BIOINFORMATICS

Vol. 00 no. 00 2015
Pages 1–8

Unbiased probabilistic taxonomic classification for DNA barcoding

Panu Somervuo [1]*, Sonja Koskela [1], Juho Pennanen [1], R. Henrik Nilsson [2], and Otso Ovaskainen [1,3]

[1]Department of Biosciences, University of Helsinki, Finland
[2]Department of Biological and Environmental Sciences, University of Gothenburg, Sweden
[3]Centre for Biodiversity Dynamics, Department of Biology, Norwegian University of Science and Technology, Norway

Received on XXXXX; revised on XXXXX; accepted on XXXXX

Associate Editor: XXXXXXX

ABSTRACT

Motivation: When targeted to a barcoding region, high-throughput sequencing can be used to identify species or operational taxonomical units from environmental samples, and thus to study the diversity and structure of species communities. Although there are many methods which provide confidence scores for assigning taxonomic affiliations, it is not straightforward to translate these values to unbiased probabilities. We present a probabilistic method for taxonomical classification (PROTAX) of DNA sequences. Given a pre-defined taxonomical tree structure that is partially populated by reference sequences, PROTAX decomposes the probability of one to the set of all possible outcomes. PROTAX accounts for species that are present in the taxonomy but that do not have reference sequences, the possibility of unknown taxonomical units, as well as mislabeled reference sequences. PROTAX is based on a statistical multinomial regression model, and it can utilize any kind of sequence similarity measures or the outputs of other classifiers as predictors.

Results: We demonstrate the performance of PROTAX by using as predictors the output from BLAST, the phylogenetic classification software TIPP, and the RDP classifier. We show that PROTAX improves the predictions of the baseline implementations of TIPP and RDP classifiers, and that it is able to combine complementary information provided by BLAST and TIPP, resulting in accurate and unbiased classifications even with very challenging cases such as 50% mislabeling of reference sequences.

Availability: Perl/R implementation of PROTAX is available at http://www.helsinki.fi/science/metapop/Software.htm.

Contact: panu.somervuo@helsinki.fi

Supplementary Information: Supplement is available at Bioinformatics online.

1 INTRODUCTION

DNA barcoding has gained much interest in recent years, the Barcode of Life project (Sarkar and Trizna, 2011) being the most widespread endeavour in this field. The great promise of barcoding approaches is that in principle any organism can be identified from a tissue (or other biological) sample. To be effective and practical, barcoding genes should have sufficient variation between species, but limited variation within species, to provide the necessary signal for species-level identification. Commonly used barcoding genes include COI in animals, ITS in fungi, and rbcL as well as matK in plants (Hebert *et al.*, 2003; Schoch *et al.*, 2012; Hollingsworth *et al.*, 2009). Barcoding approaches are particularly powerful when combined with next-generation sequencing (NGS) methods, which can be used to produce massive amounts of sequence data from environmental samples (Wall *et al.*, 2009). That way, entire species communities can be profiled for taxonomic affiliations.

Two contrasting, although not mutually exclusive approaches have been applied to analyze the environmental sequence data. First, a number of approaches have focused on classifying samples into operational taxonomic units (OTUs), which can be done in the absence of a well-resolved taxonomy and reference sequence database. The OTU approach is especially prevalent in studies on bacteria and other micro-organisms (Hao *et al.*, 2011). The complementary problem, which we address here, and which underlies the Barcode of Life project, is the case where a sequence reference database and an established taxonomy are available. Here the problem lies in classifying the environmental sequences with respect to an existing taxonomy. In popular software packages like MOTHUR (Schloss *et al.*, 2009) and QIIME (Caporaso *et al.*, 2010), it is possible to carry out both the clustering and the taxonomic classification.

Several methods exist for assigning taxonomic labels to sequence data. MEGAN (Huson *et al.*, 2007) classifies sequences according to the lowest common ancestor (LCA) node of BLAST hits. BLAST is also used in MetaPhlAn (Segata *et al.*, 2012). There are fast alternatives for BLAST, e.g. LAST (Kielbasa *et al.*, 2011) which is based on suffix array and used in Taxator-tk (Dröge *et al.*, 2015). PhymmBL (Brady and Salzberg, 2009) is based on the combination of BLAST and interpolated Markov models and mOTU (Sunagawa *et al.*, 2013) utilizes hidden Markov models (Eddy, 2011). Another set of tools uses methods from phylogenetics, e.g. TIPP (Nguyen *et al.*, 2014) is based on a phylogenetic placement algorithm. The commonly used RDP classifier (Wang *et al.*, 2007) is based on naïve Bayes classifier (NBC) which uses the 8-mer decomposition of a sequence. MyTaxa (Luo *et al.*, 2014) allows the use of multiple

[*]to whom correspondence should be addressed

markers simultaneously and Kraken (Wood and Salzberg, 2014) and CLARK (Ounit *et al.*, 2015) can utilize full-length genome sequences.

Existing classification methods give good results if the target classes are well represented with adequate training data (Austerlitz *et al.*, 2009; Bazinet and Cummings., 2012), but it has remained difficult to account for taxonomic units without reference sequences (Ross *et al.*, 2008). Another typical practical problem is the presence of mislabeled training data, which the present methods to our knowledge fail to properly account for.

With any method of sequence classification, a central question concerns the reliability of the classification. To move from sequence similarity to a more objective measure of the reliability of species identification, it is desirable to estimate the set of probabilities by which the query sequence represents the possible candidate species or higher taxonomical units. In our past work (Ovaskainen *et al.*, 2010), we developed a statistical model that estimates the probability by which the best matching reference sequence (for instance, based on a BLAST search) represents the true species. In this paper, we develop our previous approach into a general tool for probabilistic taxonomic classification, called PROTAX. PROTAX utilizes a reference sequence database and a taxonomic tree structure to estimate the probability with which an environmental sample can be placed in a given taxon. Outcomes of the classification include not only the species for which reference sequences are available, but also species for which no reference sequences are available, as well as species – or higher taxonomic units – that are not known to science in the sense that they are missing from the taxonomy. PROTAX accounts for the possibility that some of the reference sequences are mislabeled, a complication often present with real data.

PROTAX is based on a Bayesian multinomial regression model and it can use as predictors any kind of covariates, in particular any kind of sequence similarity measures. In this paper we present the general statistical framework and illustrate the use of PROTAX in the context of classifying amplicon sequencing data based on covariates generated by BLAST sequence similarity and the TIPP and RDP classifiers, as well as a combination of BLAST and TIPP. We illustrate with simulated data that PROTAX is able to perform taxonomical classification in an accurate and unbiased manner. In particular, we show how PROTAX can improve the classification accuracies of TIPP and RDP classifiers and remove the bias that they in some cases have for the estimates of classification uncertainty. We then apply PROTAX to a fungal case study, illustrating the relevance and generality of the results obtained for simulated data.

2 METHODS

Concerning the overall workflow of the PROTAX pipeline shown in Fig. 1, the input files consist of the sequence data to be identified, the taxonomy against which the sequence data are to be identified, and the reference sequence database that populates at least part of the taxonomy. The output file is the classification of the sequence data, in the fullest version the vector of probabilities of all possible outcomes for each query sequence.

The taxonomy is viewed as a tree structure in which the leaves correspond to species, and the other nodes to higher taxonomical levels such as genera and families. The taxonomy represents the set of known species and their assumed phylogenetic relationships, and it includes also those species for which no reference sequences are available. PROTAX accounts for the possibility of missing taxonomical levels, i.e. that some branches of the tree may be missing at any of the levels, due to not-yet-discovered taxonomical units. The reference sequence data are associated to leaves or higher-level nodes of the taxonomy, depending whether they have been identified to the species-level or say, the genus-level only. PROTAX estimates the fraction of the reference sequences that are mislabeled, and accounts for it when performing classifications. This is relevant, as the mislabeling rate can be high in some species groups. For example, in fungi it is estimated to be more than 10%, e.g. due to the fruit body from which the sample is derived being misidentified, the sample being contaminated, or other such problems (Nilsson *et al.*, 2012).

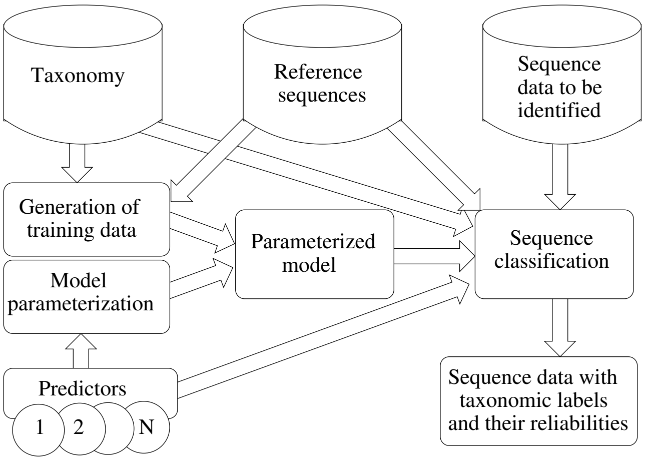
Taxonomy
Reference sequences
Sequence data to be identified
Generation of training data
Parameterized model
Sequence classification
Model parameterization
Predictors
1
2
N
Sequence data with taxonomic labels and their reliabilities

Fig. 1. Flowchart of PROTAX pipeline. PROTAX combines information of user-defined predictors which can be sequence similarities e.g. based on BLAST, or outputs of existing classifiers such as TIPP or NBC.

2.1 Modeling the probability of placement to a given taxonomical level

The aim of PROTAX is to partition the probability of species identity to all logical outcomes that can be derived from the taxonomical tree: species present in the taxonomy (with or without reference sequences), unknown species belonging to one of the levels of the taxonomy (e.g. an unknown species belonging to a known genus, or an unknown species belonging to an unknown genus that belongs to a known family). The collection of all these cases forms the set of outcomes, for each of which PROTAX estimates a probability, in such a way that the probabilities sum to one. If the query sequence can be classified with high certainty, one of these probabilities will be close to one. If there remains a high level of uncertainty, even the highest probability may be close to zero.

