

Supplementary Data

Shell, Python, and R code for running experiments and analyses, as well as data files for recreating figures are available at https://github.com/liberjul/CONSTAXv2_ms_code.

CONSTAX algorithm

CONSTAX begins by taking an input database file, formatted as one downloaded from UNITE or SILVA databases, and creating the necessary files for training the classifiers. SILVA-formatted databases have arbitrary ranks, which do not necessarily apply across all domains of life. To address this arbitrary ranking, SILVA taxonomy is assigned Rank 1 (equivalent to domain) to Rank n (lowest assigned rank). It is recommended to filter the SILVA database to a given domain (Bacteria, Archaea, Eukaryota) to preserve the meaning of assigned ranks, which can be performed with the “--select_by_keyword” option.

Classification is completed with SINTAX, UTAX, and RDP without the “-b, --blast” flag, or with SINTAX, BLAST, and RDP with the “-b, --blast” flag. The BLAST search implementation is comparable to that described in Bokulich et al. 2018 (Bokulich *et al.*, 2018). Each input sequence is searched against a BLAST database generated from the database file using the blastn algorithm. A maximum number of hits is returned according to “-m, --max_hits”, which have an e-value equal to or below “-e, --evalue” and a proportion identity equal to or above “-p, -p_iden”. A confidence score is generated based on the greatest proportion of hits which agree at the given rank. SINTAX, UTAX, and RDP are already conventional classifiers, so no special rules are required.

Returned taxonomy assignments from each classification method are reformatted to be consistent. Taxonomy assignments are then filtered according to the confidence threshold and combined to create a consensus with the following rules: 1) if no classifications are above threshold, no taxa is assigned; 2) if two or three classifications are above threshold and agree, the majority taxa is assigned; 3) if only one classification is above threshold, that taxa is

assigned unless the “--conservative” flag is used, whereby no taxa is assigned; 4) if two or three classifications are above threshold and each is unique, the highest confidence taxa is assigned.

Clade Partition Cross-Validation

We employed the approach used to validate the SINTAX classifier (Edgar, 2016), clade partition cross-validation (CPX), as a means to assess the ability for CONSTAX to classify both known and novel taxa. At both the family and genus ranks, records within sub-taxa (genera and species, respectively), were randomly partitioned to reference or query groupings. Singletons, families or genera with only one sub-taxon, were assigned to the query group as novel taxa. Sensitivity, misclassification rates, over-classification rates, and errors per query were calculated according to (Edgar, 2016), for UNITE (Fig S1) and SILVA (Fig S2) databases. Classification performance was assessed on 5 replicates for each partition (family and genus rank) and for UNITE fungal representative sequences and SILVA bacterial ‘SSURef’ sequences. The same partitions were assessed with standard and conservative voting rules, and for commonly used regions of each marker. These regions were ITS1 and ITS2 from UNITE fungal sequences, extracted using ITSx (Bengtsson-Palme *et al.*, 2013), and the V3-4 and V4 hypervariable regions from SILVA bacterial sequences, extracted using in-silico PCR via custom Python script with primer sets 357wF-785R (Van Der Pol *et al.*, 2019) and 515f-806R (Parada *et al.*, 2016; Apprill *et al.*, 2015) allowing for 3 mismatched bases. For the UNITE database, classification was implemented with UTX and BLAST implementations, with individual and consensus assignments compared for both implementations. However, given the size of the SILVA SSURef database, training time for the UTX implementation would exceed 100 hours per replicate. Therefore, only the performance of the BLAST implementation was assessed for the SILVA database. Both UNITE and SILVA datasets were compared to the qiime2-Naive-Bayes feature classifier (Bokulich *et al.*, 2018), the mothur Wang classifier, and the mothur k-nearest neighbors classifier with knn=3.

Classification Counts

Representative bacterial and fungal OTU sequences from Benucci et al. 2020 (Benucci *et al.*, 2020) were classified with the BLAST CONSTAX implementation at recommended settings with the suggested UNITE and SILVA databases. The conservative voting rule was applied for the bacterial library, but not for the fungal library, given the results observed with CPX trials.

Algorithm speed

Runtime was determined for both training and classification steps using printed timestamps 1) before calling the CONSTAX executable, 2) after training completion within the CONSTAX executable, written to STDOUT, and 3) after implementation of the CONSTAX executable.

