A standardized, extensible framework for optimizing classification improves marker-gene taxonomic assignments

Nicholas A. Bokulich[1], Jai Ram Rideout[2], Evguenia Kopylova[3], Evan Bolyen[2], Jessica Patnode[4], Zack Ellett[5], Daniel McDonald[6,7], Benjamin Wolfe[8], Corinne F. Maurice[8,9], Rachel J. Dutton[8], Peter J. Turnbaugh[8,10], Rob Knight[3,11], J. Gregory Caporaso[2,4,5,#]

[1] Department of Medicine, New York University Langone Medical Center, New York, NY 10010, USA

[2]Center for Microbial Genetics and Genomics, Northern Arizona University, Flagstaff, AZ, USA

[3]Department of Pediatrics, University of California, San Diego, CA, USA

[4]Department of Biological Sciences, Northern Arizona University, Flagstaff, AZ, USA

[5]Department of Computer Science, Northern Arizona University, Flagstaff, AZ, USA

[6]Department of Computer Science, University of Colorado, Boulder, CO, USA

[7]BioFrontiers Institute, University of Colorado, Boulder, CO, USA

[8]FAS Center for Systems Biology, Harvard University, Cambridge, MA, USA

[9] Department of Microbiology & Immunology Department, Microbiome and Disease Tolerance Centre, McGill University, Montreal, Quebec, Canada.

[10]Department of Microbiology and Immunology, G.W. Hooper Foundation, University of California San Francisco, 513 Parnassus Ave, San Francisco, CA, USA

[11]Department of Computer Science and Engineering, University of California, San Diego, CA, USA

#Corresponding author

Gregory Caporaso

Department of Biological Sciences

1298 S Knoles Drive

Building 56, 3rd Floor

Northern Arizona University

Flagstaff, AZ, USA

(303) 523-5485

(303) 523-4015 (fax)

Email: gregcaporaso@gmail.com

Abstract

Background: Taxonomic classification of marker-gene (i.e., amplicon) sequences represents an important step for molecular identification of microorganisms.

Results: We present three advances in our ability to assign and interpret taxonomic classifications of short marker gene sequences: two new methods for taxonomy assignment, which reduce runtime up to two-fold and achieve high-precision genus-level assignments; an evaluation of classification methods that highlights differences in performance with different marker genes and at different levels of taxonomic resolution; and an extensible framework for evaluating and optimizing new classification methods, which we hope will serve as a model for standardized and reproducible bioinformatics methods evaluations.

Conclusions: Our new methods are accessible in QIIME 1.9.0, and our evaluation framework will support ongoing optimization of classification methods to complement rapidly evolving short-amplicon sequencing and bioinformatics technologies. Static versions of all of the analysis notebooks generated with this framework, which contain all code and analysis results, can be viewed at http://bit.ly/srta-012.

Background

High-throughput amplicon-sequencing methods have opened new frontiers in microbial ecology, transforming our understanding of complex microbial ecosystems ranging from our bodies(*1*) to our planet(*2*). Sequencing 'universal' marker genes (e.g., bacterial 16S rRNA and fungal internal transcribed spacer (ITS) amplicons) and comparing those sequences to annotated reference sequences allows complex biological communities to be characterized taxonomically. Many taxonomic classification algorithms have been developed, but different methods can provide markedly different results, even when the same query sequences and reference database are used(*3*).

The problem of taxonomic classification of marker genes is described as follows. Given a short, possibly error-containing, fragment of a marker gene, the goal is to determine the taxonomy of the organism from which that gene was derived with the greatest possible taxonomic specificity. Accurate and specific taxonomic assignment of these reads is essential for many — but not all — aspects of microbiome analysis, but currently used methods have not been optimized on "modern" datasets (e.g., short-amplicon sequencing reads, here we used reads varying in length from 100-250 bases as described in Supplementary Table 1).

Introducing a new taxonomy classification method requires benchmarking against pre-existing methods to determine whether the new method is more computationally efficient

(e.g., faster and/or smaller memory requirements) and/or better than other methods (e.g., yields more specific taxonomic assignments and/or more sequences accurately classified). When comparing a new method to existing methods, developers must:

- identify and obtain test datasets;
- develop an evaluation framework;
- obtain and install pre-existing taxonomic assignment software; and
- determine the parameters to benchmark against in the pre-existing taxonomic assignment software.

These steps are fundamental to any methods development project, but all are subject to the developers' decisions and biases. Additionally, because the test data and evaluation framework are often not published, when a subsequent method is introduced all of these steps must be repeated by its author. This results in duplicated effort and inconsistent evaluation metrics, such that method benchmarks are often not directly comparable. Each new method is evaluated with a new, custom benchmarking pipeline.

To address these needs, we developed a computational framework for evaluating taxonomic classifiers using standardized public datasets (Figure 1), and used it to compare the performance of existing and newly developed taxonomy-classification methods. This framework will be easily applied to new methods in the future by any bioinformatics method developer. Here we apply the framework to compare the performance of four marker-gene-agnostic taxonomy classifiers (i.e., those that can be trained on a reference database of any marker gene). Two of these are pre-existing classifiers, the RDP Classifier

(*4*) (version 2.2) and QIIME's legacy BLAST-based classifier (version 2.2.22) (*5, 6*), which uses the MegaBlast parameter setting (`-n T`) and assigns the taxonomy of the single top BLAST hit. The other two classifiers are based on UCLUST (version 1.2.22q) (*7*) and SortMeRNA (*8*) (version 2.0 29/11/2014), which are newly implemented in QIIME and evaluated here for the first time. All of these are heuristic approaches to taxonomic classification.

Our framework is available on GitHub at https://github.com/gregcaporaso/short-read-tax-assignment and was used to generate all analysis results and figures in this paper (with the exception of the schematic in Figure 1). To illustrate the extensibility of our framework, we performed a subset of the analyses using the mothur (version 1.35.1) implementation of the RDP Classifier, and provide detailed descriptions in the GitHub repository of the steps we took to add this classifier that will provide users with a clear example of how to evaluate other classifiers. We executed this framework on the QIIME 1.9.0 Amazon Web Services (AWS) virtual machine image, making the full analysis easily reproducible (see the repository's README.md file). This approach is similar to that taken in several recent "executable papers" (*9-11*) (also see http://www.executablepapers.com/).

PeerJ PrePrints | https://dx.doi.org/10.7287/peerj.preprints.934v2 | CC-BY 4.0 Open Access | rec: 10 Jul 2015, publ: 10 Jul 2015

Results

Standardized and extensible evaluation of taxonomic classifiers

Our evaluation framework differs from others that we are aware of in that it can be reused to support standardized and extensible evaluation of taxonomic classifiers. In this context, we consider a standardized evaluation framework to incorporate a consistent and appropriate set of metrics, defined in the same way (e.g., specificity, sensitivity, accuracy) with sufficient unit testing (for a discussion of unit testing, see (*12*)), and a consistent and appropriate set of data. Multiple datasets are needed to minimize overfitting. An extensible evaluation framework is one where it is easy for *users* (not only the initial developers) to add new methods, add new reference databases, and add new datasets and metrics. Testing new methods with standardized datasets and evaluation metrics will relieve developers of the time needed to perform these otherwise redundant steps, and allow direct comparison of new and pre-existing methods without repeating benchmarking efforts. Extensibility enables a framework to evolve over time, incorporating useful new datasets and metrics that meet the needs of changing methods and technologies and allowing conclusions to be rapidly updated in the light of new data.

Our evaluation framework is based on IPython Notebooks(*13*), which facilitate generation of interactive, reproducible, and executable reports, and uses scikit-bio, pandas, and open-source BSD-compatible software developed for this study (https://github.com/gregcaporaso/short-read-tax-assignment/). This framework

evaluates method and parameter combinations based on their performance on standardized amplicon sequence datasets derived from simulated communities and Illumina sequencing of artificially constructed (mock) communities. Simulated communities, where sequences are compiled from reference databases, allow us to assess the "best-case" performance of classifiers in the absence of real-world issues, such as sequencing and PCR bias and error. Mock communities(*14*) (in this case, precise combinations of 12 to 67 species; Supplementary Figures 1-5) allow us to quantitatively and qualitatively assess the accuracy of taxonomic profiles, since the actual community composition is known in advance, while retaining experimental issues that are difficult to model accurately. Our evaluation framework currently utilizes 10 mock communities to minimize overfitting to conditions specific to experimental conditions or community compositions.

The framework currently employs the following strategies for evaluating and comparing classifier performance:

1) Precision, recall, and F-measure scores are qualitative metrics, only assessing the accuracy of taxonomic composition but not abundance.
2) Pearson (*r*) (*15*) and Spearman correlation coefficients (*rho*) (*16*) are quantitative measures of accuracy, incorporating accuracy of taxonomic composition as well as abundance.
3) Computational runtime is measured for each classifier as a function of reference database size and number of query sequences. (All runtime computations should

be performed on the same system with a single job running at a time to control for runtime variance. The results presented here were all performed on a single AWS instance.)