The algorithm works in a hierarchical manner from the root of the tree towards the leaves. As illustrated in Fig. 2, at any given node z, the task is to decompose the probability assigned to that node to the known branches and the possible unknown branches. The number of missing branches is unknown in reality. We denote the expectation for the number of missing branches under the node z by u_z. Expert knowledge about which parts of the taxonomy are likely to be more complete than others can be used to estimate u_z. In the absence of such information, we may simply set $u_z = 1$ for all nodes, as we do in the rest of this paper.

We denote the levels of the nodes by $l = 1, 2, \ldots, L$, so that for example the levels $l = 1, 2, 3$ could correspond to family, genera and species, respectively. Assume that the node z at level l has n_z known branches. Conditional on the query sequence belonging to the node z, we aim to decompose the unit probability into $1 + n_z$ parts, out of which the first one ($i = 0$) corresponds to the possibility that the species belongs to an unknown branch, and the remaining ($i = 1, 2, \ldots, n_z$) parts to the known branches. We denote by Y_z the random variable indicating to which of the branches the sequence belongs. We model Y_z by the multinomial regression model

$$P(Y_z = i) = \frac{w_{zi} \exp(\sum_{j=1}^{m} X_{ij}^z \beta_j^z)}{\sum_{i=0}^{n_z} w_{zi} \exp(\sum_{j=1}^{m} X_{ij}^z \beta_j^z)}. \quad (1)$$

Here β_j^z denotes the regression coefficient associated to predictor $j = 1, \ldots, m$ at node z. The matrix X^z is the $(1 + n_z) \times m$ design matrix which includes m predictors that are used to guide the classification algorithm. In principle any predictors that are informative for the classification purpose can be used. The weights w_{zi} are set to $w_{zi} = u_z$ for $i = 0$ and otherwise to $w_{zi} = 1$.

While in the more general model the values of the regression coefficients β_j^z can be node-specific, we make here the simplifying assumption that they are constant within each level, and thus we denote them henceforth by β_j^l, with $j = 1, \ldots, m$, with l ranging across the levels present in the tree. Thus, the parameters β_j^1 guide the classification from the root level to the lowest taxonomical level present in the taxonomy, and finally the parameters β_j^L guide the classification from the genus level to the species level. In the more general model also the number m can be node- or level-specific, but here we assume that it is same for all nodes and levels.

To account for the possibility that the reference sequences can be misidentified, we denote by $I(s)$ the identity of the true species behind the sequence s, and by $I^*(s)$ the species identity that is assigned to the sequence. For any species identity I, we denote by $Y(I)$ the outcome corresponding to the species identity, so that, e.g. $Y = (2, 1, 3)$, stands for family 2, genus 1 within that family, and species 3 within that genus. For another example, $Y = (2, 0, 0)$ stands for family 2, unknown genus within that family, which implies that also the species within that genus is unknown.

Given a query sequence s, the probability that the observed outcome $Y(I^*(s))$ is the outcome y is

$$\begin{aligned} P\left(Y(I^*(s)) = y|\beta, q\right) &= qP\left(Y(I^*(s)) = y\right) \\ &+ (1-q)P\left(Y(I^*(s)) = y|\beta, I^*(s) = I(s)\right). \quad (2) \end{aligned}$$

Here q is the probability that the training sequence is misidentified and $P(Y = y)$ is the prior probability of the outcome y. The probability $P\left(Y(I^*(s)) = y|\beta, I^*(s) = I(s)\right)$ is given by the product of the multinomial probabilities (Eq. 1) over the relevant levels.

We fitted the model to training data (see below) with Bayesian inference, the parameters to be estimated being the level-specific regression coefficients β_j^l and the mislabeling probability q. We assigned for each of the regression coefficients the essentially uninformative $N(0, 100^2)$ prior, and used the same prior also for the logit-transformed probability q. As described in the Supplement, we fitted the model using the Metropolis algorithm, with normally distributed proposal distributions adjusted adaptively during the burn-in.

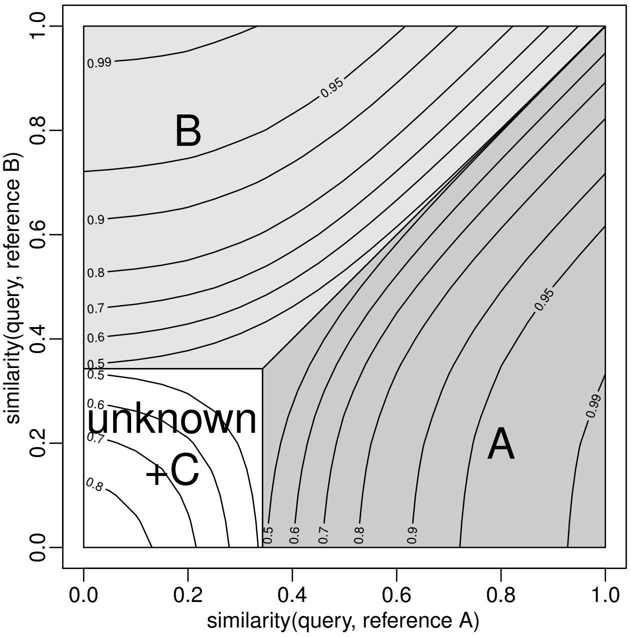

Fig. 2. Visualization of how PROTAX decomposes the probabilities of taxonomic placement among different possibilities. We consider here classification of one level only, e.g. from a given genus to the underlying species. Nodes A, B, and C represent three known species of this genus and "unknown" represents possible unknown species within this genus. We assume that species A and B have reference sequences whereas species C does not have. Classification probabilities are shown for an input sequence as a function of the sequence similarity (value between 0 and 100%) for species A (x-axis) and species B (y-axis). In this example, the design matrix is $X = ((0,0,0), (0,1,\text{simA}), (0,1,\text{simB}), (1,0,0))$, where the rows correspond to the nodes "unknown", A, B, and C, respectively, and simA and simB are the sequence similarities between the query and reference sequences. In this numerical example, we have assumed the parameter values $\beta = (0.1, -2, 8)$. In applications, this parameter vector is estimated based on training data (see main text). To simplify the illustration, we have summed here the probabilities of the unknown node and the species C without reference sequences.

2.2 Model parameterization

As the first example, denoted by PROTAX(BLAST), we construct the predictors using BLAST sequence similarity, with four predictors ($m = 4$). We set the case of a missing branch as the reference level, and thus the first row ($i = 0$) of the design matrix to $(0, 0, 0, 0)$. The subsequent n_z rows of the design matrix model the probability of the species belonging to each of the known branches. If the branch does not contain any sequence data, the row of the design matrix is set to $(1, 0, 0, 0)$, and thus the regression coefficient β_1^l models the probability (relative to a missing branch) that the species belongs to a node that is included in the taxonomy but for which no reference sequences are available. For branches for which one or more reference sequences are available, the row of the design matrix is set to $(0, 1, s_1, s_2)$, where s_1 and s_2 are respectively the mean and the maximal sequence similarity between the query sequence and a representative subset (see Supplementary material) of reference sequences under the node. The mean s_1 is first averaged over the sequences for each species, then over the species within each genus, and so on, always averaging over all units for which at least one sequence is available. Thus the regression coefficients β_2^l, β_3^l, and β_4^l model the probability by which the query sequence belongs to a node with reference sequences with a given mean and maximal similarity. The reason for including not only maximal similarity but also mean similarity is that the latter is expected to be more robust to mislabeling error.

As the second example, denoted by PROTAX(TIPP), we construct the predictors using the classification provided by TIPP, with three predictors ($m = 3$). In this case the first row ($i = 0$) of the design matrix is $(0, 0, 0)$, and rows for branches not containing sequence data are set to $(1, 0, 0)$. The remaining rows are $(0, 1, s_3)$, where $s_3 = \text{logit}(\varepsilon + (1 - 2\varepsilon)p)$, where p is the probability predicted by TIPP, $\text{logit}(p) = \log(p/(1 - p))$, and we have set the parameter $\varepsilon = 0.001$ to avoid singular cases related to classifications with $p = 0$ and $p = 1$. Similarly as with PROTAX(TIPP), we also construct the predictors for the output of RDP classifier.

As a fourth example, denoted by PROTAX(BLAST+TIPP), we utilize the predictors s_1, s_2, and s_3 for nodes with reference sequences, thus ending up to a model with five predictors ($m = 5$).

2.3 Generation of training data

As PROTAX aims to estimate the probabilities of all possible outcomes for query sequences derived from environmental data, the training data should be constructed in such a way that they represent the expected species diversity in the environmental data. In the context of Bayesian analysis, information about the expected species diversity can be incorporated as a prior. The prior is constructed by listing all the possible outcomes associated with the taxonomical tree, and giving each outcome a prior probability. Training data can then be generated by first randomizing the desired outcome from the prior, and then generating data that mimic the chosen outcome (see below). The prior that we assume here corresponds to the assumption that it is equally likely that the environmental sequence belongs to any node at each level, including the possibility of an unknown branch. Thus, if the taxonomy involves e.g. two genera in a given family, we assume that conditionally on the species belonging to this family, it belongs to either genera with probability 1/3, and to an unknown genus under this family with the remaining probability 1/3. Applying this algorithm recursively generates prior probabilities for each possible outcome, and these prior probabilities sum to unity.

Ideal training data would consist of sequences that would represent all outcomes and that would be independent of the sequences present in the reference database. While such sequences could be generated in a simulation study, they will not be available with real data (e.g. sequences that represent species unknown to science, or sequences that represent known species without reference sequences). Thus, we attempt to mimic ideal training data as closely as possible. Exactly how this is done depends on the nature of the outcome to be mimicked, for which there are the following three possibilities.

