OptiFit: an improved method for fitting amplicon sequences to existing OTUs

2021-09-07

Kelly L. Sovacool[1], Sarah L. Westcott[2], M. Brodie Mumphrey[1], Gabrielle A. Dotson[1], Patrick D. Schloss[2]†

1 Department of Computational Medicine and Bioinformatics, University of Michigan

2 Department of Microbiology and Immunology, University of Michigan

† To whom correspondence should be addressed: pschloss@umich.edu

Abstract

Assigning amplicon sequences to operational taxonomic units (OTUs) is often an important step in characterizing the composition of microbial communities across large datasets. OptiClust, a *de novo* OTU clustering method, has been shown to produce higher quality OTU assignments than other methods and at comparable or faster speeds. A notable difference between *de novo* clustering and database-dependent reference clustering methods is that OTU assignments from *de novo* methods may change when new sequences are added to a dataset. However, in some cases one may wish to incorporate new samples into a previously clustered dataset without performing clustering again on all sequences, such as when comparing across datasets or deploying machine learning models where OTUs are features. Existing reference-based clustering methods produce consistent OTUs, but they only consider the similarity of each query sequence to a single reference sequence in an OTU, thus resulting in some dissimilar sequences being clustered to the same OTU. To provide an efficient and robust method to fit amplicon sequence data to existing OTUs, we developed the OptiFit algorithm. Like OptiFit, OptiClust considers the similarity of all pairs of reference and query sequences in an OTU to produce OTUs of the best possible quality. We tested OptiFit using four microbiome datasets with two different strategies: by clustering to an external reference database or by splitting the dataset into a reference and query set and clustering the query sequences to the reference set after clustering it using OptiClust. The result is an improved implementation of closed and open-reference clustering. OptiFit produces OTUs of similar quality as OptiClust and at faster speeds when using the split dataset strategy, although the OTU quality and processing speed depends on the database chosen when using the external database strategy. OptiFit provides a suitable option for users who require consistent OTU assignments at the same quality afforded by *de novo* clustering methods.

Importance

Advancements in DNA sequencing technology have allowed researchers to affordably generate millions of reads from microorganisms in diverse natural communities. Efficient and robust software tools are needed to assign microbial sequences into taxonomic groups for characterization and comparison of communities. The OptiClust algorithm produces high quality groups by comparing sequences to each other, but the assignments can change when new sequences are added to a dataset, making it difficult to compare different studies. Other approaches assign sequences to groups by comparing them to sequences in a reference database to produce consistent assignments, but the quality of the groups produced is reduced compared to OptiClust. We developed OptiFit, a new reference-based algorithm that produces consistent yet high quality assignments like OptiClust. OptiFit allows researchers to compare microbial communities across different studies or add new data to existing studies without sacrificing the quality of the group assignments.

Introduction

Amplicon sequencing is a mainstay of microbial ecology. Researchers can affordably generate millions of sequences to characterize the composition of hundreds of samples from microbial communities without the need for culturing. In many analysis pipelines, 16S rRNA gene sequences are assigned to operational taxonomic units (OTUs) to facilitate comparison of taxonomic composition between communities to avoid the need for taxonomic classification. A distance threshold of 3% (or sequence similarity of 97%) is commonly used to cluster sequences into OTUs based on pairwise comparisons of the sequences within the dataset. The method chosen for clustering affects the quality of OTU assignments and thus may impact downstream analyses of community composition (1–3).

There are two main categories of OTU clustering algorithms: *de novo* and reference-based. OptiClust is a *de novo* clustering algorithm which uses the distance score between all pairs of sequences in the dataset to cluster them into OTUs by maximizing the Matthews Correlation Coefficient (MCC) (1). This approach takes into account the distances between all pairs of sequences when assigning query sequences to OTUs, in contrast to other *de novo* methods such as the greedy clustering algorithms implemented in USEARCH and VSEARCH, which only consider the distance between the query sequence and a representative centroid sequence in the OTU (4, 5). In methods employing greedy clustering algorithms, only the distance between each sequence and the centroid sequence is considered while clustering. As a result, pairs of sequences in the same OTU may have a greater distance than the specified threshold, i.e. they are false positives. In contrast, the OptiClust algorithm takes into account the distance between all pairs of sequences when considering how to cluster sequences into OTUs and is thus less willing to take on false positives. A limitation of *de novo* clustering is that different OTU assignments will be produced when new sequences are added to a dataset, making it difficult to use *de novo* clustering to compare OTUs between different studies. Furthermore, since *de novo* clustering requires calculating and comparing distances between all sequences in a dataset, the execution time can be slow for very large datasets. Reference clustering attempts to overcome the limitations of *de novo* clustering methods by using a representative set of sequences from a database, with each reference sequence seeding an OTU. Commonly, the Greengenes set of representative full length sequences clustered at 97% similarity is used as the reference with VSEARCH (5–7). Query sequences are then clustered into OTUs based on their similarity to the reference sequences. Any query sequences that are not within the distance threshold to any of the reference sequences are either thrown out (closed reference clustering) or clustered *de novo* to create additional OTUs (open reference clustering). While reference-based clustering is generally fast, it is limited by the diversity of the reference database. Rare or novel sequences in the sample will be lost in closed reference mode if they are not represented by a similar sequence in the database. Just like in *de novo* clustering, reference-based methods that use greedy clustering algorithms only consider the distance between each query and a single reference sequence, causing some sequences in the same OTU to be more dissimilar to each other than the specified threshold. Previous studies found that the OptiClust *de novo* clustering algorithm created the highest quality OTU assignments of all clustering methods (1).