The framework identifies optimal parameter configurations for each classification method, the top-performing methods and parameter configurations for each test dataset, and generates publication-quality figures illustrating evaluation scores and distributions for each method, configuration, and dataset.

Our evaluation framework, test datasets, and pre-computed taxonomic assignment results (for the methods and parameter combinations presented here) are hosted on GitHub, an online software revision control and collaboration tool (for a discussion of revision control and its importance, see (*12*)). This provides a major benefit that should drive widespread adoption of this strategy in bioinformatics method evaluations: our analysis is not static. A developer of a new taxonomic classifier (call it *Classifier X*) can download our test dataset, generate taxonomic assignments for all simulated and mock communities, and quickly assess how *Classifier X* compares to pre-existing methods. If the developer determines that *Classifier X* is promising, they can submit the classification results to our repository as a Pull Request. The *Classifier X* results can then be merged with the pre-existing results, so that future methods developers can evaluate their tool in the context of pre-existing methods, which will now include the results generated by *Classifier X*. The evaluation of future methods therefore uses the same data used here (although new test datasets can

also be added using Pull Requests), and evaluators need not have working installations of all pre-existing methods, which can be difficult to configure and install.

We applied this evaluation framework to compare the performance of two commonly used, pre-existing taxonomy classifiers (RDP Classifier(*4*) and BLAST(*5*)) across multiple parameter settings, and two new taxonomic classifiers presented here for the first time. The first is an adaptation of SortMeRNA (*8*), and the second is based on UCLUST (*7*) (see methods for more details on code, the taxonomic assignment algorithms, and data availability).

Performance of classifiers on bacterial and fungal mock communities

We first evaluated assignment accuracy of all classification methods using mock communities. As expected, assignment accuracy for all methods decreased with increasing assignment depth (Figure 2). From phylum to family level, different assignment methods (with optimized parameters) performed similarly, but selection became important for accurate genus- and species-level assignments. SortMeRNA achieved the highest precision and F-measures at these levels, though RDP yielded better recall scores and correlation coefficients for most mock communities (Figures 2-3; Supplementary Figures 6-7). RDP recall and correlation coefficients performed best at low confidence thresholds ($c < 0.5$), though precision and F-measure improved with increasing assignment confidence (Figure 3; Supplementary Figures 6-7). UCLUST was not the top performer for any metric but

demonstrated good balance between recall (better than SortMeRNA) and precision (better than RDP), thus consistently delivering high-quality assignments (Figures 2-3; Supplementary Figures 6-7). BLAST assignments generally performed worse than low-confidence RDP assignments of bacteria for all evaluation metrics except for Pearson *r*, but performed similarly to RDP for fungal assignments (Figure 3; Supplementary Figures 6-7). SortMeRNA performed best for fungal community assignments, delivering top precision, recall, and F scores through species level (Figure 3; Supplementary Figures 6-7).

Performance of classifiers on bacterial and fungal simulated communities

The mock communities currently available in the framework resemble human fecal (datasets B1-B8) or cheese microbial communities (datasets F1-F2), so only contain select clades. Thus, we tested classifier performance on simulated sequencing reads derived from the entire Greengenes (*17*) and UNITE (*18*) databases, representing all taxa currently annotated in those databases. Taxonomy assignments of simulated communities exhibited similar trends, indicating that classifiers and configurations performed similarly across taxa (Figure 4; Supplementary Figure 8). Simulated communities were generated from reference sequences randomly selected from the reference databases (see *Methods* for details). Taxonomy classifications were then made against either the remaining sequences in the reference database (as in a cross-validation scheme, referred to as "partial reference" classification, so the query sequences would not be included in the reference database) or

the full reference database (which includes exact matches to the query sequences, but ensures that all taxa present in the query data are represented in the reference database). The partial reference classification simulates a typical sequence assignment scenario, whereby many query sequences may not have perfect matches in the database. The full reference classification is still informative, as some sequences removed from the partial reference database may represent unique taxonomic lineages without nearby matches, particularly within the fungal database.

Similar classifier performance and optimization behaviors were observed for classification of simulated communities (Figure 4; Supplementary Figure 8). Species-level assignment performed well on simulated sequences for bacteria, using both partial and full reference databases. Surprisingly, classification precision approached perfect scores for most methods, but precision was progressively degraded by low RDP confidence thresholds (c), SortMeRNA best alignments (b) = 1, and UCLUST max accepts (a) = 1 (Figure 4). As expected, precision demonstrated an inverse relationship to recall for simulated reads, and the same parameters that minimized precision maximized recall. Fungal assignment accuracy suffered using the partial reference database, indicating that many sequences removed for classification represent unique lineages: as expected, when the single sequence representing a specific taxonomic group (e.g., a species) was removed from the reference database, it was no longer possible to classify that organism at that taxonomic level, but only at higher taxonomic levels (e.g., its genus). On average, the approximate fraction of species represented in the full database were not represented in the partial

database was 3.7% for bacteria and 5.8% for fungi. These data are presented in Supplementary Figure 9 for all taxonomic levels. Fungal assignment accuracies using the full reference database mirrored the bacterial assignments, and the same methods were optimal for both. Using the partial reference database, optimized UCLUST and SortMeRNA classifications yielded the best F-measure scores for bacterial communities; RDP and mothur tied as best for fungal communities (Table 1). Using the full reference database, UCLUST yielded best precision, recall, and F-measure scores for full-length simulated communities, while SortMeRNA performed best for 100-nt simulated reads (Table 1).

Classifier parameter optimization is essential

Parameter optimization was essential for all methods (Figures 3-4; Table 2, Supplementary Figures 6-7). For example, the F-measure of RDP results ranged from very low (with c ≥ 0.8) to among the best (with c ≅ 0.4-0.5) with different confidence threshold settings on our simulated dataset (Figure 4, Table 2). These values align with the current recommendation for RDP-based classification of very short amplicon sequences (c = 0.5) (http://rdp.cme.msu.edu/classifier/class_help.jsp#conf) (*3, 19*). SortMeRNA and UCLUST displayed similar behavior, where different parameters resulted in very different method performance, and performed best with slightly reduced similarity thresholds (s= 0.8 for both). Only BLAST was relatively unaffected by different parameter configurations (in this case, e-values, which in experiments performed on a subset of test data due to practical

runtime constraints we varied to as low as 1e-30). Parameter sweeps of this type should therefore be considered an essential component of bioinformatics methods comparisons.

Classifier runtime

Classifier choice also substantially impacted computational runtime, as a function of both query sequence count and reference sequence count. Many classification methods first create an index of the reference sequence database prior to classifying query sequences, and thus indexing time in an important measure of a classifier's scalability to both large and small reference databases. To measure the effects of reference sequence database size on runtime, a single query sequence is searched against the reference database (89,339 total sequences of 302 ± 88 nt (mean ± SD)). This tells us how long it takes to assign taxonomy to the first query sequence, and therefore provides a measure of time needed to index the reference. To measure the effects of query sequence count on runtime, increasing numbers of query sequences (up to 998,300 total sequences of 302 ± 88 nt (mean ± SD)) were classified using the same reference sequence database. Since database indexing is included in all of these steps, we care most about the slope of the line and very little about the y-intercept (which represents how long the database takes to index, and is typically a process that can be performed once and the index re-used).

UCLUST delivered the fastest query assignment (Figure 5A) and reference sequence indexing times (Figure 5B), and the smallest slope for query sequence assignment (Figure 5A). BLAST also demonstrated very little runtime increase in response to reference

database size, but exhibited a phenomenally high slope for query sequence count, indicating very slow performance relative to the other classifiers (Figure 5). RDP and SortMeRNA displayed comparable query sequence assignment times and slopes, but SortMeRNA required the most indexing time of any method, exhibiting increasing runtimes proportional to reference sequence count (Figure 5A). RDP required more indexing time than BLAST and UCLUST, but demonstrated little runtime increase as more reference sequences were added (Figure 5A).

Applying the framework to other classifiers

To illustrate the process of evaluating a "new" method in our framework, we tested the performance of the mothur classifier (a commonly used implementation of the RDP Classifier). We adapted one of our IPython Notebooks to generate taxonomic assignments using mothur 1.35.1, and then performed all of the simulated data analyses using mothur with small modifications to the evaluation notebooks. This analysis very clearly illustrates that the mothur classifier achieves nearly identical results to the RDP Classifier, as would be expected. These data are presented in Tables 1-2 and Supplementary Figure 10, and the notebooks are included in the `mothur-evaluation` directory of the GitHub repository. Documentation exists in that directory so it can serve as an example of how to apply the framework to other classifiers in the future.

Discussion

Evaluating methods for taxonomic classification of different marker genes and read lengths is critical for interpreting taxonomic assignments, allowing us to determine if assignments at a specific taxonomic level, or for a specific taxonomic group, are reliable. Additionally, different features of taxonomic classifier performance may be important for different applications. For example, for a medical diagnosis based on indicator taxa, false positives may be preferred over false negatives (i.e., optimizing recall at the expense of precision) if a cheap, minimally invasive treatment is available. However, for forensic applications, false negatives may be preferred over false positives (i.e., optimizing precision at the expense of recall), if a false positive could lead to a wrongful conviction. For more general applications, users likely prefer balanced precision and recall (i.e., optimized F-measure), and we base our parameter recommendations on this assumption. Thus, understanding classifier performance and limitations, and how to optimize classifiers for different applications, are essential for the usefulness of these methods.