The outcome is a known species with at least one reference sequence available. In this case ideal training data would be an additional query sequence of that species. If the species has at least two reference sequences, we used a randomly selected sequence as the query sequence, and left the other sequences to the reference database. If the species has only one sequence, it is not possible to generate training data for that particular species, as no reference sequences would remain available after taking the only sequence as the query sequence. In this case we generated training data for another outcome, which is as similar as possible to the originally chosen one, but for which the required data are available. This other outcome is the node which has the smallest taxonomic distance to the original node and which is on the same taxonomic level as the original node.

The outcome is a known species with no reference sequences available. In this case ideal training data would be a test sequence for the chosen species. As such data do not exist, we selected another species which had at least one reference sequence and was as related to the original species as possible, following the recursive algorithm described above. We randomly selected one of the reference sequences as the query sequence, and removed all the other reference sequences for that species to mimic the situation where no reference sequences would be available.

The outcome is an unknown branch of a given node. In this case ideal training data would be a sequence for a randomly selected species under that node. As such data are not available, we mimicked such a situation as follows: if the node had any branches with sequences, we selected randomly one of those branches, proceeded to a randomly selected species with at least one sequence, and selected randomly one of the sequences as the query sequence. We then removed the entire branch under the node to generate training data mimicking the case of a missing branch. If the originally chosen node had no branches with sequences, we generated training data for another node, which was at the same level as the originally chosen node (e.g. both representing genera) and as related to the original node as possible (in the sense of the recursive algorithm as described above), and which had the required data.

2.4 Evaluating the performance of the algorithm with simulated data

We evaluated the performance of the algorithm with the help of simulated data, including taxonomies, reference sequences and validation sequences. We assumed that the taxonomy consists of two levels, called here genus and species. We thus consider classification within a single family, and assume that in the underlying reality it is split to 10 genera, each of which is split to 10 species, thus yielding 100 species together. We model barcoding sequences by strings consisting of 300 nucleotides (A, C, T or G). We first generated an "ancestral sequence" to the root level by assigning each nucleotide randomly to one of the four possibilities. We generated a mutated sequence for each genus by assuming that each nucleotide mutates to a random one with probability ϵ_1. Similarly, we generated a sequence for each species by mutating each nucleotide of the genus-level sequence with probability ϵ_2. Finally, to generate multiple sequences for each species, we mutated the nucleotides of the species-level sequences with probability ϵ_3. The first 50 nucleotides were assumed to form a conserved region, and they were thus kept identical for all sequences.

We tested the algorithm with eight case studies summarized in Supplement Table 1. Case 1 was chosen as an easy starting point in which sequence identification should be successful to the species level. In this case, we assume that all species were known to science and thus included in the taxonomy, and that 4 sequences were available for all species, and that none of the sequences were mislabeled. The levels of sequence dissimilarity were set to $\epsilon_1 = 0.05$, $\epsilon_2 = 0.02$, $\epsilon_3 = 0.01$, and thus we assumed larger differences among genera than among species within a genus, and larger differences among species within a genus than among individuals within a species.

The remaining cases are derived from Case 1 by changing some of the parameters. In Case 2 we made the classification more difficult by setting $\epsilon_1 = \epsilon_2 = \epsilon_3 = 0.01$, and thus having only small differences among genera and species. Case 3 is otherwise identical to Case 1, but reference sequences are available only for the fraction $p = 0.5$ of all species. Cases 4 and 5 differ from Case 3 in that we assume that 20% ($q = 0.2$) or 50% ($q = 0.5$) of the sequences are mislabeled, respectively. Case 6 differs from Case 3 in that only 7 out of the 10 genera are assumed to be known to science, and that within each known genus only 7 out of the 10 species are assumed to be known to science. Thus in this case the taxonomy file consists of 49 species while the underlying reality still consists of the full set of 100 species. Case 7 is otherwise as Case 3 but the level of sequence similarity is assumed to vary within the taxonomical tree: when generating the mutations for a given node at level l, we randomized the level of sequence dissimilarity using a uniform distribution in the range $(0, 2\epsilon_l)$, so that the mean mutation probability is ϵ_l. The motivation for this case is that while such variation in sequence similarity is present in real taxonomies, the statistical model ignores it. Finally, Case 8 combines the aspects of Case 4 (mislabeled reference sequences), Case 6 (incomplete taxonomy), and Case 7 (heterogeneity in levels of sequence similarity) that are expected to make taxonomic classification difficult in the case of real data.

For each of these eight cases, we simulated 5 independent replicates of the taxonomy and the reference database, as well as 100 independent validation sequences, one for each species. We generated training data with 1,000 data points for each of the four PROTAX models (BLAST, TIPP, RDP classifier, and BLAST + TIPP), and used the MAP (maximum a posterior) parameter estimate for classification. We performed a full classification of the validation sequences by decomposing the probability of one among all possible outcomes of the taxonomic tree. We selected the outcome with the highest probability, and examined if it was the correct one, i.e. if it matched with the species behind the generated validation sequence. We assessed the accuracy and bias of the classifications by plotting the cumulative predicted probabilities against the cumulative number of cases in which the outcome with the highest probability was correct. Concerning a possible bias, if an outcome is assigned e.g. the probability 0.9, it should be correct in 90% of the cases and wrong in 10% of the cases. If in reality the answer would be correct, say in 50% or in 99% of the cases, the assessment of classification reliability is biased. An unbiased classification is accurate if it assigns high probabilities to some outcomes and low to others, while it is not accurate if it assigns equally low probabilities for all outcomes. For example, if the algorithm would return just the prior probabilities, it would be unbiased, but not accurate.

While we predicted the probabilities separately for each possible outcome, we pooled some of them before comparing to the true species identity in the validation data. The reason here is that if a genus has several species with no reference sequences, the classification probabilities for these outcomes are necessarily identical. Thus we summed these to obtain a single probability for the outcome being any species within a particular genus with no reference sequences. Further, we added to this prediction the probability of the sequence belonging to an unknown species of that genus, and thus the pooled outcome was ”an unknown species within this genus, or one of the species in this genus without reference sequences”.

To compare the performance of the PROTAX models with the baseline implementations of the TIPP and RDP classifiers, we performed sequence classification with these (without the PROTAX extension) and assessed the bias and accuracy in the same way as we did for the PROTAX models.

2.5 Singling out mislabeled reference sequences

In addition to identifying environmental sequences (mimicked above by the validation sequences), the fitted model can be used to single out mislabeled reference sequences by treating the reference sequences as test sequences, and by comparing the probabilities of the outcomes to the species label. While making such predictions, the sequence to be classified is removed from the set of reference sequences to avoid circular results due to the 100% match to itself.

We considered a reference sequence as inconsistent if the model predicted a wrong genus with high confidence. To be conservative, we required the probability of the most likely classification to be at least 100 times the probability of the outcome corresponding to the species label. We examined how well this heuristic approach performed by counting the fraction of truly mislabeled sequences that were classified as inconsistent (in the optimal case this would be one), as well as the fraction of correctly labeled sequences that were classified as inconsistent (in the optimal case this would be zero). We did this test for the three PROTAX models as well as the baseline TIPP model.

2.6 Case study of Polyporales of Finland

For the evaluation of the method with real data, we used a database consisting of ITS sequences of all Polyporales (one fungal order) species of Finland available through UNITE (Kõljalg *et al.*, 2013). We used a two-level taxonomy consisting of the genus and species levels. Our database involves 265 known species in 75 genera, and 336 ITS sequences that belong to 162 species in 58 genera. Sequence lengths vary between 502 and 972 bp, mean length being 688 bp. In addition to 103 species without any reference sequences, more than half (89/162) of the species with sequence information contained only one reference sequence. We computed pairwise sequence similarities with BLAST, as detailed in the Supplement.

Unlike with simulated data, with real data there are no validation sequences for which the species identity would be known with full certainty. Thus, we tested the performance of the fitted model by identifying each of the reference sequences against the database from which the focal sequence was excluded. We then assessed the performance of the model like we did with simulated data, with the

