current taxonomic assignment strategies for metabarcoding eukaryotes. *bioRxiv* (2020). URL https://www.biorxiv.org/content/early/2020/07/22/2020.07.21.214270. https://www.biorxiv.org/content/early/2020/07/22/2020.07.21.214270.full.p

13. Li, H. & Durbin, R. Fast and accurate short read alignment with burrows–wheeler transform. *bioinformatics* **25**, 1754–1760 (2009).

14. Needleman, S. B. & Wunsch, C. D. A general method applicable to the search for similarities in the amino acid sequence of two proteins. *Journal of molecular biology* **48**, 443–453 (1970).

15. Marco-Sola, S., Moure, J. C., Moreto, M. & Espinosa, A. Fast gap-affine pairwise alignment using the wavefront algorithm. *Bioinformatics* **37**, 456–463 (2021).

16. Huson, D. H., Auch, A. F., Qi, J. & Schuster, S. C. Megan analysis of metagenomic data. *Genome research* **17**, 377–386 (2007).

17. Truong, D. T. *et al.* Metaphlan2 for enhanced metagenomic taxonomic profiling. *Nature methods* **12**, 902–903 (2015).

18. Schirmer, M. *et al.* Insight into biases and sequencing errors for amplicon sequencing with the illumina miseq platform. *Nucleic acids research* **43**, e37–e37 (2015).

19. Gohl, D. M. *et al.* Systematic improvement of amplicon marker gene methods for increased accuracy in microbiome studies. *Nature biotechnology* **34**, 942–949 (2016).

20. Braukmann, T. W. *et al.* Metabarcoding a diverse arthropod mock community. *Molecular ecology resources* **19**, 711–727 (2019).

21. Stamatakis, A. Raxml version 8: a tool for phylogenetic analysis and post-analysis of large phylogenies. *Bioinformatics* **30**, 1312–1313 (2014).

22. Jukes, T. H., Cantor, C. R. *et al.* Evolution of protein molecules. *Mammalian protein metabolism* **3**, 21–132 (1969).

23. Felsenstein, J. Evolutionary trees from dna sequences: a maximum likelihood approach. *Journal of molecular evolution* **17**, 368–376 (1981).

24. Deorowicz, S., Debudaj-Grabysz, A. & Gudyś, A. Famsa: Fast and accurate multiple sequence alignment of huge protein families. *Scientific reports* **6**, 1–13 (2016).

25. Li, H. Aligning sequence reads, clone sequences and assembly contigs with bwa-mem. *arXiv preprint arXiv:1303.3997* (2013).

26. Siegwald, L. *et al.* Assessment of common and emerging bioinformatics pipelines for targeted metagenomics. *PLoS One* **12**, e0169563 (2017).

27. Sczyrba, A. *et al.* Critical assessment of metagenome interpretation—a benchmark of metagenomics software. *Nature methods* **14**, 1063–1071 (2017).

28. Curd, E. E. *et al.* Anacapa toolkit: an environmental dna toolkit for processing multilocus metabarcode datasets. *Methods in Ecology and Evolution* (2018).

29. Caporaso, J. G. *et al.* Ultra-high-throughput microbial community analysis on the illumina hiseq and miseq platforms. *The ISME journal* **6**, 1621–1624 (2012).

30. Leray, M. *et al.* A new versatile primer set targeting a short fragment of the mitochondrial coi region for metabarcoding metazoan diversity: application for characterizing coral reef fish gut contents. *Frontiers in zoology* **10**, 34 (2013).

31. Geller, J., Meyer, C., Parker, M. & Hawk, H. Redesign of pcr primers for mitochondrial cytochrome c oxidase subunit i for marine invertebrates and application in all-taxa biotic surveys. *Molecular ecology resources* **13**, 851–861 (2013).

32. Amaral-Zettler, L. A., McCliment, E. A., Ducklow, H. W. & Huse, S. M. A method for studying protistan diversity using massively parallel sequencing of v9 hypervariable regions of small-subunit ribosomal rna genes. *PloS one* **4**, e6372 (2009).

33. De Vargas, C. *et al.* Eukaryotic plankton diversity in the sunlit ocean. *Science* **348** (2015).

34. Leray, M. & Knowlton, N. Dna barcoding and metabarcoding of standardized samples reveal patterns of marine benthic diversity. *Proceedings of the National Academy of Sciences* **112**, 2076–2081 (2015).