Here we show that after parameter optimization, classification methods perform similarly for assigning taxonomy from phylum to family level, but performance decreased to different degrees for all methods at genus and species levels. This reflects the limitations of accurately classifying short marker-gene sequences, but indicates that some methods/configurations (detailed below) are superior for handling short reads. Using longer read lengths and less conserved marker genes, these performance differences may

become more or less distinct. Thus, methods should be re-evaluated as sequencing technologies continue to advance and longer read lengths become available. Our extensible evaluation framework easily supports the addition of test data sets, so these same method benchmarks could be performed on updated test data in the future.

Performance also varied for fungal ITS and bacterial 16S rRNA sequences, indicating that no method is universally superior across organisms and marker genes. SortMeRNA (with optimized parameters, described below) delivered the most accurate fungal assignments to species level, even outperforming RDP for recall and correlation coefficients. However, RDP yielded better recall scores for bacterial sequences, indicating that performance depends on the target marker gene (and its corresponding reference database). As next-generation sequencing read lengths continue to grow, larger sections of the bacterial 16S rRNA and other marker genes will become promising targets for amplicon sequencing but may impact classifier performance. In addition, protein-coding genes have high potential for strain-level and functional profiling of microbial communities, but likely alter classifier behavior. Assignment methods should be re-evaluated for other existing and new marker genes and continuously be re-optimized with each update in molecular targets and sequence technology to maximize performance.

Optimal method selection ultimately depends upon the priorities of the end user. For bacterial 16S rRNA and fungal ITS sequences, as analyzed here, no method is universally superior under all conditions or for all evaluation metrics. For maximum precision and F-

measure scores, high-confidence RDP (c=1.0), UCLUST, and SortMeRNA perform best. For top recall and correlation coefficients, lower-confidence RDP (c≤0.6) performs best. For a consistent balance of recall and precision, UCLUST and medium-confidence RDP (c=0.4–0.6) are reliable choices. For fungal ITS assignments, SortMeRNA performs best across all metrics, though RDP, UCLUST, and BLAST also perform well. When runtime is an important factor, such as with very large or high-diversity datasets, UCLUST performs faster than all other methods benchmarked here. Based on assignment accuracies, mock community reconstruction, computational runtime, simulated read classification, and availability of source code and minimal licensing restrictions, we recommend using the SortMeRNA classifier, with 0.51 consensus, 0.8 similarity, and a maximum 1 best alignment, 0.8 coverage, and an e value of 0.001. All methods tested here (and some additional methods) are also included and configurable in QIIME 1.9.0 to support the needs of QIIME's diverse user base. The supplementary IPython Notebooks available in the project repository (and for static viewing at http://bit.ly/srta-012) contain detailed results on all parameter configurations tested here.

Our current evaluation framework is designed for comparison of taxonomy classifiers, but the standardized mock communities and evaluation metrics would be equally useful for optimizing other aspects of short-amplicon sequence analysis and is readily adaptable for this purpose. This includes bioinformatics processing steps, such as quality filtering (*14*), OTU picking (*20*), paired-end and overlapping read alignment, and chimera filtering. The evaluation framework could also allow comparison of bench techniques that impact

microbial detection, such as sample storage/handling (*21*), DNA extraction (*21*), PCR (*22*), library preparation, sequencing platforms/technologies, reagent contamination (*23*), and technical precision by individual users, between users, and between laboratories. Any pre-sequencing comparisons would require the generation of new, standardized mock communities, but these would in turn enrich our growing database of mock communities, a public asset that will support ongoing improvements in sequence analysis techniques, and could easily be added to the framework.

The optimal methods for taxonomic assignment of a given marker gene or sequencing technology is unlikely to generalize across marker genes or sequencing technologies. The evaluation framework used here is not specific to a single marker gene, but instead provides immediately applicable information for optimizing taxonomic assignment of 16S and ITS sequences generated on the Illumina platforms, and can be adapted to the rapidly evolving needs of the next-generation sequencing community. The evaluation framework (Figure 1) facilitates iterative re-evaluation of these conditions as new classification methods, sequencing read lengths, marker-gene targets, and sequencing chemistries are released, and as additional metrics of performance are desired. We hope that this will become a model for standardized, extensible evaluation frameworks for bioinformatics method comparisons.

Methods

Data availability

Sequence data used in this study are publicly available in the Qiita database (http://qiita.microbio.me/) under the study identities listed in Supplementary Figure 1. Raw data are also available via links our GitHub repository: https://github.com/gregcaporaso/short-read-tax-assignment/blob/0.1.2/data/raw-data-urls.txt. All other data generated in this study, and all new software, is available in our GitHub repository under the BSD license. Our GitHub repository can be found at: https://github.com/gregcaporaso/short-read-tax-assignment.

Data Analysis

All analyses were performed using QIIME 1.9.0 on the QIIME 1.9.0 AWS Virtual Machine Image (AMI: ami-ea2a7682) and the taxonomy classifier comparison workflow hosted on GitHub: https://github.com/gregcaporaso/short-read-tax-assignment (tag: 0.1.2). Static versions of all of our analysis notebooks, which contain all code and analysis results, can be viewed at http://bit.ly/srta-012. All specific notebooks referenced below can be viewed via this page.

Mock communities

Mock communities analyzed in this study were generated by 10 separate sequencing runs on the Illumina GAIIx (n = 2), HiSeq2000 (n = 5), and MiSeq (n = 4) (Supplementary Figure 1). These consisted of genomic DNA from known species isolates deliberately combined at defined rRNA copy-number ratios (Supplementary Figure 1). These sequencing runs were performed on different instruments at different sites—Illumina Cambridge Ltd (datasets B4, B6), Broad Institute (datasets B3, B5), Washington University School of Medicine (datasets B1-B2), and Harvard FAS Center Core Facility (datasets B7-B8, F1-F2)— with the goal of assessing the impact of filtering parameters across a broad set of sequencing conditions.

DNA extraction, PCR, and sequencing for all sequencing runs were described previously (*24*). The only sample collections not published previously (mock communities F1-F2) had DNA extracted using the PowerSoil kit (MoBio) according to manufacturer instructions. PCR amplifications were performed in duplicate using primers ITS1-F (5'-*AATGATACGGCGACCACCGAGATCTACAC*TATGGTAATT**CT CTTGGTCATTTAGAGGAAGTAA**-3') and ITS4 (5'-*CAAGCAGAAGACGGCATACGAGAT*NNNNNNNNNNNNAGTCAGTCAGAT**GCTGCGTTCTTCATC GATGC**-3') (*24*). Both oligonucleotides consisted of the actual primer sequence (boldface text) with an Illumina adapter (italics), pad, and linker sequences (both underlined) and a 12-nt Golay error-correcting barcode (*25*) (represented by a poly-N region) in the reverse primer. Reaction conditions consisted of denaturing at 98˚C for 3 min, followed by 30

cycles of 98˚C for 45 s, 50˚C for 30 s, and 72˚C for 45 s, followed by a final extension of 72˚C for 5 min. Pooled amplicons were purified using the AMPure XP kit (Agencourt), quantified with Picogreen reagent (Invitrogen), combined at equimolar ratios, and gel purified (cutting out bands between 100-500 bp) using the Gel Gen Elute Gel Extraction kit (Sigma-Aldrich) prior to sequencing.

Raw Illumina fastq files were de-multiplexed, quality-filtered, and analyzed using QIIME (v. 1.6.0-dev)(*6*). Reads were truncated at any site of more than three sequential bases receiving a Phred quality score < 20, and any read containing ambiguous base calls or barcode/primer errors were discarded, as were reads with < 75% (of total read length) consecutive high-quality base calls (*14*). Operational taxonomic units (OTUs) were assigned using the QIIME open-reference OTU-picking pipeline with the UCLUST-ref(*7*) wrapper. After prefiltering sequences with > 60% dissimilarity from its closest match in the reference database (listed below), sequences were clustered into OTUs based on their closest match in the reference collection with greater than 97% pairwise nucleotide sequence identity (97% ID). Sequences which failed to hit a sequence in the reference database at 97% ID we subsequently clustered *de novo*. The cluster centroid for each OTU was chosen as the OTU representative sequence.

Simulated communities

The simulated communities used here were derived from the reference databases using the "Simulated community analyses / Simulated community generation" notebook in our

project repository. Beginning with a full reference database (either Greengenes or UNITE), 10% of the sequences were extracted at random and the corresponding primers were used to simulate amplification and slice out either the full region between those primers (B1 and F1) or the first 100 bases downstream (3') of the forward primer (B2 and F2). The bacterial primers used were 515F/806R (*26*), and the fungal primers used were BITSf/B58S3r (*27*). The remaining 90% of the full-length database sequences were used as the "partial reference" database, and all of the database sequences were used as the "full reference" database. This process was performed in five iterations to generate twenty different simulated communities (five each of B1, B2, F1 and F2).