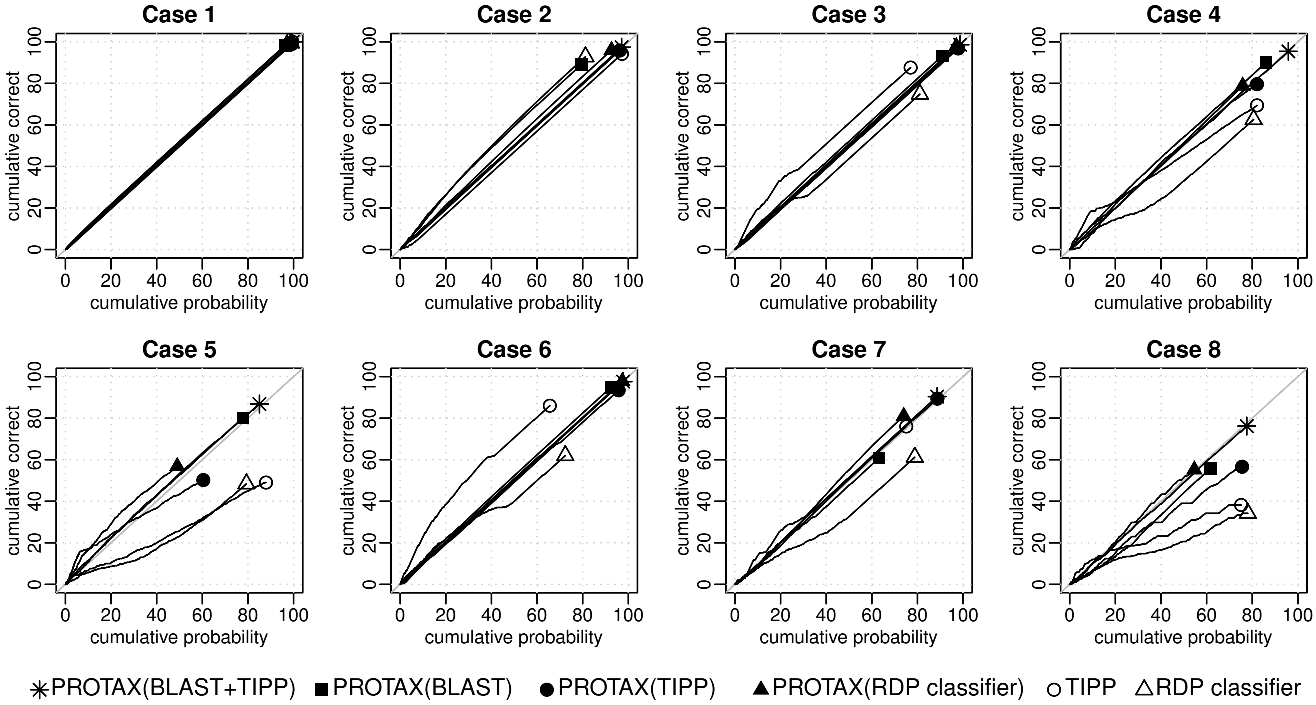

Fig. 3. Accuracy and bias of sequence identification by different versions of PROTAX in comparison to the baseline TIPP and RDP classifiers. The black lines show a cumulative plot for the predicted probabilities of the best outcome (x-axis) and the correctness of the prediction (y-axis). The grey lines show the identity line. The model-predicted probabilities are unbiased if the black lines follow the identity line, and they are the more accurate the longer the black lines are. The panels show the results for the Cases 1-8 (see Supplementary Table 1). Each black line is the summary of five individual test runs. Results from individual runs are shown in Supplement.

assumption that the query sequences were not mislabeled. Thus, our assessments of the model's identification success are likely to be conservative. We compared the performances between TIPP, RDP classifier, and the four versions of PROTAX as with simulated data. In addition, we constructed PROTAX for the full combination where BLAST, TIPP, and RDP classifier were used as predictors in the same model.

3 RESULTS

3.1 Simulated data

The performance of PROTAX against the simulated case studies is shown in Fig. 3. As expected, in Case 1 all algorithms resulted in almost perfect classifications, which are both accurate (cumulative probability close to 100%) and unbiased (the lines depicting the relationship between predicted and true identities fall very close to the identity line). In other words, all algorithms assigned in almost all cases a very high probability to only one of the outcomes, and in almost all cases this outcome was the correct one.

Making the test case more challenging decreased the accuracy, and in some cases introduced a bias in the predictions (Fig. 3). The presence of species without reference sequences (Case 3 vs. Case 1) as well as the presence of species not included in the taxonomy (Case 6 vs. Case 3) made the baseline TIPP algorithm produce conservative estimates of the identification probabilities. The PROTAX extension of TIPP corrected the bias on identification uncertainty, and also improved the accuracy of the identifications. However, the presence of 50% mislabeling (Case 5 vs. Case 3) made the predictions of TIPP and RDP classifier overconfident, which bias PROTAX was able to correct for on average (Fig. 3) but not within the individual replicates (Supplement Figs. 2 and 4). In contrast, PROTAX(BLAST) provided essentially unbiased estimates for all cases studied here.

Interestingly, PROTAX(BLAST) performed better in some cases, whereas PROTAX(TIPP) performed better in other cases. In particular, BLAST was a better predictor in the presence of mislabeling, whereas TIPP was a better predictor in the presence of heterogeneity in sequence similarity. The latter result is to be expected, as the phylogenetic model behind TIPP does not assume that the realized level of mutations is identical among the branches. The fact that BLAST and TIPP carry different kinds of information suggests that together they should perform better than either method in isolation. This was indeed the case: PROTAX(BLAST+TIPP) performed at least equally well as PROTAX(BLAST), PROTAX(TIPP), or the baseline implementation of TIPP in all the cases studied. Quite unexpectedly, even in the very challenging Case 5, where 50% of the reference sequences were mislabeled, PROTAX(BLAST+TIPP) was able to

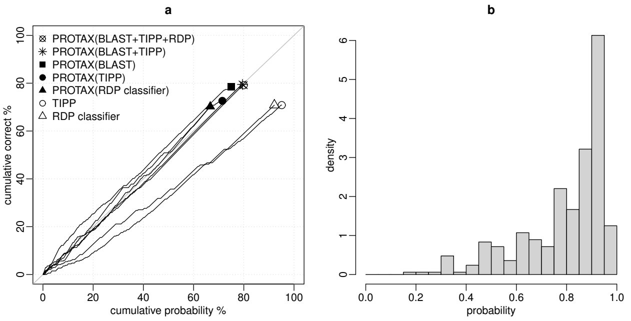

Fig. 4. a) Accuracy and bias of sequence identification by different versions of PROTAX, TIPP, and RDP classifier for 336 fungal ITS sequences. For 262 sequences, the best outcome of PROTAX(BLAST+TIPP) was a species with reference sequences. b) Histogram of best outcome probabilities from PROTAX(BLAST+TIPP).

correctly classify 87% of the validation sequences, and its estimates of identification uncertainty were essentially unbiased (Fig. 3).

The PROTAX models which included BLAST as a predictor were able to estimate the mislabeling probability with good accuracy in Cases 4 and 8 with $q = 0.2$ as well as in Case 5 with $q = 0.5$ (Supplement Fig. 7). In contrast, the way in which the statistical model accounts for mislabeling was not compatible with the predictive information provided by TIPP, as e.g. with $q = 0.5$ PROTAX(TIPP) tended to estimate the mislabeling probability either to zero or to one.

All models were successful in assessing the great majority of mislabeled sequences as unreliable and the great majority of the correctly labeled sequences as reliable (Supplement Fig. 8). The combined model PROTAX(BLAST+TIPP) worked the best also in the task of singling out mislabeled reference sequences. Interestingly, PROTAX(TIPP) was almost equally successful as PROTAX(BLAST) in singling out mislabeled reference sequences (Supplement Fig. 8) in spite of its poor ability to estimate mislabeling probability (Supplement Fig. 7).

3.2 Case study of Polyporales of Finland

Results based on PROTAX, TIPP, and RDP classifier are shown in Fig. 4. The conclusions are in concordance with the results from the simulated data. PROTAX was able to correct the bias of the TIPP and RDP classifiers, and the baseline correct classification rate of 71% from TIPP and RDP classifiers increased to 80% when combining BLAST and TIPP in the PROTAX model. Both in terms of the classification accuracy and the smallest bias, PROTAX(BLAST+TIPP) gave the best results. Including the output of the RDP classifier to the combination of BLAST and TIPP did not improve the results further, suggesting that TIPP and RDP classifier did not yield complementary information.

4 DISCUSSION

In this paper, we have introduced PROTAX, a probabilistic method for taxonomical classification. PROTAX converts sequence similarities into unbiased taxon membership probabilities. It takes into account uncertainties of both the taxonomy and the content of the reference sequence database. We have demonstrated its use in the context of fungal ITS amplicon sequencing, but the method is general and can be used with any markers. In addition to DNA sequences, it can be used also for classifying other types of data.

We emphasize that PROTAX can include any covariates in the regression model and thus the examples presented in this paper are demonstrations of only some possible choices. Besides using pairwise sequence similarities as a proxy for taxon membership, any node-sequence similarities can be used. As an example of this, we used the node probability from TIPP and RDP classifier as a covariate in PROTAX. Further, we note that the choice of using the maximal and mean sequence similarities in PROTAX(BLAST) is one choice among many possible sets converting sequence similarity into PROTAX predictors. We believe that all higher-level analyses, such as characterization of sample abundance profiles, benefit if the first step of assigning a sequence read into a taxonomic unit is done in a manner that enables reliable assessment of identification uncertainty.

PROTAX provides a statistical model that can be used with present classifiers or their combinations. It is not a new classifier per se but it combines the information obtained from user defined covariate sources. An important feature of PROTAX is that it gives unbiased probabilities of taxonomic placement. Furthermore, it explicitly models uncertainty related to missing data and missing branches in the taxonomy which to our knowledge other classifiers do not properly take into account.

In the experiments so far we have started with the simple assumption that the set of explanatory variables are the same for all nodes, and that the regression coefficients are specific to the level l only, so that $\beta_j^z = \beta_j^{l(z)}$, where $l(z)$ is the level to which node z belongs. The results have been satisfactory already with this approach, but to account for sequence similarity variation between taxonomical units at the same level (e.g. between genera), it would be possible to model β_j^z e.g. as a random effect with a multivariate normal structure.

We provide Perl and R scripts for training the models and using them for classification. Both speed and memory consumption depend mainly on the choice of covariate sources. As an example, we have constructed models for a large fungal ITS database with 75,000 reference sequences using a 6-level taxonomy with 130,000 species. For classifying 1,000 sequences using a single processor on a standard Linux desktop, it took 40 seconds to calculate sequence similarities using LAST (Kielbasa *et al.*, 2011) and 104 seconds to perform the taxonomic classification and output all nodes with probabilities above 0.01. Memory consumption was 477Mb. When classifying large amounts of sequences, speed and memory usage can be improved by pre-clustering the data and applying a naive parallelization of the algorithm.

ACKNOWLEDGEMENT

We thank Otto Miettinen for comments on the manuscript and Nerea Abrego for help with the Polyporales case study. This work was stimulated by a workshop organized by OO, during which Keith Harris, Jenni Nordén, Otto Miettinen, Christopher Quince and Oscar Westesson provided valuable input.