35. David, L. A. *et al.* Diet rapidly and reproducibly alters the human gut microbiome. *Nature* **505**, 559–563 (2014).

36. Pipes, L. & Nielsen, R. Ancestralclust: clustering of divergent nucleotide sequences by ancestral sequence reconstruction using phylogenetic trees. *Bioinformatics* **38**, 663–670 (2022).

Acknowledgements We would like to thank Rachel Meyer and CALeDNA for their support in this project. We acknowledge Thorfinn Sand Korneliussen for advice on parallelization of the method.

Funding This work used the Advanced Cyberinfrastructure Coordination Ecosystem: Services & Support (ACCESS) Bridges system at the Pittsburgh Supercomputing Center through allocation BIO180028 and was supported by NIH grants 1R01GM138634-01 and 1K99GM144747-01.

Competing Interests We declare that we have no known competing financial interests or personal relationships that influenced this work.

Inclusion and diversity We support inclusive, diverse, and equitable conduct of research.

Correspondence Correspondence and requests for materials should be addressed to Rasmus Nielsen (email: rasmus_nielsen@berkeley.edu).

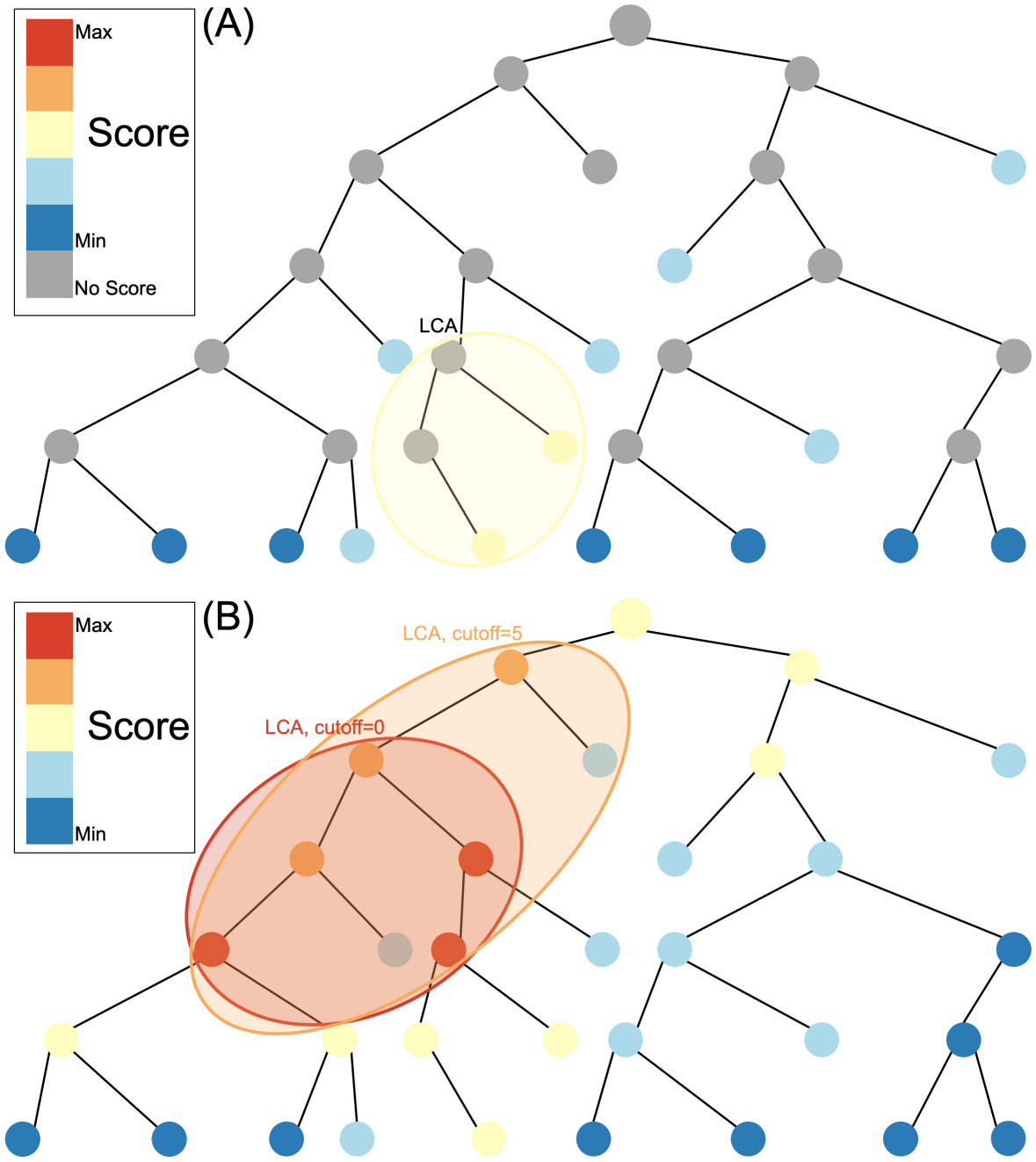

Figure (1): Species assignment in alignment-based methods (A) vs. Tronko (B). In Tronko, scores are calculated for all nodes in the tree based on the query's global alignment to the best `BWA-MEM` hit. The query is assigned to the LCA of the highest scoring nodes within the cut-off threshold.