Taxonomy classification

OTU representative sequences were classified taxonomically using QIIME-based wrappers of the following taxonomy classifiers and confidence settings:

1. Ribosomal Database Project (RDP) naïve Bayesian classifier(*4*), using variable confidence thresholds (c) for taxonomic assignment between c = 0.0 to c = 1.0 in steps of 0.1.

2. BLAST(*5*) using e-value thresholds (e) for taxonomic assignment of e = 1e-9, 0.001, and 10000.0.

3. SortMeRNA(*8*) with the following parameters: minimum consensus fraction (f), similarity (s), best N alignments (b), coverage, and e value. See description below.

4. UCLUST(*7*) with the following parameters: minimum consensus fraction (f), similarity (s), and maximum accepts (a). See description below.

Reference Databases

The bacterial and archaeal 16S rRNA reference sequence database for OTU picking and taxonomy-classifier retraining was the Greengenes 13_8 16S rRNA gene database(*17*) preclustered at 97% ID.

The fungal ITS reference sequence database for OTU picking and taxonomy-classifier retraining was the UNITE+INSD database (9-24-12 release)(*18*) prefiltered at 97% ID, and from which sequences with incomplete taxonomy strings and empty taxonomy annotations (e.g., uncultured fungus) were removed, as described previously(*27*).

Runtime analyses

Taxonomy classifier runtimes were logged while performing assignments of the same random subset of 16S rRNA sequences, following the workflow described above. All runtimes were computed on a single AWS instance to control for runtime variance across cloud instances, and only one assignment process was run at a time during runtime benchmarking.

The exact commands used for runtime analysis are presented in the "Runtime analyses" notebook in the project repository.

Performance analyses using mock and simulated communities

Precision, recall and F-measure are used for qualitative compositional analyses of mock and simulated communities.

At a given taxonomic level, a taxonomic assignment is a:

- true positive (*TP*), if that taxonomic assignment is present in the results and in the mock community
- false positive (*FP*), if that taxonomic assignment is present in the results, but is not present in the mock community
- false negative (*FN*), if a taxonomic assignment is not present in the results, but is present the mock community
- true negative (*TN*), if a taxonomic assignment is not present in the results, and is not present the mock community

Classic qualitative methods for evaluating the retrieval of expected observations—in this case expected taxa—are precision, recall, and F-measure. Here these are defined as:

- *Precision=TP/(TP+FP)* or the fraction of taxonomic assignments that actually matches members of the mock community
- *Recall=TP/(TP+FN)* or the fraction of the mock community members that were observed in the results

- *F-measure* = 2 × Precision × Recall / (Precision + Recall)

Thus, precision and recall represent the relative recovery of relevant observations and expected observations, respectively. F-measure is the harmonic mean of both these scores, providing a balanced metric for simultaneously considering each score as an overall measure of accuracy. These three measures were used to evaluate the accurate recovery of expected taxa in sequenced mock communities and simulated communities, without regards to taxon abundance (i.e., qualitative).

Pearson and Spearman correlations are used for quantitative compositional analyses of mock and simulated communities (*15, 16*). At a given taxonomic level, these compute the correlation between the relative abundances of the taxa as predicted by the taxonomy assigner, and the known community compositions.

Mock communities cannot be considered accurately assigned on the basis of detection of expected species (i.e., qualitatively) alone. As defined collections of microbial species, assignment accuracy must be judged both on recovery of expected taxa and on the reconstruction of expected community composition. In other words, a good classification method should identify the expected community members in their known abundances. We compute this as the correlation between the relative abundances of observed taxa with their expected abundances as added to the mock community. The ideal correlation (r = 1.0) is highly unlikely under real-world conditions as, in addition to taxonomy misclassifications, primer bias, contamination, PCR error, sequencing error, copy number variation, and other procedural artifacts may all theoretically skew observations.

The exact commands used for the mock community and simulated community analyses are presented in the "Mock community analyses" and "Simulated community analyses" notebooks in the project repository.

UCLUST-based and sortmerna-based consensus taxonomy assigners

We introduce two new methods for taxonomic assignment, based on the uclust and sortmerna software packages.

Our UCLUST-based taxonomy assigner (which differs from utax (http://drive5.com/usearch/manual/utax_algo.html)) is available in QIIME 1.9.0 (assign_taxonomy.py and parallel_assign_taxonomy_uclust.py). Although UCLUST itself is not open source or free, it is licensed for free use with QIIME. QIIME's uclust-based taxonomy assigner is open source, though it makes calls to uclust (note: the version implemented here is an older version than the current, closed-source USEARCH version). Internal to QIIME, query sequences are searched against the reference database with the command:

```
uclust --libonly
      --allhits
      --id <similarity>
      --maxaccepts <max-num-results>
      --input <input-file-query-fasta>
      --lib <input-file-reference-fasta>
```

```
        --uc <output-file-uc>
```

The `ma` and `similarity` values (in addition to `input-file-*` and `output-file-*`) are specified by the user.

Our sortmerna-based taxonomy assigner is also available in QIIME 1.9.0. sortmerna is open source, as is the QIIME wrapper that adapts this for taxonomic assignment. Internal to QIIME, query sequences are searched against the reference database with the command:

```
sortmerna --ref <input-file-reference-fasta>,<input-file-reference-index>
        -e <e-value>
        --aligned <output-file-bl9>
        -a 1
        --print_all_reads
        --log
        --blast 3
        --reads <input-file-query-fasta>
        -v
        --best <max-num-results>
```

The `e-value and max-num-results values (in addition to input-file-* and output-file-*) are specified by the user.`

Both of these classifiers can potentially return up to `max-num-results` database hits per query sequence. Taxonomic classification of query sequence is then performed by computing a consensus assignment from those query results. This is achieved by starting at the highest taxonomic level (domain, in Greengenes for example) and determining if the classification at that level is present in at least `min_consensus_fraction` of the query results, where `min_consensus_fraction` is a user-defined value (default is `0.51`, based on the results of our analyses). If so, the query sequence is given that classification at that level, and the classification of the query results are compared at the next taxonomic level. Once a classification is identified that is not present in at least of the `min_consensus_fraction` query results, the taxonomic classification for the query sequence is truncated. For example, if a query sequence *q1 hit to three query results in the reference database with the classifications:*

d__Bacteria; p__Proteobacteria; c__Alphaproteobacteria

d__Bacteria; p__Proteobacteria; c__Gammaproteobacteria

d__Bacteria; p__Cyanobacteria; c__Oscillatoriophycideae

if `min_consensus_fraction was 0.51`, *q1 would be classified as d__Bacteria; p__Proteobacteria.*

Acknowledgements

This work was supported in part by the 2014 Kinsella Memorial Award (NAB), NIH P50 GM068763 (PJT and CFM), NSF IGERT grant number 1144807 (DM), and the NIH and the Howard Hughes Medical Institute (RK).

Author Contributions

NAB, JRR, RK, and JGC conceived and designed the experiments, JRR, NAB, JP, EK, ZE, and JGC designed and wrote software and analysis pipelines, and NAB performed the experiments and data analysis. DM provided reference database support. CFM and PJT provided mock community B7-B8 sequencing data, and BW and RJD provided mock community F1-F2 sequencing data. NAB, JRR, RK, and JGC wrote the manuscript.

Competing Interests

The authors declare that they have no competing interests.