Funding: This study was funded by the Academy of Finland (Grant no. 250444 to OO) and the Research Council of Norway (Centres of Excellence funding scheme, project number 223257).

REFERENCES

Austerlitz,F. *et al.* (2009) DNA barcode analysis: a comparison of phylogenetic and statistical classification methods. *BMC Bioinformatics*, **10**, (Suppl 14):S10.

Bazinet, A. and Cummings, M. (2012). A comparative evaluation of sequence classification programs. *BMC Bioinformatics* **13**, 1471–2105.

Brady, A. and Salzberg, S. (2009). Phymm and PhymmBL: metagenomic phylogenetic classification with interpolated Markov models. *Nat. Methods*, **6**, 673–676.

Caporaso, J. *et al.* (2010). QIIME allows analysis of high-throughput community sequencing data. *Nat. Methods*, **7**, 335–336.

Dröge, J. *et al.* (2015). Taxator-tk: precise taxonomic assignment of metagenomes by fast approximation of evolutionary neighborhoods. *Bioinformatics*, **31**, 817–824.

Eddy, S. (2011). Accelerated profile HMM searches. *PLoS Comp. Biol.*, **7**, e1002195.

Hao, X. L. *et al.* (2011). Clustering 16S rRNA for OTU prediction: a method of unsupervised Bayesian clustering. *Bioinformatics*, **27**, 611–618.

Hebert, P. *et al.* (2003). Biological identifications through DNA barcodes. *Proc. R. Soc. Lond. B Biol. Sci.*, **270**, 313–321.

Hollingsworth, P. *et al.* (2009). A DNA barcode for land plants. *Proc. Natl. Acad. Sci. USA*, **106**, 12794–12797.

Huson, D. *et al.* (2007). MEGAN analysis of metagenomic data. *Genome Res.*, **17**, 377–386.

Kielbasa, S. *et al.* (2011). Adaptive seeds tame genomic sequence comparison. *Genome Res.*, **21**, 487–493.

Kõljalg, U. *et al.* (2013). Towards a unified paradigm for sequence-based identification of fungi. *Mol. Ecol.*, **22**, 5271–5277.

Luo, C. *et al.* (2014). MyTaxa: an advanced taxonomic classifier for genomic and metagenomic sequences. *Nucleic Acids Res.*, **42**, e73.

Nilsson, R. *et al.* (2012). Five simple guidelines for establishing basic authenticity and reliability of newly generated fungal ITS sequences. *MycoKeys*, **4**, 37–63.

Nguyen, N. *et al.* (2014). TIPP: taxonomic identification and phylogenetic profiling. *Bioinformatics*, **30**, 3548–3555.

Ounit, R. *et al.* (2015). CLARK: fast and accurate classification of metagenomic and genomic sequences using discriminative k-mers. *BMC Genomics*, **16**, 236.

Ovaskainen, O. *et al.* (2010). Identifying wood-inhabiting fungi with 454 sequencing – what is the probability that BLAST gives the correct species. *Fungal Ecology*, **3**, 274–283.

Ross, H. *et al.* (2008). Testing the reliability of genetic methods of species identification via simulation. *Syst. Biol.*, **57**, 216–230.

Sarkar, I. N. and M. Trizna. (2011). The Barcode of Life data portal: bridging the biodiversity informatics divide for DNA barcoding. *PLoS ONE*, **6**, e14689.

Schloss, P. *et al.* (2009). Introducing mothur: open-source, platform-independent, community-supported software for describing and comparing microbial communities. *Appl. Environ. Microbiol.*, **75**, 7537–7541.

Schoch, C. *et al.* (2012). Nuclear ribosomal internal transcribed spacer (ITS) region as a universal DNA barcode marker for fungi. *Proc. Natl. Acad. Sci. USA*, **109**, 6241–6246.

Segata, N. *et al.* (2012). Metagenomic microbial community profiling using unique clade-specific marker genes. *Nat. Methods*, **9**, 811–814.

Sunagawa, S. *et al.* (2013). Metagenomic species profiling using universal phylogenetic marker genes. *Nat. Methods*, **10**, 1196–1199.

Wall, P. *et al.* (2009). Comparison of next generation sequencing technologies for transcriptome characterization. *BMC Genomics*, **10**, 347.

Wang, Q. *et al.* (2007). Naïve Bayesian classifier for rapid assignment of rRNA sequences into the new bacterial taxonomy. *Appl. Environ. Microbiol.*, **73**, 5261–5267.

Wood, D. and Salzberg, S. (2014). Kraken: ultrafast metagenomic sequence classification using exact alignments. *Genome Biol.*, **15**, R46.

Supplement: Unbiased probabilistic taxonomic classification for DNA barcoding

P. Somervuo, S. Koskela, J. Pennanen, R.H. Nilsson, O. Ovaskainen

1 Simulated data cases 1-8

Table 1: Simulated scenarios used to test the performance of the PROTAX algorithm. The table describes for each of the Cases 1-8 the true number of branches within each level, the number of branches that are assumed to be known in the taxonomy, the frequency of mutations when moving from one level to the next one, variance in frequency of mutations, proportion of species for which reference sequences are generated, and the assumed mislabeling probability. The number of sequences per species was four in all cases.

Case	True number of branches within each level	Number of branches assumed to be known	Frequency of mutations when moving from one level to the next one $[\epsilon_1, \epsilon_2, \epsilon_3]$	Variance in frequency of mutations	Proportion of species for which reference sequences are available	Mislabeling probability
1	10	10	[0.05, 0.02, 0.01]	no	1	0
2	10	10	[0.01, 0.01, 0.01]	no	1	0
3	10	10	[0.05, 0.02, 0.01]	no	0.5	0
4	10	10	[0.05, 0.02, 0.01]	no	0.5	0.2
5	10	10	[0.05, 0.02, 0.01]	no	0.5	0.5
6	10	7	[0.05, 0.02, 0.01]	no	0.5	0
7	10	10	[0.05, 0.02, 0.01]	yes	0.5	0
8	10	7	[0.05, 0.02, 0.01]	yes	0.5	0.2

The following figures show accuracy and bias of sequence identification by different versions of PROTAX, TIPP, and RDP classifier. The black lines show a cumulative plot for the predicted probabilities of the best outcome (x-axis) and the correctness of the prediction (y-axis). The grey lines show the identity line. The model-predicted probabilities are unbiased if the black lines follow the identity line, and they are the more accurate the longer the black lines are. Each panel shows the results for five replicate data sets. The panels show the results for the Cases 1-8 (see Table 1).

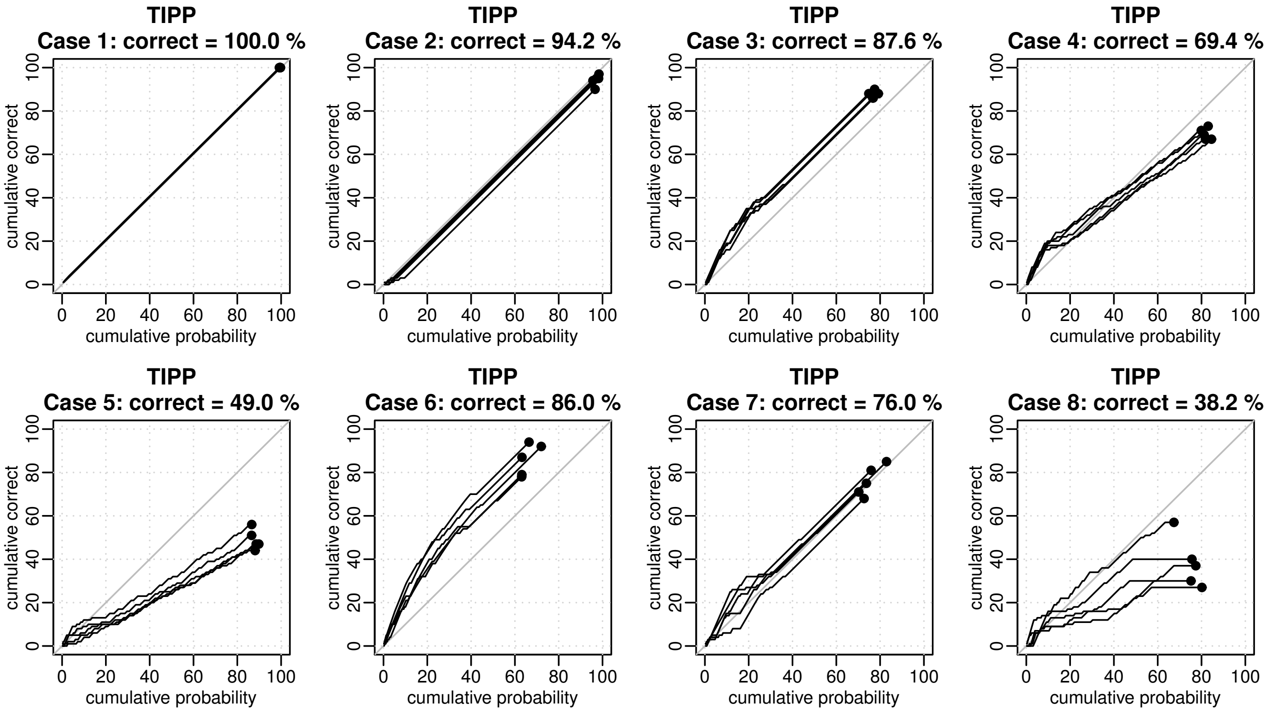

Figure 1: Baseline TIPP.