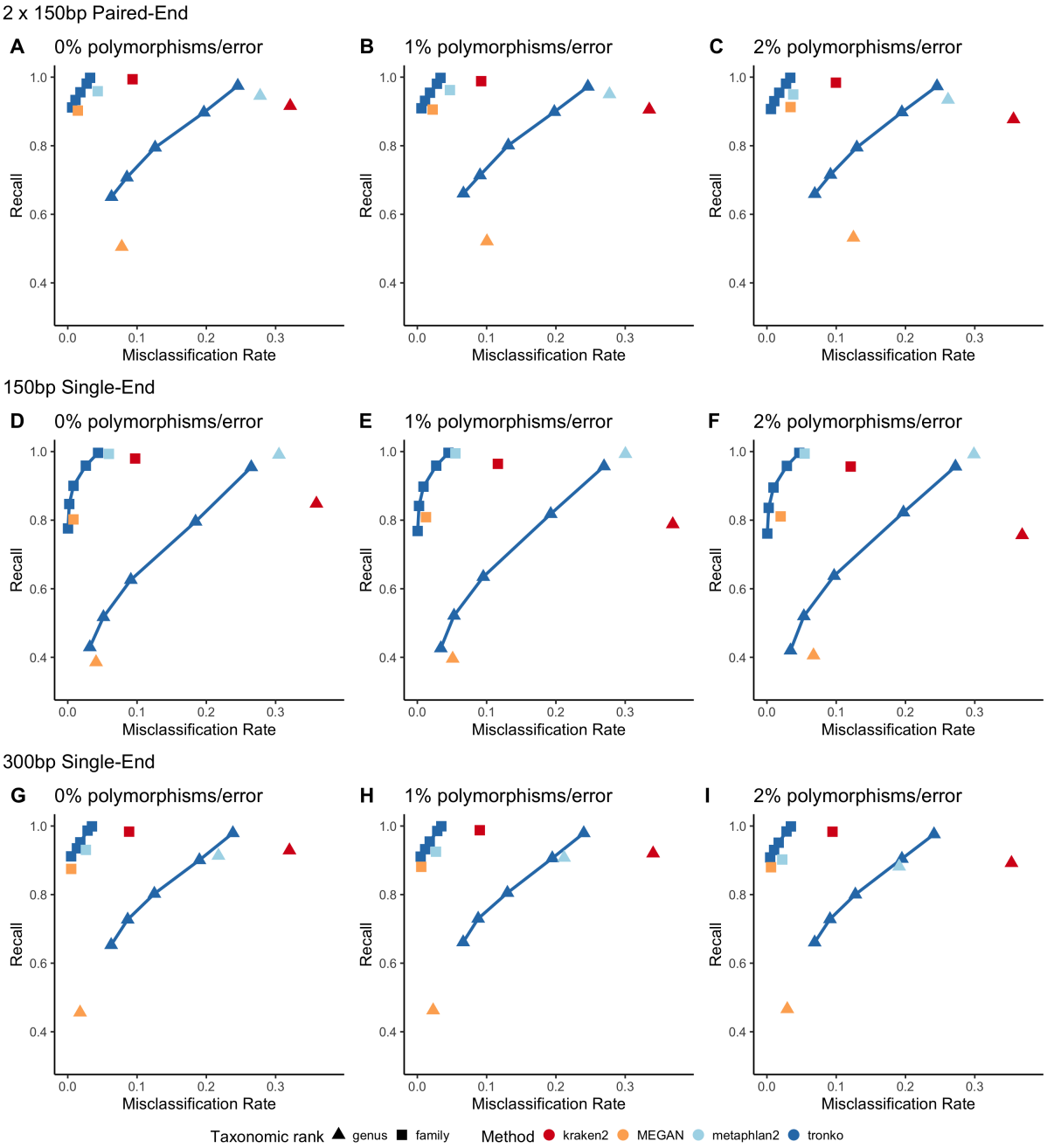

Figure (2): Recall vs. Misclassification rates using leave-one-species-out analysis with paired-end 150bp x 2 reads with 0% (A), 1% (B), and 2% (C) error/polymorphism, single-end 150bp reads with 0% (D), 1% (E), and 2% (F) error/polymorphism, and single-end 300bp reads with 0% (G), 1% (H), and 2% (I) error/polymorphism using kraken2, metaphlan2, MEGAN, and Tronko with cut-offs of 0, 5, 10, 15, and 20 using the Needleman-Wunsch alignment (solid line).