Training was performed on a single core on an Intel(R) Xeon(R) CPU E5-2680 v4 @ 2.40GHz processor with 32 GB of requested memory. Each training database consisted of 500, 1000, 2000, 4000, 8000, or 16,000 sequence records sampled from the reference databases of the SILVA CPX test sets. Classification was performed with 1, 4, 8, 16, 32, 64, and 96 cores on a Intel(R) Xeon(R) CPU E7-8867 v4 @ 2.40GHz processor with 16 GB of requested memory, using 1000, 2000, or 4000 sequence records sampled from bacterial sequences in SILVA SSURef release 138. Training and classification were each performed with the default UTAX implementation or the “-b,--blast” BLAST implementation.

Definition of classification metrics

The classification performance framework from Edgar (2016) included the following classification performance metrics for clade-partition cross validation:

$$\text{Sensitivity} = TP/N_{\text{known}}$$

$$\text{Misclassification rate} = FP_{\text{mis}}/N_{\text{known}}$$

$$\text{Over – classification rate} = FP_{\text{over}}/N_{\text{novel}}$$

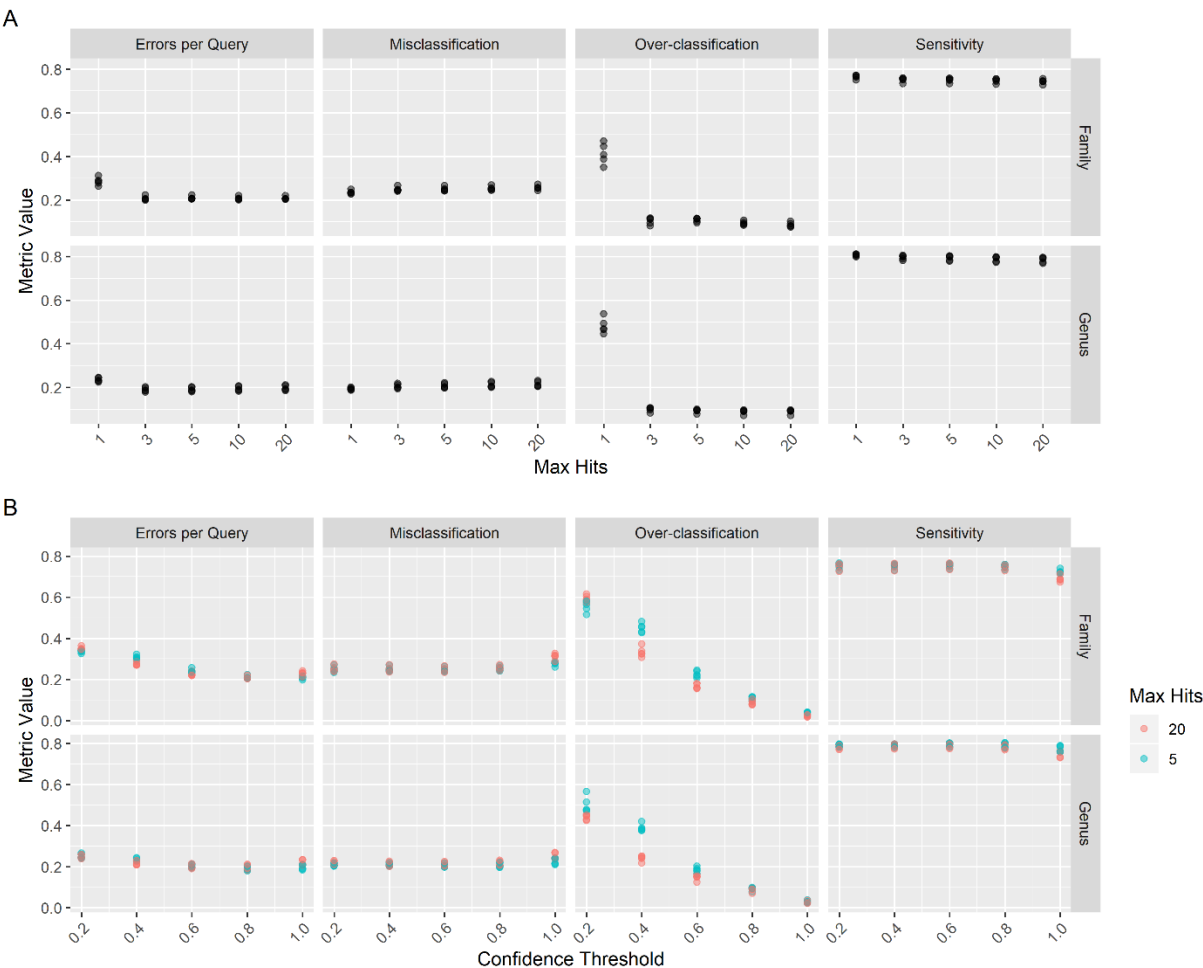
$$\text{Errors per query} = (FP_{\text{mis}} + FP_{\text{over}})/N$$