References

1. C. Human Microbiome Project, A framework for human microbiome research. *Nature* **486**, 215-221 (2012).
2. J. Gilbert , J. Jansson, R. Knight, The Earth Microbiome project: successes and aspirations. *BMC Biology* **12**, 69 (2014).
3. Z. Liu, T. Z. DeSantis, G. L. Andersen, R. Knight, Accurate taxonomy assignments from 16S rRNA sequences produced by highly parallel pyrosequencers. *Nucleic Acids Res* **36**, e120 (2008).
4. Q. Wang, G. M. Garrity, J. M. Tiedje, J. R. Cole, Naive Bayesian classifier for rapid assignment of rRNA sequences into the new bacterial taxonomy. *Applied and Environmental Microbiology* **73**, (2007).
5. S. F. Altschul, W. Gish, W. Miller, E. W. Myers, D. J. Lipman, Basic local alignment search tool. *J. Mol. Biol.* **215**, 403-410 (1990).
6. J. G. Caporaso, J. Kuczynski, J. Stombaugh, K. Bittinger, F. D. Bushman, E. K. Costello, N. Fierer, A. Gonzalez Pena, J. K. Goodrich, J. I. Gordon, G. A. Huttley, S. T. Kelley, D. Knights, J. E. Koenig, R. E. Ley, C. A. Lozupone, D. McDonald, B. D. Muegge, M. Pirrung, J. Reeder, J. R. Sevinsky, P. J. Turnbaugh, W. A. Walters, J. Widmann, T. Yatsunenko, J. Zaneveld, R. Knight, Qiime allows analysis of high-throughput community sequence data. *Nature Methods* **7**, 335-336 (2010).
7. R. C. Edgar, Search and clustering orders of magnitude faster than BLAST. *Bioinformatics* **26**, 2460-2461 (2010).
8. E. Kopylova, L. Noe, H. Touzet, SortMeRNA: fast and accurate filtering of ribosomal RNAs in metatranscriptomic data. *Bioinformatics* **28**, 3211-3217 (2012).
9. B. Ragan-Kelley, W. A. Walters, D. McDonald, J. Riley, B. E. Granger, A. Gonzalez, R. Knight, F. Perez, J. G. Caporaso, Collaborative cloud-enabled tools allow rapid, reproducible biological insights. *ISME J* **7**, 461-464 (2013).
10. C. T. Brown, A. Howe, Q. Zhang, A. B. Pyrkosz, T. Brom, A Reference-Free Algorithm for Computational Normalization of Shotgun Sequencing Data. *arXiv*, arXiv:1203.4802 (2012).
11. P. Nowakowski, E. Ciepiela, D. Harężlak, J. Kocot, M. Kasztelnik, T. Bartyński, J. Meizner, G. Dyk, M. Malawski, The collage authoring environment. *Procedia Computer Science* **4**, 608-617 (2011).
12. G. Wilson, D. A. Aruliah, C. T. Brown, N. P. Chue Hong, M. Davis, R. T. Guy, S. H. D. Haddock, K. D. Huff, I. M. Mitchell, M. D. Plumbley, B. Waugh, E. P. White, P. Wilson, Best practices for scientific computing. *PLoS Biology* **12**, e1001745 (2014).
13. F. Perez, B. E. Granger, IPython: A System for Interactive Scientific Computing. *Computing in Science and Engineering* **9**, 21-29 (2007).
14. N. A. Bokulich, S. Subramanian, J. J. Faith, D. Gevers, J. I. Gordon, R. Knight, D. A. Mills, J. G. Caporaso, Quality-filtering vastly improves diversity estimates from Illumina amplicon sequencing. *Nature Methods* **10**, 57-59 (2013).
15. K. Pearson, Notes on regression and inheritance in the case of two parents. *Proceedings of the Royal Society of London* **58**, 240-242 (1895).

16. C. Spearman, The proof and measurement of association between two things. *Amer. J. Psychol.* **15**, 72-101 (1904).
17. D. McDonald, M. N. Price, J. Goodrich, E. P. Nawrocki, T. Z. DeSantis, A. Probst, G. L. Andersen, R. Knight, P. Hugenholtz, An improved Greengenes taxonomy with explicit ranks for ecological and evolutionary analyses of bacteria and archaea. *ISME J* **6**, 610-618 (2012).
18. U. Kõljalg, R. H. Nilsson, K. Abarenkov, L. Tedersoo, A. F. S. Taylor, M. Bahram, S. T. Bates, T. D. Bruns, J. Bengtsson-Palme, T. M. Callaghan, B. Douglas, T. Drenkhan, U. Eberhardt, M. Dueñas, T. Grebenc, G. W. Griffith, M. Hartmann, P. M. Kirk, P. Kohout, E. Larsson, B. D. Lindahl, R. Lücking, M. P. Martín, P. B. Matheny, N. H. Nguyen, T. Niskanen, J. Oja, K. G. Peay, U. Peintner, M. Peterson, K. Põldmaa, L. Saag, I. Saar, A. Schüßler, J. A. Scott, C. Senés, M. E. Smith, A. Suija, D. L. Taylor, M. T. Telleria, M. Weiß, L. K.-H., Towards a unified paradigm for sequence-based identification of Fungi. *Molecular Ecology* **22**, 5271–5277 (2013).
19. M. J. Claesson, O. O'Sullivan, Q. Wang, J. Nikkila, J. R. Marchesi, H. Smidt, W. M. de Vos, R. P. Ross, P. W. O'Toole, Comparative Analysis of Pyrosequencing and a Phylogenetic Microarray for Exploring Microbial Community Structures in the Human Distal Intestine. *PLoS One* **4**, e6669 (2009).
20. J. R. Rideout, Y. He, J. A. Navas-Molina, W. A. Walters, L. K. Ursell, S. M. Gibbons, J. Chase, D. McDonald, A. Gonzalez, A. Robbins-Pianka, J. C. Clemente, J. A. Gilbert, S. M. Huse, H. W. Zhou, R. Knight, J. G. Caporaso, Subsampled open-reference clustering creates consistent, comprehensive OTU definitions and scales to billions of sequences. *PeerJ* **2**, e545 (2014).
21. G. D. Wu, J. D. Lewis, C. Hoffmann, Y. Y. Chen, R. Knight, K. Bittinger, J. Hwang, J. Chen, R. Berkowsky, L. Nessel, H. Li, F. D. Bushman, Sampling and pyrosequencing methods for characterizing bacterial communities in the human gut using 16S sequence tags. *BMC Microbiology* **10**, 206 (2010).
22. M. T. Suzuki, S. J. Giovannoni, Bias caused by template annealing in the amplification of mixtures of 16S rRNA genes by PCR. *Applied and Environmental Microbiology* **62**, 625-630 (1996).
23. S. J. Salter, M. J. Cox, E. M. Turek, S. T. Calus, W. O. Cookson, M. F. Moffatt, P. Turner, J. Parkhill, N. J. Loman, A. W. Walker, Reagent and laboratory contamination can critically impact sequence-based microbiome analyses. *BMC Biology* **12**, 87 (2014).
24. J. G. Caporaso, C. L. Lauber, W. A. Walters, D. Berg-Lyons, C. A. Lozupone, P. J. Turnbaugh, N. Fierer, R. Knight, Global patterns of 16S rRNA diversity at a depth of millions of sequences per sample. *Proceedings of the National Academy of Sciences of the United States of America* **108**, 4516-4522 (2011).
25. M. Hamady, J. J. Walker, J. K. Harris, N. J. Gold, R. Knight, Error-correcting barcoded primers for pyrosequencing hundreds of samples in multiplex. *Nature Methods* **5**, 235-237 (2008).
26. J. G. Caporaso, C. L. Lauber, W. A. Walters, D. Berg-Lyons, J. Huntley, N. Fierer, S. M. Owens, J. Bentley, L. Fraser, M. Bauer, N. Gormley, J. A. Gilbert , G. Smith, R. Knight, Ultra-high-throughput microbial community analysis on the Illumina HiSeq and MiSeq platforms. *ISME J* **6**, 1621-1624 (2012).

27. N. A. Bokulich, D. A. Mills, Improved Selection of internal transcribed spacer-specific primers enables quantitative, ultra-high-throughput profiling of fungal communities. *Applied and Environmental Microbiology* **79**, 2519-2526 (2013).
28. D. McDonald, J. C. Clemente, J. Kuczynski, J. R. Rideout, J. Stombaugh, D. Wendel, A. Wilke, S. Huse, J. Hufnagle, F. Meyer, R. Knight, J. G. Caporaso, The biological observation matrix (BIOM) format or: how I learned to stop worrying and love the ome-ome. *Gigascience* **1**, 7 (2012).

Tables/Figures

Table 1. Comparisons of Optimized Method Performance For Species Assignment of Simulated Community Datasets, Ordered Best to Worst

	Partial Reference					Full Reference				
Dataset[a]	**Method**	**Configuration**[b]	**P**[c]	**R**	**F**	**Method**	**Configuration**	**P**	**R**	**F**
B1	**UCLUST**	0.51:0.8:1	0.85	0.82	0.83	**UCLUST**	0.51:0.8:1	0.99	0.99	0.99
	SortMeRNA	0.51:0.8:1:0.8:1.0	0.85	0.81	0.83	**BLAST**	0.001	0.99	0.99	0.99
	BLAST	1.00E-09	0.85	0.81	0.83	**SortMeRNA**	0.51:0.8:1:0.8:1.0	0.99	0.98	0.98
	RDP/mothur	0.4	0.81	0.84	0.82	**RDP/mothur**	0.5	0.92	0.97	0.95
B2	**SortMeRNA**	0.51:0.8:1:0.8:1.0	0.83	0.77	0.8	**SortMeRNA**	0.51:0.8:1:0.8:1.0	0.96	0.93	0.94
	UCLUST	0.51:0.8:1	0.81	0.77	0.79	**UCLUST**	0.51:0.8:1	0.96	0.93	0.94
	BLAST	1.00E-09	0.83	0.76	0.79	**BLAST**	0.001	0.96	0.92	0.94
	RDP/mothur	0.4	0.85	0.72	0.78	**RDP/mothur**	0.5	0.93	0.87	0.90
F1	**RDP/mothur**	0.4	0.34	0.23	0.28	**UCLUST**	0.51:0.8:1	0.98	0.93	0.96
	SortMeRNA	0.51:0.8:1:0.8:1.0	0.28	0.23	0.25	**SortMeRNA**	0.51:0.8:1:0.8:1.0	0.98	0.93	0.96
	BLAST	1.00E-09	0.27	0.23	0.25	**BLAST**	0.001	0.98	0.93	0.95
	UCLUST	0.51:0.8:1	0.27	0.21	0.24	**RDP/mothur**	0.5	0.96	0.90	0.93
F2	**RDP/mothur**	0.4	0.36	0.19	0.25	**SortMeRNA**	0.51:0.8:1:0.8:1.0	0.95	0.90	0.93
	BLAST	1.00E-09	0.27	0.2	0.23	**BLAST**	0.001	0.95	0.90	0.93
	SortMeRNA	0.51:0.8:1:0.8:1.0	0.25	0.21	0.23	**UCLUST**	0.51:0.8:1	0.95	0.90	0.93
	UCLUST	0.51:0.8:1	0.26	0.2	0.23	**RDP/mothur**	0.5	0.95	0.84	0.89

[a]Datasets B1 and F1 represent simulated communities comprising full-length reference sequences; B2 and F2 represent 100-nt simulated reads.