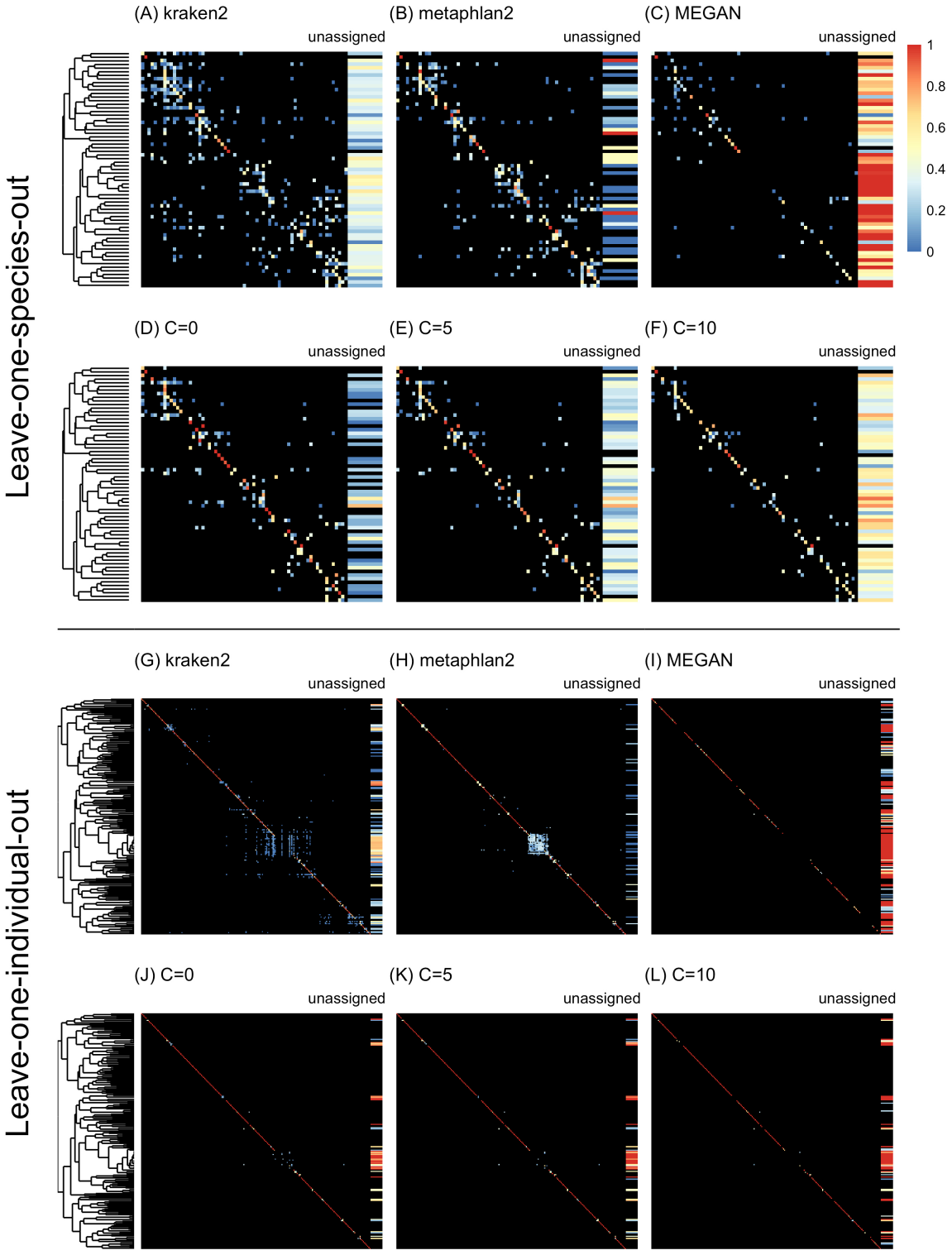

Figure (3): Confusion matrices at the genus level of the order Charadriiformes using the

Figure 3: Leave-one-species-out analysis with paired-end 150bp x 2 reads with 2% error/polymorphism using kraken2 (A), metaphlan2 (B), MEGAN (C), and Tronko using the Needleman-Wunsch alignment (NW) for cut-offs 0 (D), 5 (E), and 10 (F). Unassigned column contains both unassigned queries and queries assigned to a lower taxonomic level. Phylogenetic tree represents ancestral sequences at the genus level. Confusion matrices at the species level of the order Charadriiformes using the leave-one-individual-out analysis with paired-end 150bp x 2 reads with 2% error/polymorphism using kraken2 (G), metaphlan2 (H), MEGAN (I), and Tronko using the Needleman-Wunsch alignment (NW) for cut-offs 0 (J), 5 (K), and 10 (L). Phylogenetic tree represents sequences at the species level for leave-one-individual-out analyses and genus level for leave-one-species-out analyses.