Where N_{known} and N_{novel} are the number of queries known (at a rank above or equal to the partition level) and novel (at a rank below the partition level), TP is the true positive predictions of known queries, FP_{mis} is the number of false positive predictions of known queries, and FP_{over} is the number of false positive predictions of novel queries. N is the total number of queries and the sum of N_{known} and N_{novel} .

Plotting and analysis

Data generated via CONSTAX testing runs were parsed and reorganized with Python scripts, and uploaded into R 3.6.1 (R Core Team, 2019) for analysis. Plotting and preparation of tables were performed with tidyverse 1.3.0 (Wickham, Averick, *et al.*, 2019), including tibble 3.0.5 (Müller and Wickham, 2019), tidyr 1.1.2 (Wickham and Henry, 2020), dplyr 1.0.3 (Wickham, François, *et al.*, 2019), and forcats 0.5.0 (Wickham, 2020), and ggplot2 3.2.1 (Wickham, 2016, 2). Patchwork 1.0.0 (Pedersen, 2019) and maditr 0.7.4 (Demin, 2020) were used for figure preparation. Classification performance metrics were compared between classifiers at each region, partition level, and database using a generalized mixed effects model with *glmer* function in lme4 1.1-21 (Bates *et al.*, 2015, 4), in which classifier and region are random effects and partition iteration is a fixed effect, and the metrics are modeled according to the binomial distribution. Pairwise comparisons were performed with emmeans 1.3.5 (Lenth, 2020) and multcomp 1.4-13 (Hothorn *et al.*, 2008). Several scripts involved the Python packages pandas (The pandas development team, 2020; McKinney, 2010), numpy (Harris *et al.*, 2020), and xlsxwriter (McNamara, 2021).

95 **Figures and Tables**



96

97 **Figure S1. Effects of max hits and confidence threshold parameters on UNITE**

98 **classification.** Errors per Query, Misclassification, Over-classification, and Sensitivity were

99 determined using Clade-Partition Cross Validation while varying the “--max_hits” (A) or “--conf”

100 (B) parameters on 1000 query sequences from the UNITE Fungi database. Confidence

101 threshold effects were compared at both 5 and 20 max hits.