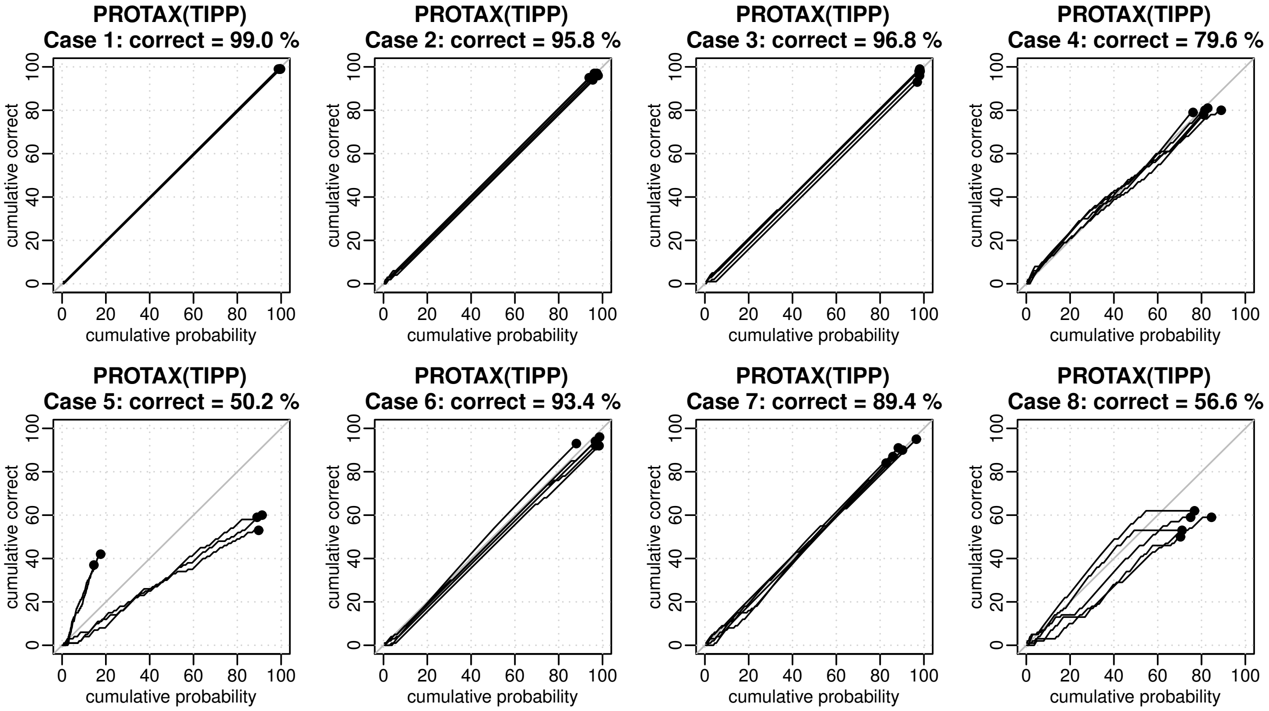

Figure 2: TIPP output as a covarite in PROTAX.

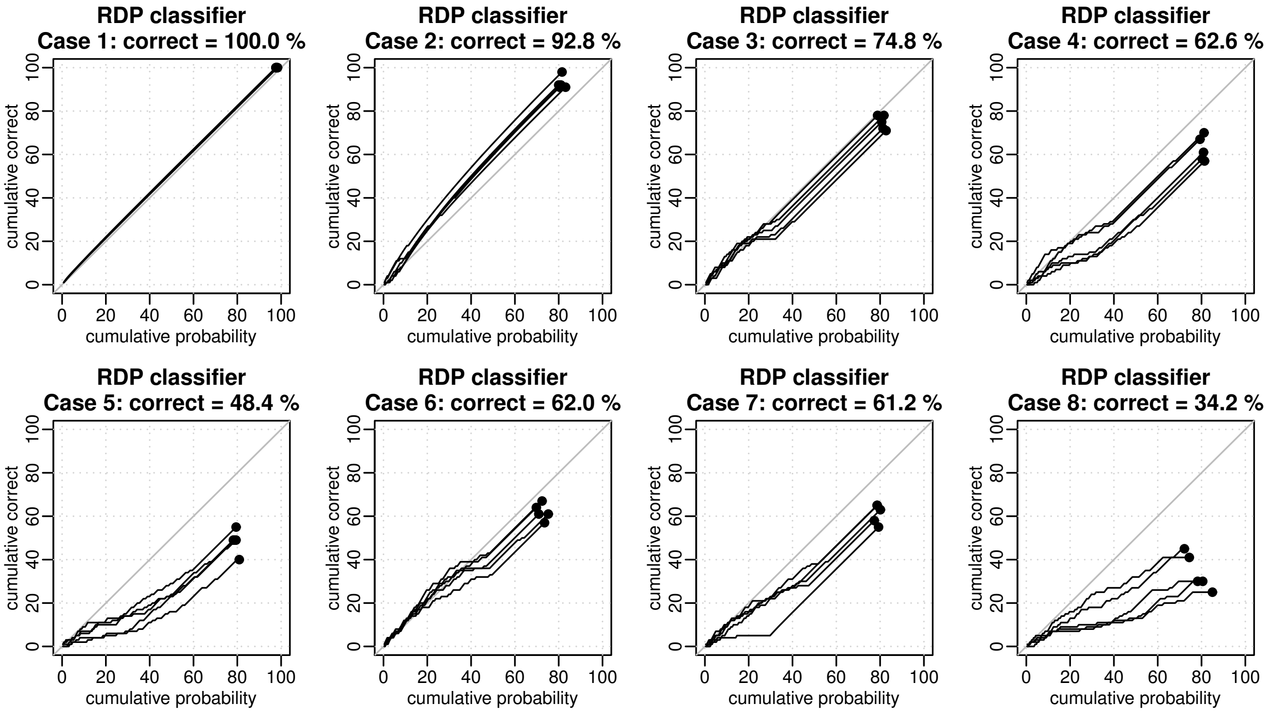

Figure 3: Baseline RDP classifier.

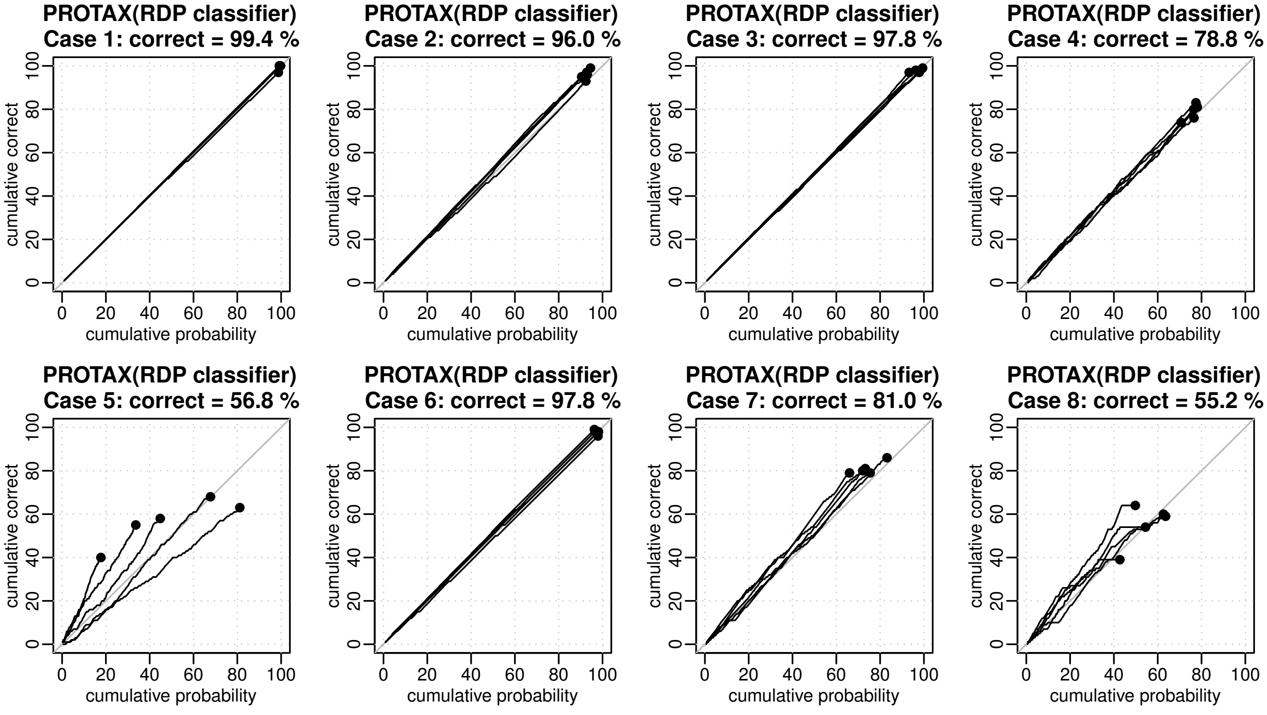

Figure 4: RDP classifier output as a covariate in PROTAX.