To overcome the limitations of current reference-based and *de novo* clustering algorithms while maintaining OTU quality, we developed OptiFit, a reference-based clustering algorithm. While other tools represent reference OTUs with a single sequence, OptiFit uses multiple sequences in existing OTUs as the reference and fits new sequences those reference OTUs. In contrast to other tools, OptiFit considers all pairwise distance scores between reference and query sequences when assigning sequences to OTUs in order to produce OTUs of the highest possible quality. Here, we tested the OptiFit algorithm with the reference as a public database (e.g. Greengenes) or *de novo* OTUs and compared the performance to existing tools. To evaluate the OptiFit algorithm and compare to existing methods, we used four published datasets isolated from soil (8), marine (9), mouse gut

(10), and human gut (11) samples. OptiFit is available within the mothur software program.

Results

The OptiFit algorithm

OptiFit leverages the method employed by OptiClust of iteratively assigning sequences to OTUs to produce the highest quality OTUs possible, and extends this method for reference-based clustering. OptiClust first seeds each sequence into its own OTU as a singleton. Then for each sequence, OptiClust considers whether the sequence should move to a different OTU or remain in its current OTU, choosing the option that results in a better Matthews correlation coefficient (MCC) (1). The MCC uses all values from a confusion matrix and ranges from zero to one, with a score of one occurring when all sequence pairs are true positives and true negatives, and a score of zero when all pairs are false positives and false negatives. Sequence pairs that are similar to each other (i.e. within the distance threshold) are counted as true positives if they are clustered into the same OTU, and false negatives if they are not in the the same OTU. Sequence pairs that are not similar to each other are true negatives if they are not clustered into the same OTU, and false positives if they are not in the same OTU. OptiClust iterations continue until the MCC stabilizes or until a maximum number of iterations is reached. This process produces *de novo* OTU assignments with the most optimal MCC given the input sequences.

OptiFit begins where OptiClust ends, starting with a list of reference OTUs and their sequences, a list of query sequences to cluster to the reference OTUs, and the sequence pairs that are within the distance threshold (e.g. 0.03) (Figure 1). Initially, all query sequences are placed into separate OTUs. Then, the algorithm iteratively reassigns the query sequences to the reference OTUs to optimize the MCC. Alternatively, a sequence will remain unassigned if the MCC value is maximized when the sequence is a singleton rather than clustered into a reference OTU. All query and reference sequence pairs are