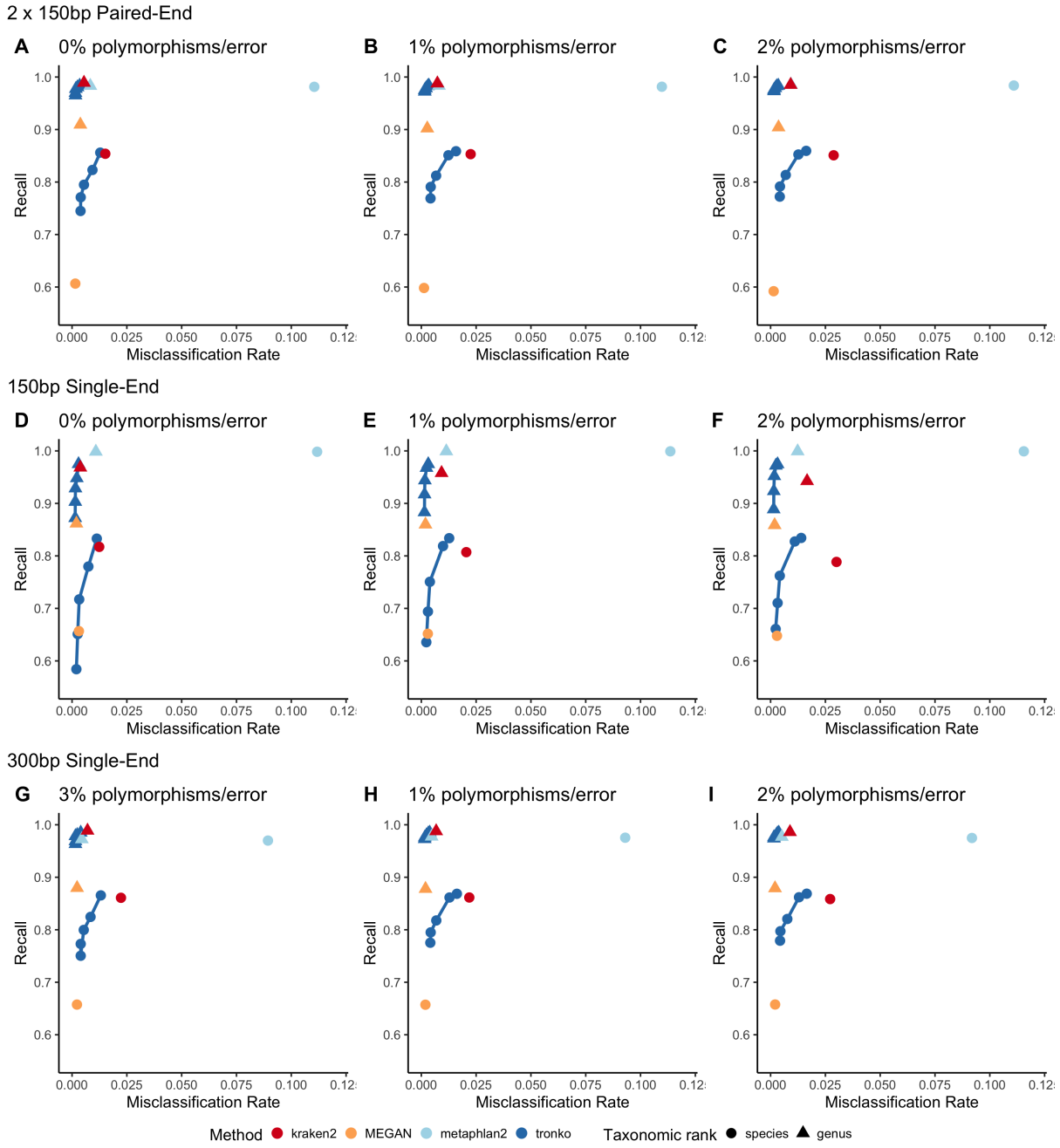

Figure (4): Recall vs. Misclassification rates using leave-one-individual-out analysis with paired-end x 2 150bp reads with 0% (A), 1% (B), and 2% (C) error/polymorphism, single-end 150bp reads with 0% (D), 1% (E), and 2% (F) error/polymorphism, and paired-end 300bp reads with 0% (G), 1% (H), and 2% (I) error/polymorphism using kraken2, metaphlan2, MEGAN, and Tronko with cut-offs of 0, 5, 10, 15, and 20 using the Needleman-Wunsch alignment (solid line).