[b]The optimal parameter configuration tested in the between-method comparisons. See Table 2 for details.

[c]P = precision; R = recall; F = F-measure.

PeerJ PrePrints | https://dx.doi.org/10.7287/peerj.preprints.934v2 | CC-BY 4.0 Open Access | rec: 10 Jul 2015, publ: 10 Jul 2015

Table 2. Within-Method Parameter Optimization Across Simulated Community Datasets Using Partial Reference Database, Ordered By F-Measure

Method	Configuration[a]	Precision	Recall	F-measure
SortMeRNA	0.51:0.8:1:0.8:1.0	0	20	20
	0.76:0.8:1:0.8:1.0	0	20	20
	1.0:0.9:1:0.9:1.0	0	20	20
	1.0:0.9:1:0.8:1.0	0	20	20
	1.0:0.8:1:0.9:1.0	0	20	20
	1.0:0.8:1:0.8:1.0	0	20	20
	0.76:0.9:1:0.9:1.0	0	20	20
	0.51:0.8:1:0.9:1.0	0	20	20
	0.76:0.8:1:0.9:1.0	0	20	20
	0.76:0.9:1:0.8:1.0	0	20	20
	0.51:0.9:1:0.9:1.0	0	20	20
	0.51:0.9:1:0.8:1.0	0	20	20
	0.51:0.9:3:0.9:1.0	13	8	19
	0.51:0.9:3:0.8:1.0	13	8	19
	0.51:0.8:3:0.8:1.0	10	5	14
RDP	0.4	0	20	20
	0.5	0	20	20
	0.6	5	11	20
	0.7	5	3	20
	0.3	0	20	19
	0.2	0	20	11
	0.1	0	20	10
	0	0	20	9
mothur	0.4	0	20	20
	0.5	0	20	20
	0.6	5	10	20
	0.3	0	20	19
	0.7	5	3	19
	0.2	0	20	11
	0	0	20	10
	0.1	0	20	10
UCLUST	0.51:0.8:1	0	20	20
	0.76:0.8:1	0	20	20
	1.0:0.9:1	0	20	20
	1.0:0.8:1	0	20	20
	0.76:0.9:1	0	20	20
	0.51:0.9:1	0	20	20
	0.51:0.9:3	7	10	19
	0.51:0.8:3	10	10	18

	0.51:0.9:5	20	0	12
	0.51:0.8:5	20	0	10
	0.76:0.8:3	20	0	0
	0.76:0.8:5	20	0	0
	0.76:0.9:3	18	0	0
BLAST	1E-9	20	12	19
	0.001	6	20	7
	1E+4	6	20	7

[a]Parameter configurations used for classification. RDP/mothur = confidence threshold; BLAST = e-value threshold; SortMeRNA = minimum consensus fraction (f):similarity (s):best N alignments (b):coverage:e value; UCLUST = minimum consensus fraction (f):similarity (s):and maximum accepts (a). E.g., "SortMeRNA 0.51:0.8:1:0.8:1.0" indicates 0.51 minimum consensus fraction, 0.8 similarity, 1 best alignment, 0.8 coverage, and e-value threshold = 1.0.

Figure 1. Evaluation framework for new taxonomic assignment methods. We provide test data and an evaluation framework, facilitating the benchmarking of future methods for short read taxonomy assignment in the context of the results presented here. All mock-community and natural-community test data are provided in our data store (hosted on GitHub). The developer of a new method can assign taxonomy to these test data and generate BIOM(*28*) files (green). Those BIOM files can then be passed to the evaluation framework, where they will be compared to pre-computed BIOM files from the data store (red and blue) based on three evaluations of accuracy of the taxonomic assignments. If the new method does not outperform the pre-computed results, it should be abandoned or optimized before an attempt is made to apply or publish it. If it does out-perform the pre-computed results, it indicates that the developer should pursue publication of the method.

Finally, the developer can submit their best BIOM tables to the data store using the GitHub Pull Request mechanism, so a comparison against their methods will be included in future evaluations by other method developers.

Figure 2. Taxonomy classifier selection critically shapes assignment accuracy of mock communities. Violin plots illustrate the distribution of precision, recall, F-measure, Pearson *r*, and Spearman *rho* values across all mock communities and all parameter configurations for a given method for family-level (left), genus-level (middle), or species-level taxonomy assignments (right). Heavy dashed lines indicate median values, fine dashed lines indicate quartiles.

Figure 3. Taxonomy classifier configuration and mock community composition alter assignment accuracy at genus-level. Heatmaps indicate the precision, recall, F-measure, Pearson *r*, and Spearman *rho* values for taxonomy classification of each mock community (columns) by each method configuration (rows). The shade of the intersecting box indicates the score for a given evaluation metric, as indicated in the color keys on the right. Bacterial mock communities B1-B8 are on the left side of each panel; Fungal mock communities F1-F2 appear on the right side of each panel. Parameters: e = BLAST e-value; c = confidence; f = minimum consensus fraction; d = similarity; b = best N alignments; a = maximum accepts. Not all SortMeRNA parameters/configurations are marked, as different coverage and e values did not measurably influence any scores. See the pre-computed repository results for fully annotated plots: http://bit.ly/srta-012.

Figure 4. Taxonomy classifier configuration alters assignment accuracy of simulated communities. Simulated reads were generated by randomly selecting 10% of the sequences from the reference sequence databases (Greengenes for bacteria, UNITE for fungi); taxonomy classifications were then made using either the full reference database (full reference) or the remaining 90% of the sequences as the reference database (partial reference). Heatmaps indicate the precision (P), recall (R), and F-measure (F) values for taxonomy classification of each simulated community (columns) by each method configuration (rows) at species levels. The shade of the intersecting box indicates the score for a given evaluation metric, as indicated in the color key on the right. Bacterial simulated communities B1-B2 are on the left side of each panel; Fungal simulated communities F1-F2 appear on the right side of each panel. B1 and F1 represent classifications of full-length sequences, B2 and F2 represent classifications of simulated 100 nt sequencing reads. Four iterations for each classifier are shown for each simulated community. Parameters: e = BLAST e-value; c = confidence; f = minimum consensus fraction; d = similarity; b = best N alignments; a = maximum accepts. Not all SortMeRNA parameters/configurations are marked, as different coverage and e values did not measurably influence any scores. See the pre-computed repository results for fully annotated plots: http://bit.ly/srta-012.

Figure 5. Classifier choice influences computational runtime. A) Computational runtime for each classifier was tested as a function of query sequence count (up to 998300 total sequences of 302 ± 88 nt (mean ± SD)). Subsets of query sequences were classified against

a single reference sequence database. As total runtime includes reference database indexing time (*y*-intercept), the slope of these curves is the best indicator of query sequence classification time. Note, full BLAST results are not shown due to the steep slope relative to other methods. B) Computational runtime as a function of reference sequence count (89339 total sequences of 302 ± 88 nt (mean ± SD)). A single query sequence was classified against increasingly large subsets of the reference sequence database to determine the how reference database size influences database indexing time for each classifier. Results also available at http://nbviewer.ipython.org/github/gregcaporaso/short-read-tax-assignment/blob/0.1.2/ipynb/runtime/base.ipynb.

Supplementary Material

Supplementary Figure 1. Mock community datasets analyzed in this study.

Supplementary Figure 2. Mock community A composition.

Supplementary Figure 3. Mock community B composition.

Supplementary Figure 4. Mock community C composition.

Supplementary Figure 5. Mock community D composition.