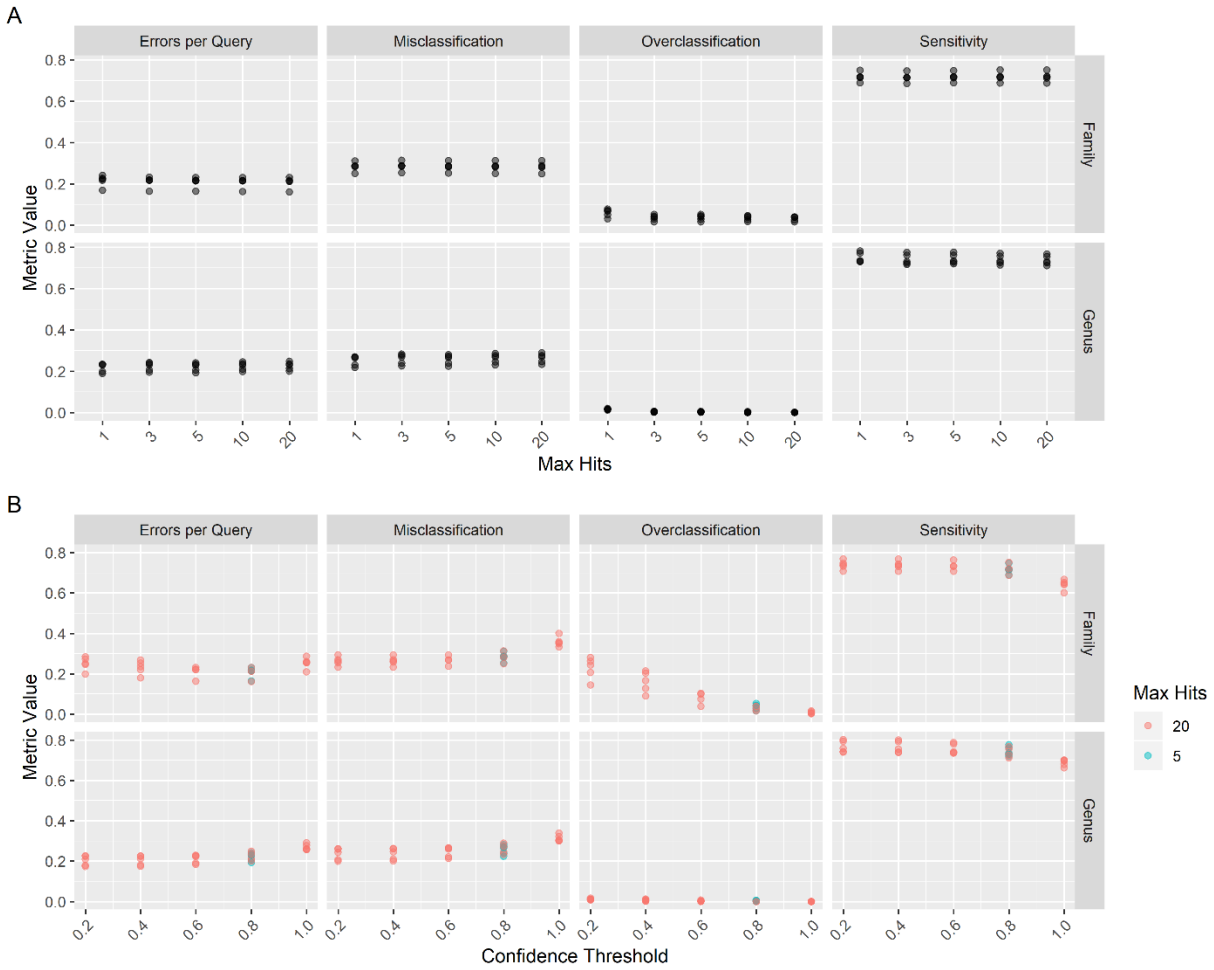


Figure S2. Effect of max hits and confidence threshold parameters on SILVA

classification. Errors per Query, Misclassification, Over-classification, and Sensitivity were determined using Clade-Partition Cross Validation while varying the “--max_hits” (A) or “--conf” (B) parameters on 1000 query sequences from the SILVA SSURef release 138.

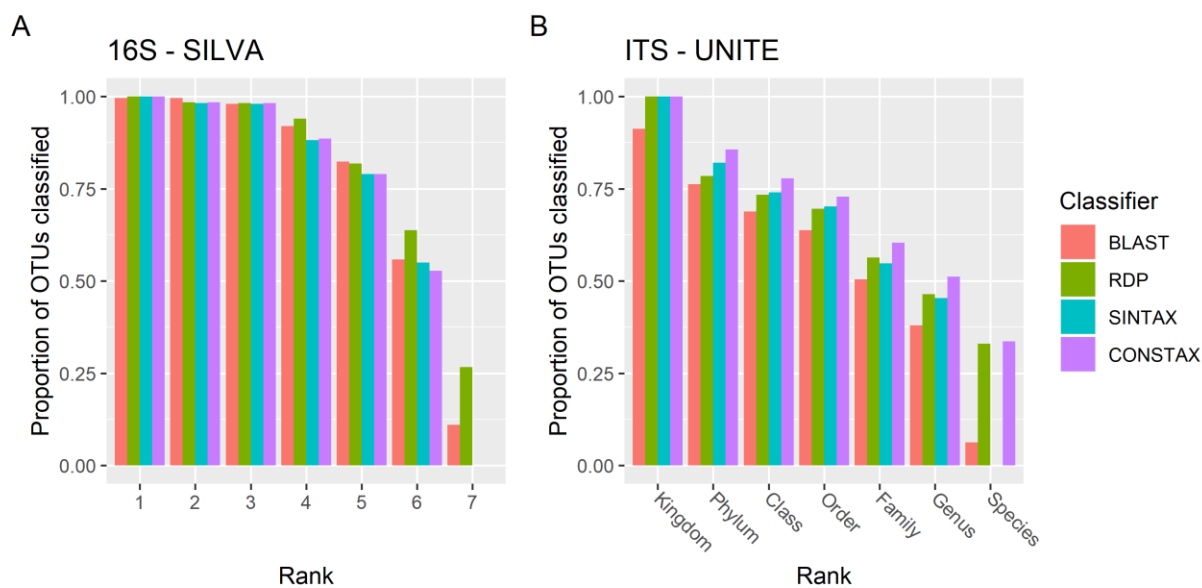


Figure S3. Classification counts for each classifier and the CONSTAX classification.