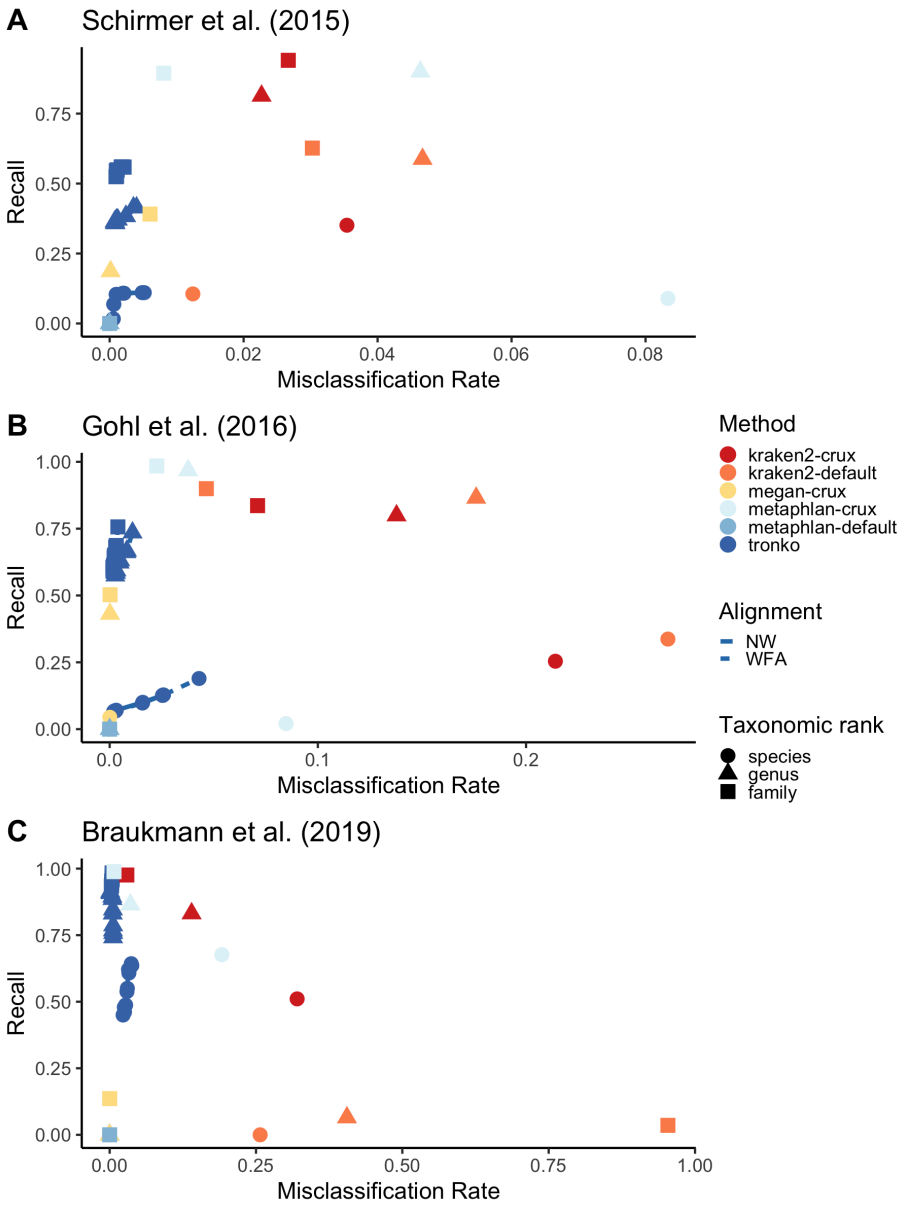

Figure (5): Recall vs. misclassification rates using mock communities from Schirmer et al. (2015)[18](A), Gohl et al. (2016)[19](B), and Braukmann et al. (2019)[20](C) using both Needleman-Wunsch and Wavefront alignment algorithms. Figures with smaller misclassification rates on the x-axis are available for Schirmer et al. (2015), Gohl et al. (2016), Braukmann et al. (2019) in