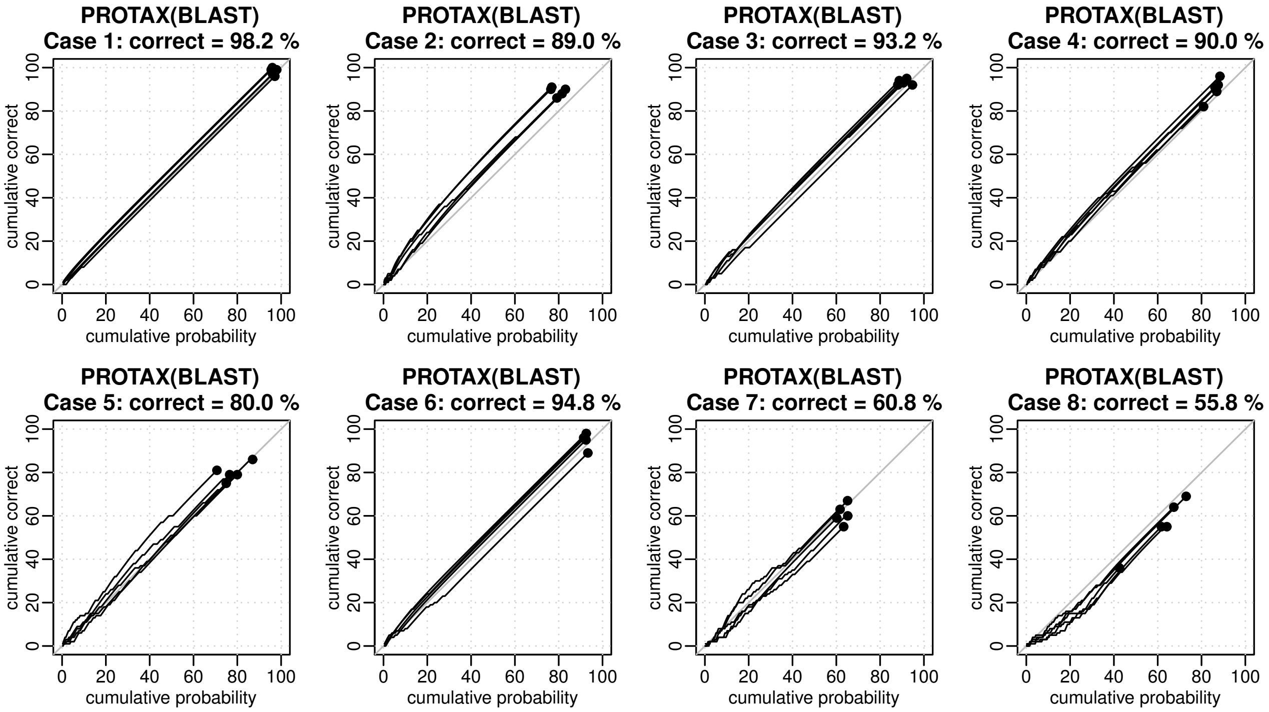

Figure 5: BLAST mean and max similarities as covariates in PROTAX.

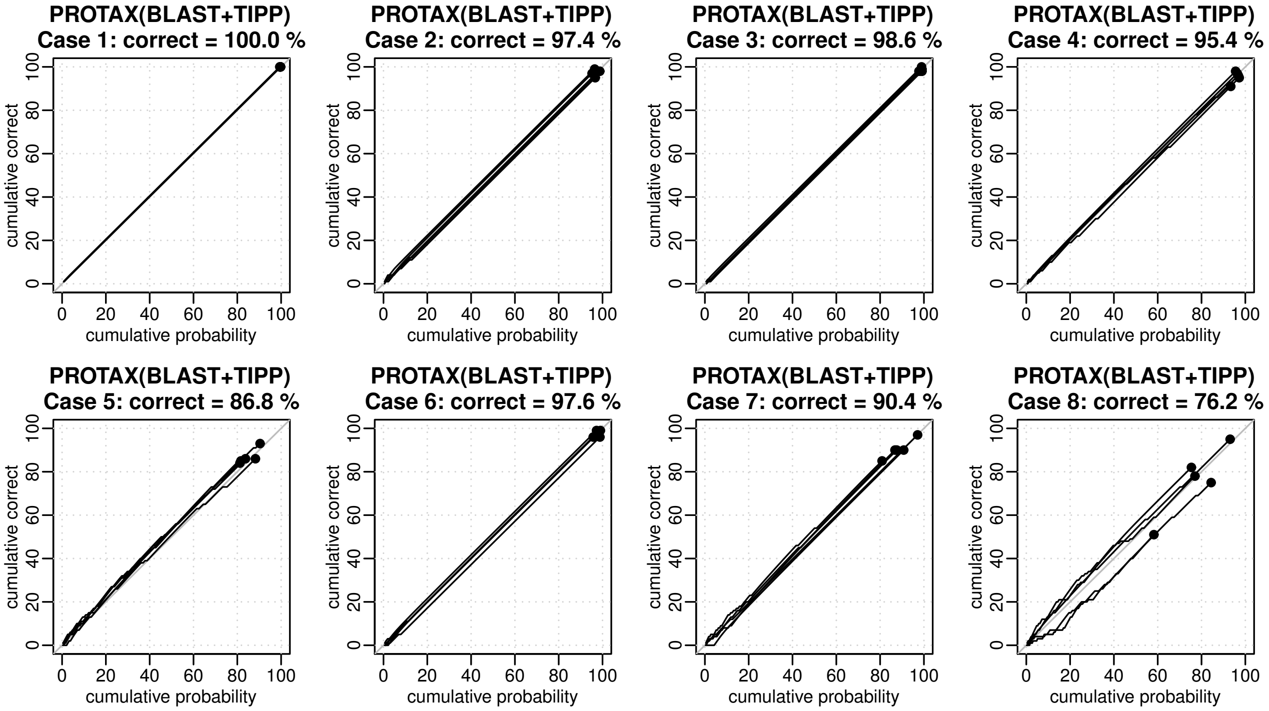

Figure 6: BLAST and TIPP as covariates in PROTAX.

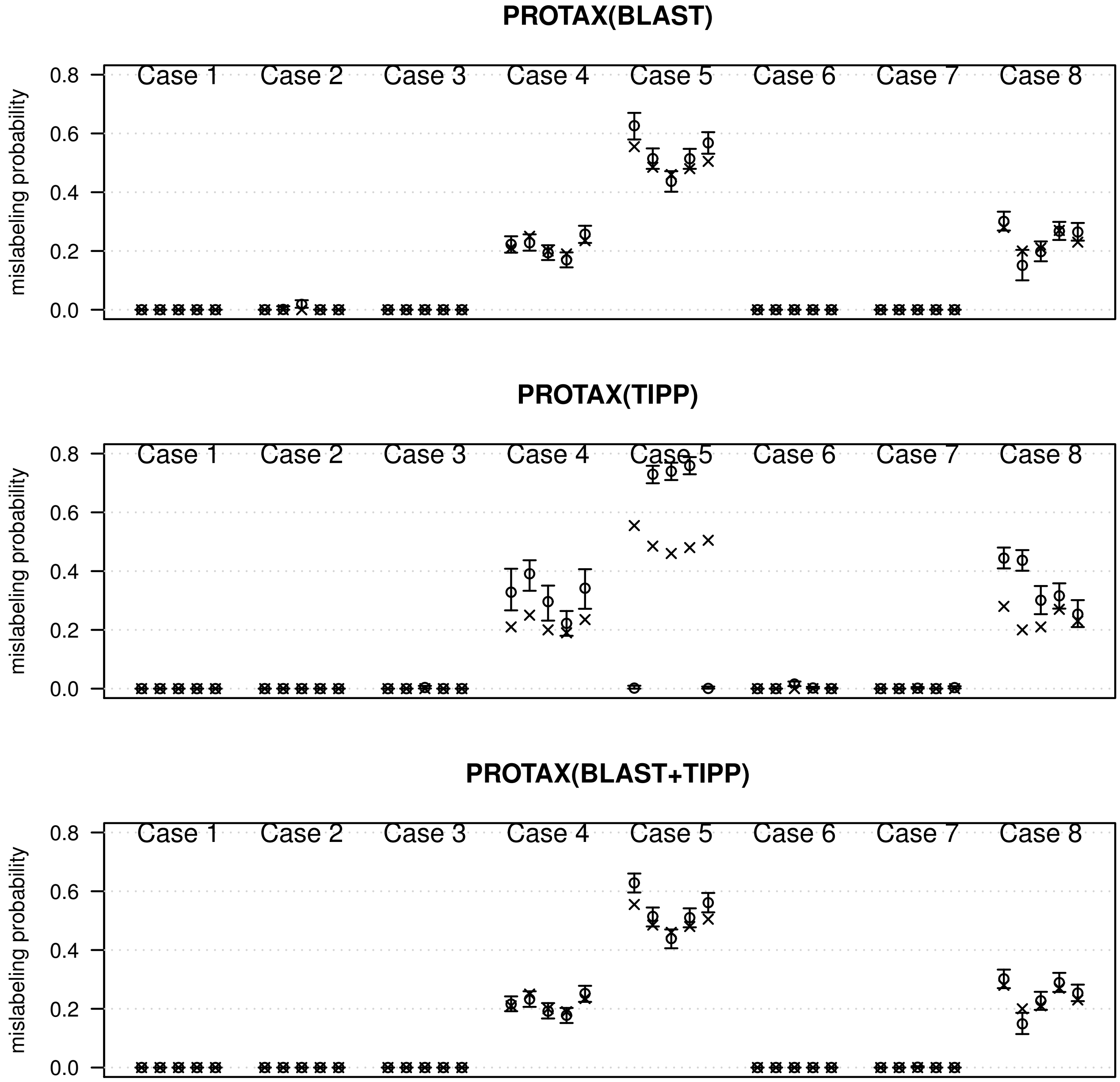

Figure 7: The ability of different PROTAX versions to estimate mislabeling probability. The circles and the error bars show the posterior mean estimate and the 95% credibility interval for mislabeling probability. The true mislabeling frequencies of the datasets are shown with the crosses. Results are shown for 5 replicates of each Case (see Table 1).