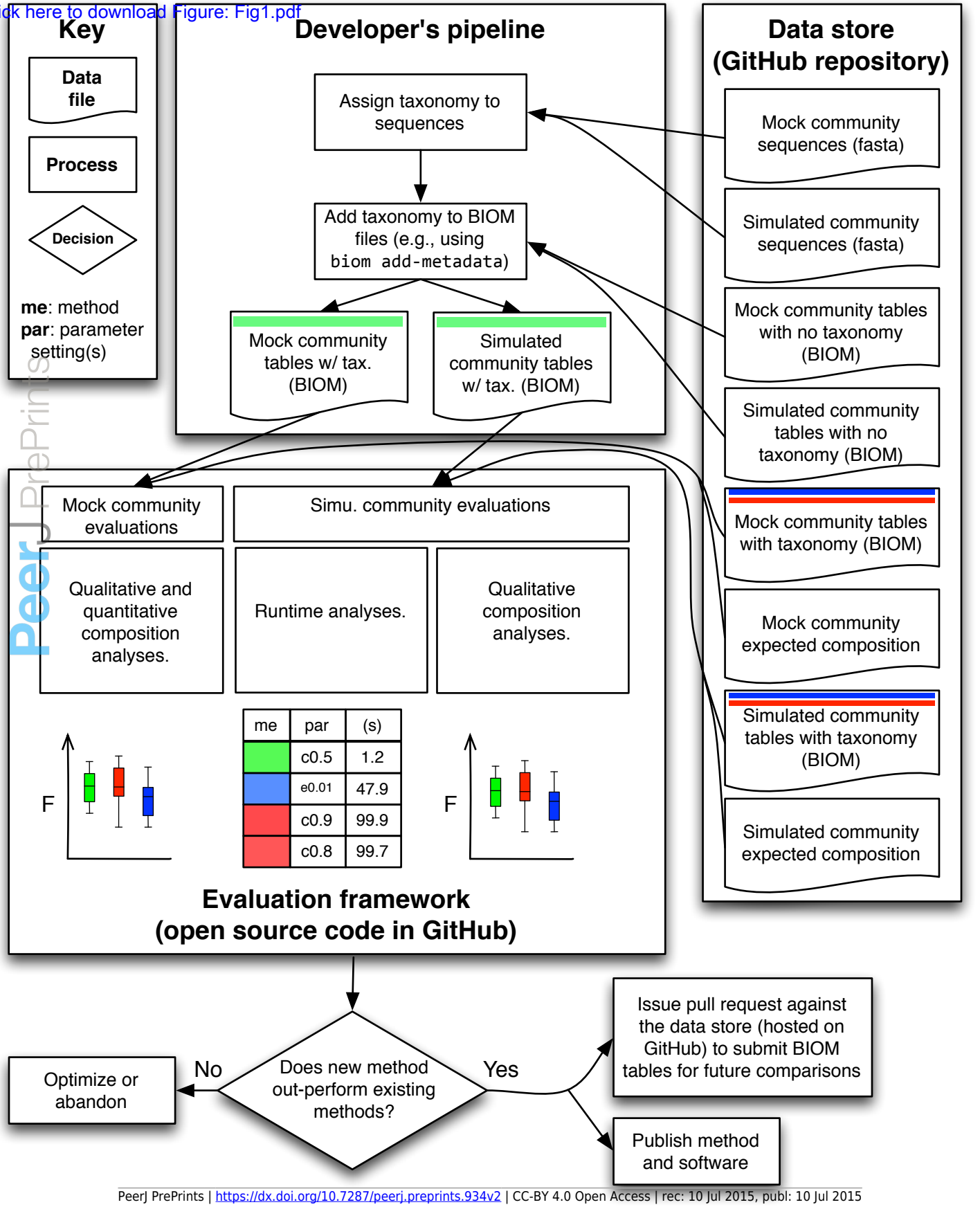
Supplementary Figure 6. Taxonomy classifier configuration and mock community composition alter assignment accuracy at family-level.

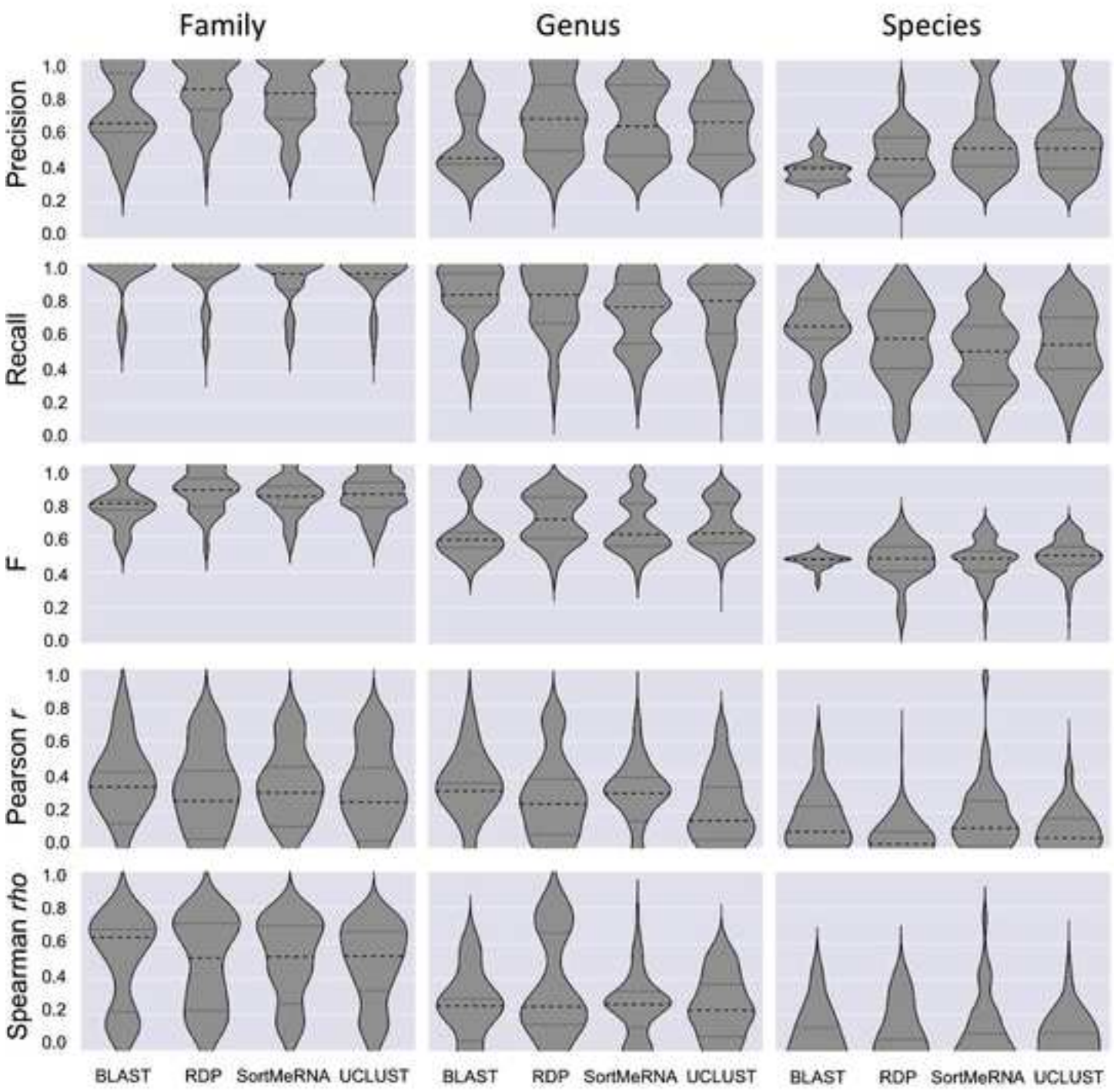
Supplementary Figure 7. Taxonomy classifier configuration and mock community composition alter assignment accuracy at species-level.