Supplementary Figures S6, S7 and S8, respectively. For metaphlan2-default, no reads were assigned for any of the mock communities.

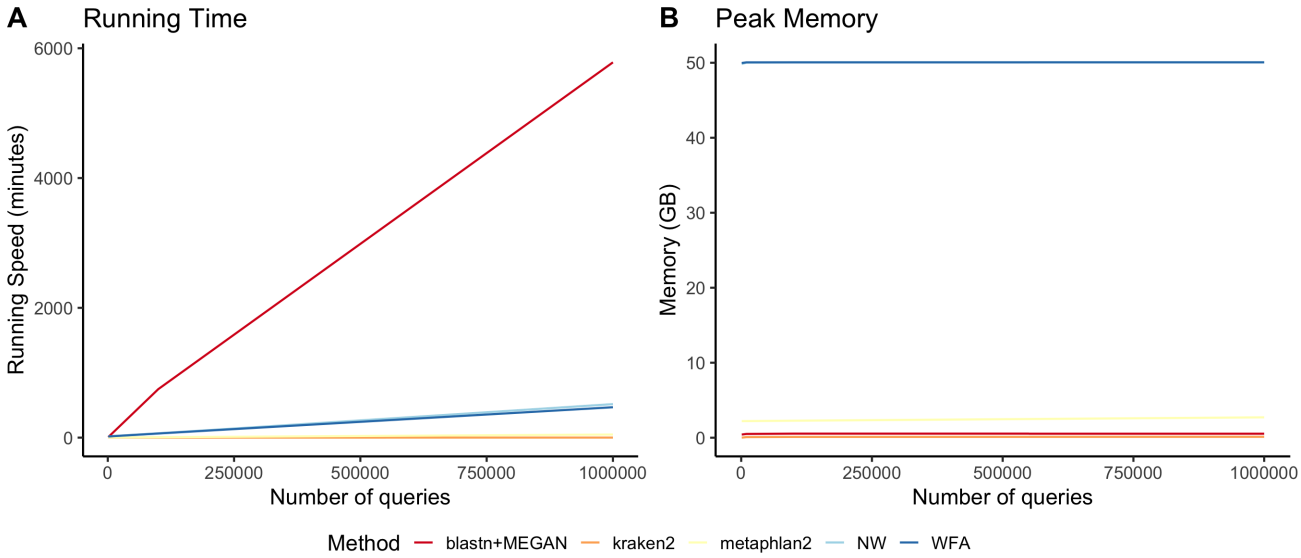

Figure (6): Comparisons of running time (A) and peak memory (B) using 100, 1000, 10000, 100000, and 1000000 queries for Tronko, blastn+MEGAN, kraken2, and metaphlan2 using the COI reference database. Needleman-Wunsch is NW and Wavefront alignment is WFA.