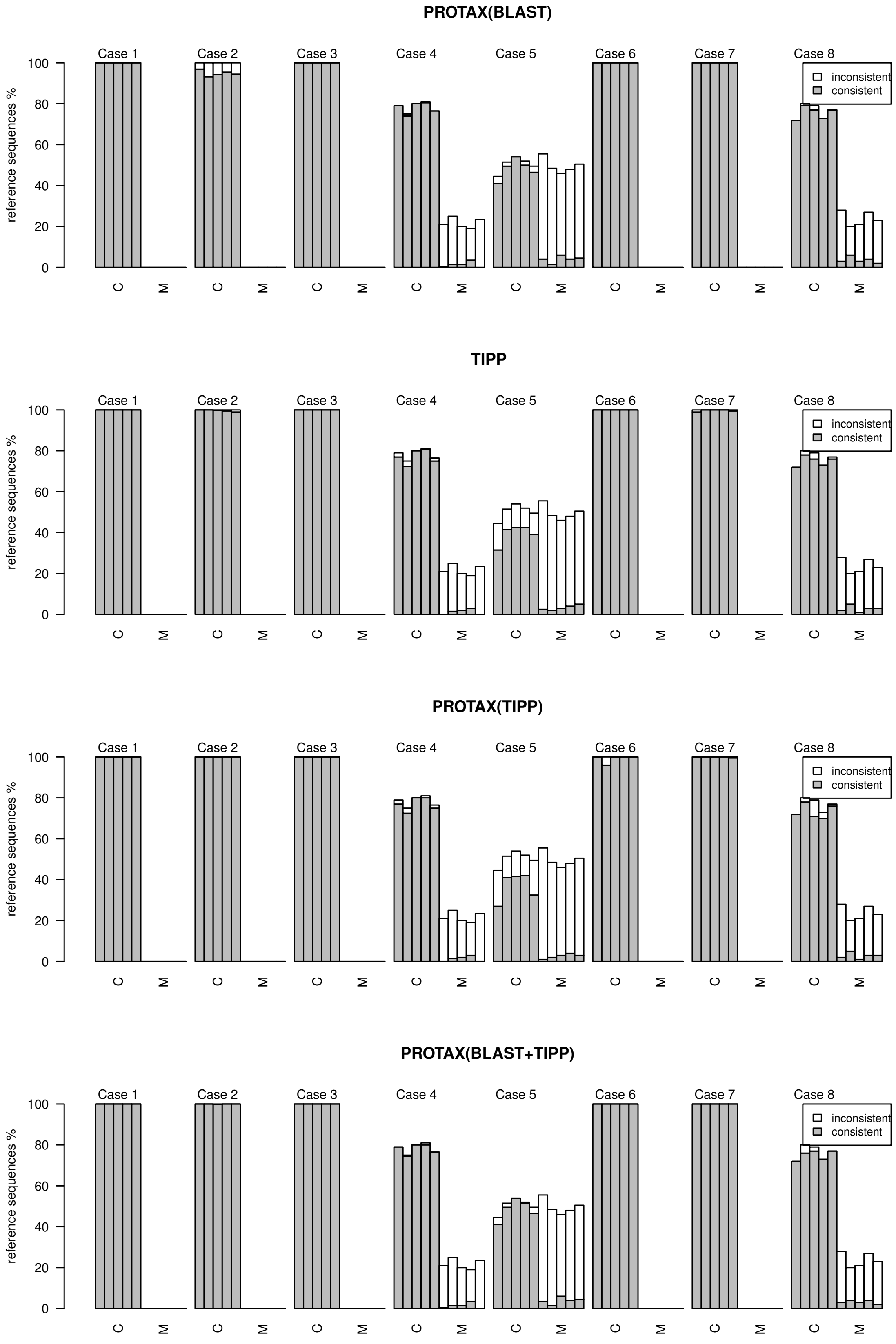

Figure 8: The ability of different versions of PROTAX and TIPP to detect mislabeled reference sequences. The numbers of reference sequences which are classified as consistent (grey) or inconsistent (white) are shown for both correct (C) and mislabeled (M) reference sequences. Results are shown for 5 replicates of each Case (see Table 1).

2 Selecting representative sequences

When using pairwise sequence similarities as a proxy for taxon membership, like e.g. in PROTAX(BLAST), each node of the taxonomy is associated with the set of representative sequences. Query sequence is compared against the representative sequences and the resulting similarities are used as covariates in PROTAX. PROTAX works by dividing the probability of one starting from the root node of the taxonomy and proceeding to higher level nodes until the node probability is below a user-defined threshold (e.g. 1%). In order for this hierarchical process to produce meaningful class predictions, intermediate sequence similarities should reflect the structure of the taxonomy so that the probability is divided correctly for higher-level taxonomy nodes. The number of available reference sequences may vary considerably between different branches of the taxonomy, which may bias the similarity score. Therefore the selection process tries to balance the representation of each taxonomic branch. The selection of representative sequences starts from the species level and proceeds to the root. During this taxonomy tree traversal, the representative sequences of each node are selected among the representative sequences of the present node's child nodes.

In our software, there are three choices for selecting the representative sequences attached to each taxon node:

1. use all reference sequences
2. pick random subset with user-defined maximum number
3. cluster the reference sequences with user-defined pairwise sequence similarity threshold

In the experiments with simulated data, the maximum number of representative sequences of each node was set to 20. In the case study of Polyporales of Finland, representative reference sequences attached to each node was limited to be 5 at the species level and 30 at the genus level. In both cases, random sampling (choice 2) was used.

3 Large taxonomy with sparse data

Taxonomy with large number of nodes without reference sequences is the reality in many cases, e.g. fungi where reference sequence data are available for less than 0.5% of the estimated 6 million extant species. The central question is how reliably we can say that a new sequence belongs to a species with existing reference data or whether it belongs to a species without any reference sequences. PROTAX does this via its multinomial regression model whose parameters have been estimated based on the given taxonomy and all its reference sequences. The sampled (node,sequence) pairs are used as training data in parameter estimation. Due to the way data generation works as explained in the Methods section of the main text, the nodes without any reference sequences will be mimicked by the nodes with reference sequences. As a consequence, if the number of the nodes with reference sequences is small, they are likely to be replicated in the training data. As implemented in our software, by finding the replicates, it is possible to make the size of the training data smaller. Each individual sample is weighted by the number of its replicates during MCMC so that compressed training data will not cause any bias.

Although the taxonomy in the Polyporales of Finland case study was not very large, for several taxa there were no sequence data available, so it can be used as an example of compressing the training data. We sampled 5,000 nodes in the taxonomy and the number of samples representing different types of nodes were distributed in the following way:

- unknown taxon: 1862
- known taxon without any reference sequences: 1117
- known taxon with only one reference sequence: 1042
- known taxon with at least two reference sequences: 979

The 5,000 randomly picked taxonomy node samples represented 994 unique (taxonomy node, query sequence, representative sequence) triplets and therefore the training data could be compressed into 1/5 of its original size.

4 Sequence similarities

In the Polyporales of Finland case study, pairwise sequences similarities were computed with BLAST initiated dynamic programming. Since BLAST gives local alignments, the overlap between two sequences can be fragmented into several BLAST hits. We extended the BLAST results so that the final similarity score represents the entire overlap region for each sequence pair. There are two possibilities for the overlap between two sequences:

1. one sequence is completely within another
2. two sequences form a dove tail type of alignment where the beginning of one sequence covers the end of another sequence.

The overlap alignment must therefore contain at least one beginning and one end nucleotide of a sequence. Sequence similarity is defined as the number of matching nucleotides divided by the length of the overlap region. In case there are no initial BLAST hits, the final pairwise similarity score is zero.

The procedure for calculating the final similarity score is the following: first a fast similarity method (BLAST) is used to find initial seed segments that are used as anchors for the final alignment done by rigorous dynamic programming (DP). Since the seed anchors are fixed, they effectively split the original two-dimensional DP trellis into smaller pieces, which reduces the computation. Furthermore, since we are not interested in the alignment but only the resulting similarity score, it is enough to use only two columns of the DP trellis in the computer memory: one for storing the cumulative similarity score, and another for storing the instantaneous similarity score.

If the number of sequences needed to represent each taxon is very large, there are many sequence comparisons to be done. Computing all pairwise sequence similarities may simply be too time consuming when using large taxonomies and data sets. Instead of calculating all pairwise sequence similarities, a database search approach can be used where only top-N hits are calculated for each query sequence. However, besides the max similarity, one of the PROTAX regression model covariates is the mean similarity, which requires the sequence comparisons against all representative sequences of the taxon. Thus, we need a way to handle the sequence similarities outside the top-N list. Since in the amplicon sequencing each sequence read represents the same genomic region, the similarity between any two sequences is expected to be nonzero. A possibility is to find an empirical floor value which is used by default for all pairwise sequence similarities. If a sequence pair is present in the top-N hit list and the corresponding similarity value exceeds the default value, the new value is used instead of the default value.

5 Parameter estimation

We used Markov chain Monte Carlo (MCMC) sampling with adaptive proposal distribution as explained in Ovaskainen et al. (2008), "Bayesian methods for analyzing movements in heterogeneous landscapes from mark-recapture data", Ecology, 89, 542–554.

The proposal distribution is based on the eigenvalue decomposition of the parameter covariance matrix. All eigenvectors are utilized one by one within each MCMC cycle. For the i:th eigenvector, the proposal for the new parameter vector is $\beta_{new} = \beta_{old} + re_i$, where r is a random sample from $N(0, (k_i d_i)^2)$, k_i is the step size, d_i is the square root of the i:th eigenvalue, and e_i is the i:th eigenvector. Here β includes both regression coefficients and the parameter for mislabeling probability. The parameter covariance matrix and step sizes are updated during the adaptation phase, which serves as burn-in after which the proposal distribution is fixed. Step size is adapted based on acceptance ratio of previous iterations, the targeted acceptance ratio being 0.44. In case the Markov chain has slow mixing, the adaptation phase can be repeated and the MCMC sampling be continued with new fixed proposal distribution. In the re-adaptation, the parameter vector is initialized by the values of the most recent iteration, the parameter covariance matrix is initialized to be an identity matrix, and the adaptive step sizes are initialized to be equal to one.

In our experiments, we fitted the model with 5,000 MCMC iterations (out of which 2,000 were used for burn-in). After training, we checked visually the parameter trace plots, and if the chain was not well mixed, applied new adaptation phase for 2,000 iterations and continued sampling another 3,000 iterations.