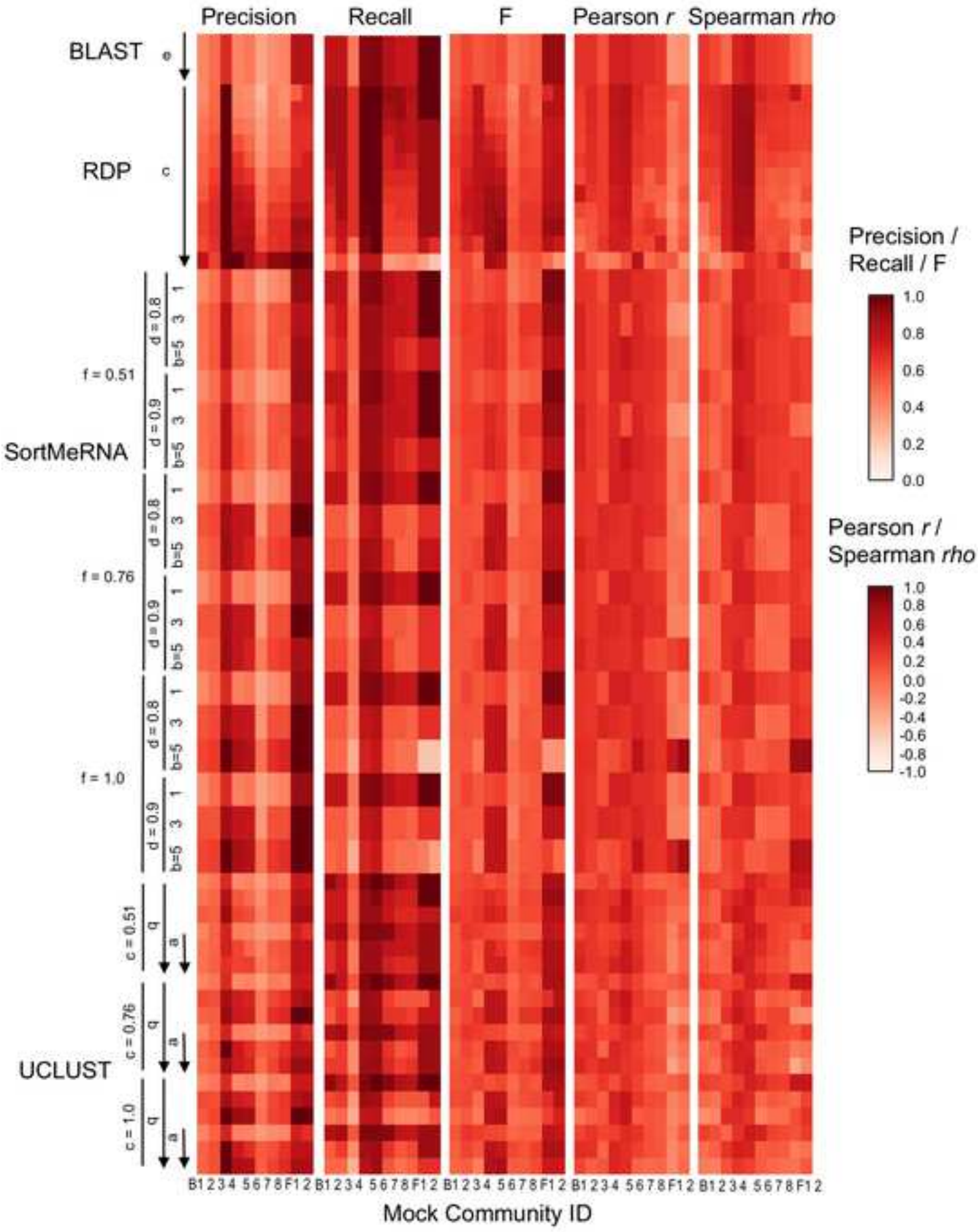
Supplementary Figure 8. Taxonomy classifier selection critically shapes assignment accuracy of simulated communities. Violin plots illustrate the distribution of precision, recall, and F-measure values across all simulated communities and all parameter configurations for a given method for family-level (left), genus-level (middle), or species-level taxonomy assignments (right). Heavy dashed lines indicate median values, fine dashed lines indicate quartiles.

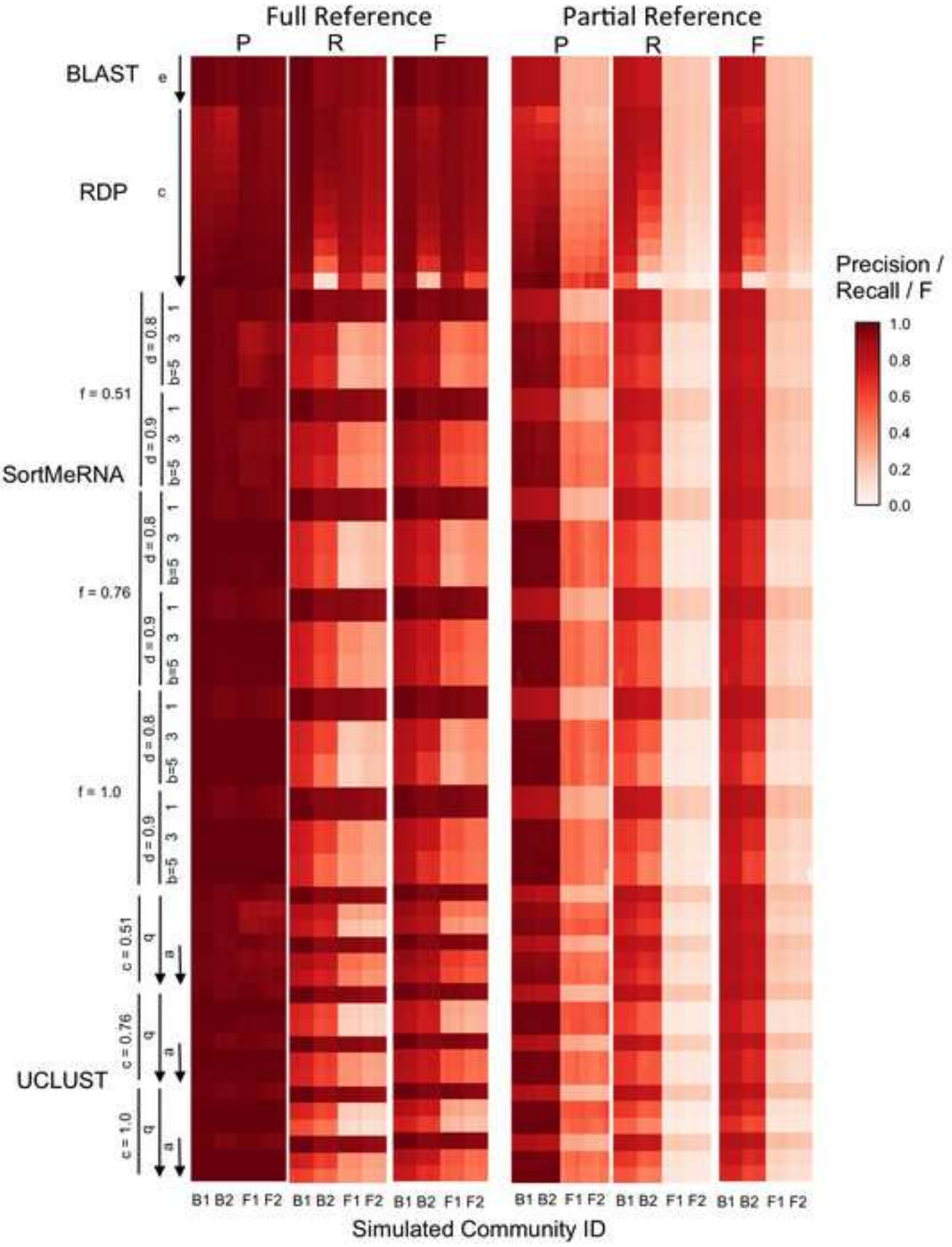
Supplementary Figure 9. Taxonomic lineages represented in reference databases

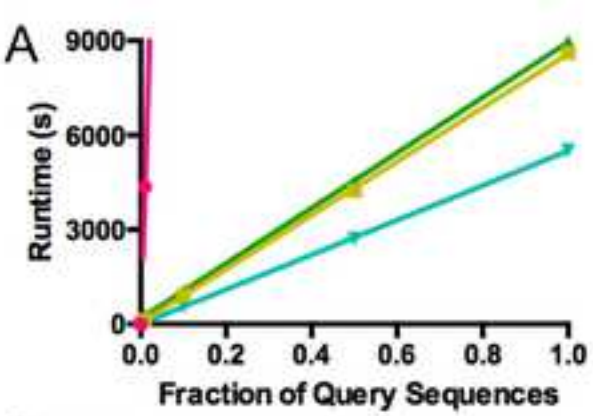
Supplementary Figure 10. Evaluation of mothur taxonomy classifier. A, Distribution of F-measure scores across all partial-reference simulated communities and all parameter configurations for each method for species-level taxonomy assignments (right). Heavy dashed lines indicate median values, fine dashed lines indicate quartiles. SM = SortMeRNA. B, Confidence configuration and simulated community composition alter assignment accuracy at species-level. See figure 4 for full description of analysis and comparison to other classifiers and configurations.



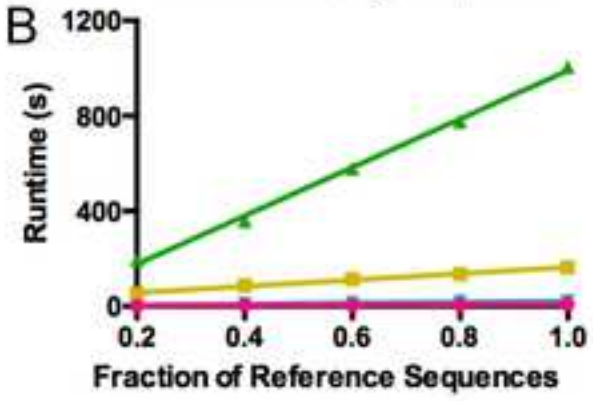








	Slope	Intercept
BLAST	430214 ± 416.3	-49.46 ± 106.2
RDP	8538 ± 60.74	43.03 ± 30.49
SortMeRNA	8776 ± 32.32	173.8 ± 16.23
UCLUST	5501 ± 30.49	0.4976 ± 15.31



	Slope	Intercept
BLAST	7.1 ± 0.45	1.9 ± 0.30
RDP	134 ± 7.0	30 ± 4.6
SortMeRNA	1017 ± 31	-27 ± 21
UCLUST	20 ± 0.93	1.6 ± 0.62