OTUs from Benucci et al. (2020), 500 each from bacterial and fungal libraries, which have been classified using databases for bacteria and fungi at recommended settings. Counts indicate the number of OTUs which had a taxon assigned at or above the confidence threshold of 0.8 at each rank. For bacteria, rank 1 corresponds to domain and decreases with higher rank numbers.

Table S1. Classification performance of each classifier, for each database, region, and partition level. Values are percentages: mean±SD, with entries sharing letters are not significantly different at $\alpha = 0.01$ for a given database, region, and partition level, as determined by a generalized linear mixed model using a binomial distribution, with region and classifier as random effects and partition iterations as a blocking effect. Performance metrics are defined in Supplementary Information. CB - CONSTAX with BLAST, CBC - CONSTAX with BLAST and conservative rule, CU - CONSTAX with UTX, CUC - CONSTAX with UTX and conservative rule.

Supplement References

- Apprill, A. *et al.* (2015) Minor revision to V4 region SSU rRNA 806R gene primer greatly increases detection of SAR11 bacterioplankton. *Aquat. Microb. Ecol.*, **75**, 129–137.
- Bates, D. *et al.* (2015) Fitting Linear Mixed-Effects Models Using {lme4}. *J. Stat. Softw.*, **67**, 1–48.
- Bengtsson-Palme, J. *et al.* (2013) Improved software detection and extraction of ITS1 and ITS2 from ribosomal ITS sequences of fungi and other eukaryotes for analysis of environmental sequencing data. *Methods Ecol. Evol.*, **4**, 914–919.
- Benucci, G.M.N. *et al.* (2020) Patient propagules: Do soil archives preserve the legacy of fungal and prokaryotic communities? *PLOS ONE*, **15**, e0237368.
- Bokulich, N.A. *et al.* (2018) Optimizing taxonomic classification of marker-gene amplicon sequences with QIIME 2's q2-feature-classifier plugin. *Microbiome*, **6**, 90.
- Demin, G. (2020) maditr: Fast Data Aggregation, Modification, and Filtering with Pipes and 'Data.table'.
- Edgar, R.C. (2016) SINTAX: a simple non-Bayesian taxonomy classifier for 16S and ITS sequences. *bioRxiv*, 074161.
- Harris, C.R. *et al.* (2020) Array programming with NumPy. *Nature*, **585**, 357–362.
- Hothorn, T. *et al.* (2008) Simultaneous Inference in General Parametric Models. *Biom. J.*, **50**, 346–363.
- Lenth, R.V. (2020) emmeans: Estimated Marginal Means, aka Least-Squares Means.
- McKinney, W. (2010) Data Structures for Statistical Computing in Python. In, Walt, S. van der and Millman, J. (eds), *Proceedings of the 9th Python in Science Conference.*, pp. 56–61.
- McNamara, J. (2021) Creating Excel files with Python and XlsxWriter.
- Müller, K. and Wickham, H. (2019) tibble: Simple Data Frames.
- Parada, A.E. *et al.* (2016) Every base matters: assessing small subunit rRNA primers for marine microbiomes with mock communities, time series and global field samples. *Environ. Microbiol.*, **18**, 1403–1414.
- Pedersen, T.L. (2019) patchwork: The Composer of Plots. *R Package Version 100*.
- R Development Core Team 3.0.1. (2013) A Language and Environment for Statistical Computing. *R Found. Stat. Comput.*, **2**, <https://www.R-project.org>.
- The pandas development team (2020) pandas-dev/pandas: Pandas Zenodo.
- Van Der Pol, W.J. *et al.* (2019) In Silico and Experimental Evaluation of Primer Sets for Species-Level Resolution of the Vaginal Microbiota Using 16S Ribosomal RNA Gene Sequencing. *J. Infect. Dis.*, **219**, 305–314.
- Wickham, H., François, R., *et al.* (2019) dplyr: A Grammar of Data Manipulation.
- Wickham, H. (2020) forcats: Tools for Working with Categorical Variables (Factors).
- Wickham, H. (2016) ggplot2: Elegant Graphics for Data Analysis Springer-Verlag New York.
- Wickham, H., Averick, M., *et al.* (2019) Welcome to the tidyverse. *J. Open Source Softw.*, **4**, 1686.
- Wickham, H. and Henry, L. (2020) tidyr: Tidy Messy Data.