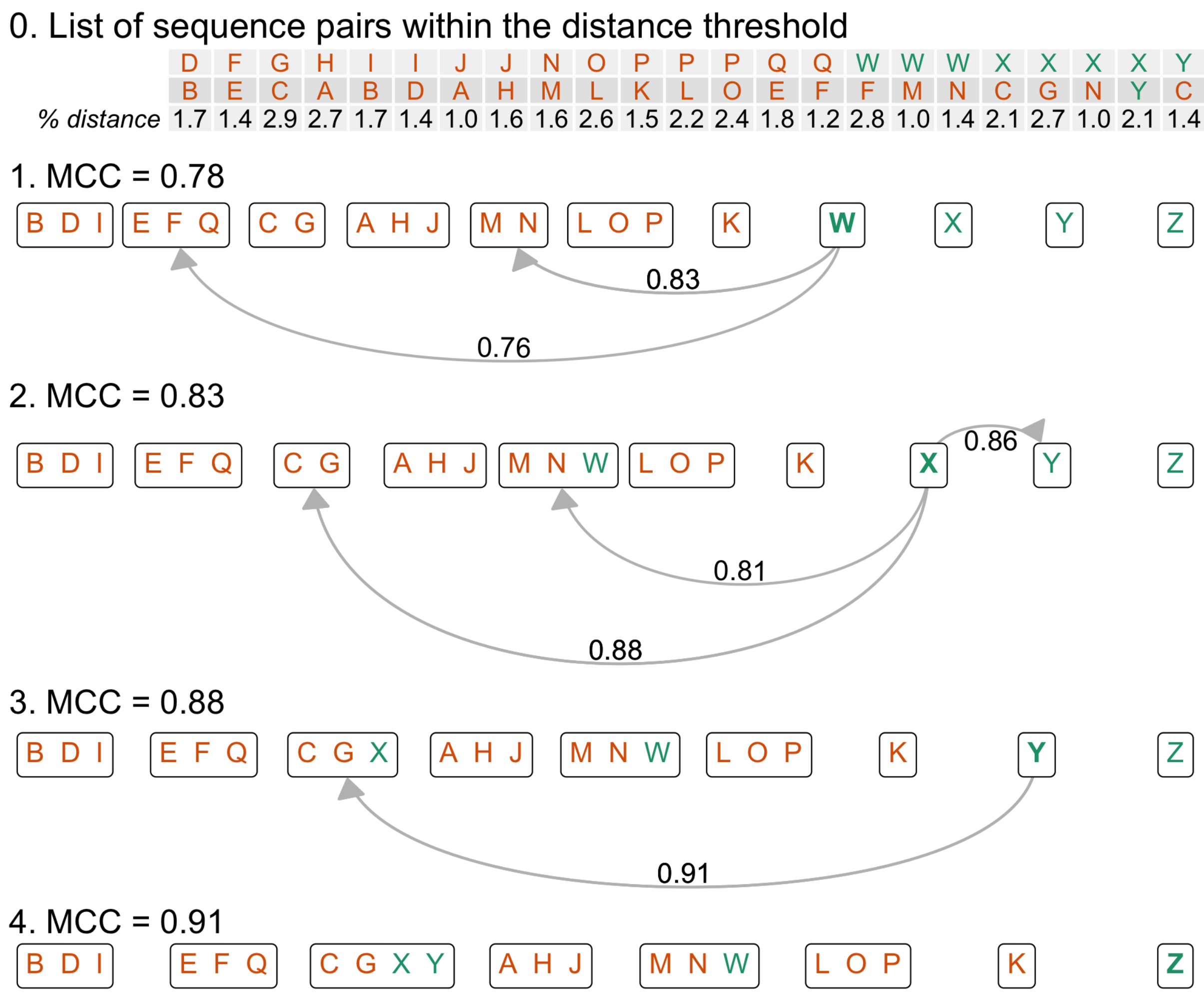

Figure 1: The OptiFit Algorithm. Here we present a toy example of the OptiFit algorithm fitting query sequences to existing OTUs, given the list of all sequence pairs that are within the distance threshold (here 3% is used). The goal of OptiFit is to assign the query sequences W through Z (colored green) to the reference OTUs created by clustering Sequences A through Q (colored orange) which were previously clustered *de novo* with OptiClust (see the OptiClust supplemental text (1)). Initially, OptiFit places each query sequence in its own OTU. Then, for each query sequence (**bolded**), OptiFit determines what the new MCC score would be if that sequence were moved to one of the OTUs containing at least one other similar sequence. The sequence is then moved to the OTU which would result in the best MCC score. OptiFit stops iterating over sequences once the MCC score stabilizes (in this example; only one iteration over each sequence is needed).

considered when calculating the MCC. This process is repeated until the MCC changes by no more than 0.0001 (default) or until a maximum number of iterations is reached (default: 100). In the closed reference mode, any query sequences that cannot be clustered into reference OTUs are discarded, and the results only contain OTUs that exist in the original reference. In the open reference mode, unassigned query sequences are clustered *de novo* using OptiClust to generate new OTUs. The final MCC is reported with the best OTU assignments. There are two strategies for generating OTUs with OptiFit: 1) cluster the query sequences to reference OTUs generated by *de novo* clustering an independent database, or 2) split the dataset into a reference and query fraction, cluster the reference sequences *de novo*, then cluster the query sequences to the reference OTUs.

Reference clustering with public databases

While *de novo* clustering produces the highest quality OTUs, researchers may prefer to perform reference clustering to a public database because reference-based methods produce consistent OTUs and are generally faster than *de novo* methods. In closed reference mode, sequences that cannot be clustered into reference OTUs are thrown out, so that the final clustering contains only OTUs that exist in the reference. To test how OptiFit performs for this purpose, we clustered each dataset to three databases of reference OTUs: the Greengenes database, the SILVA non-redundant database, and the Ribosomal Database Project (RDP) (6, 12, 13). Reference OTUs for each database were created by performing *de novo* clustering with OptiClust at a distance threshold of 3% using the V4 region of each sequence (see Figure 2). The *de novo* MCC scores were 0.72, 0.74, and 0.73 for Greengenes, RDP, and SILVA, respectively. clustering sequences to Greengenes and SILVA in closed reference mode performed similarly, with median MCC scores of 0.80 and 0.72 respectively, while the median MCC was 0.33 when clustering to RDP (see Figure 3). For comparison, clustering datasets with OptiClust produced an average MCC score of 0.83. This gap in OTU quality mostly disappeared when clustering in open reference mode, which produced median MCCs of 0.82 with Greengenes, 0.81 with SILVA, and 0.82 with the RDP. Thus, open reference OptiFit produced OTUs of very similar quality as *de novo* clustering, and closed reference OptiFit followed closely behind as long as a suitable reference database was chosen.

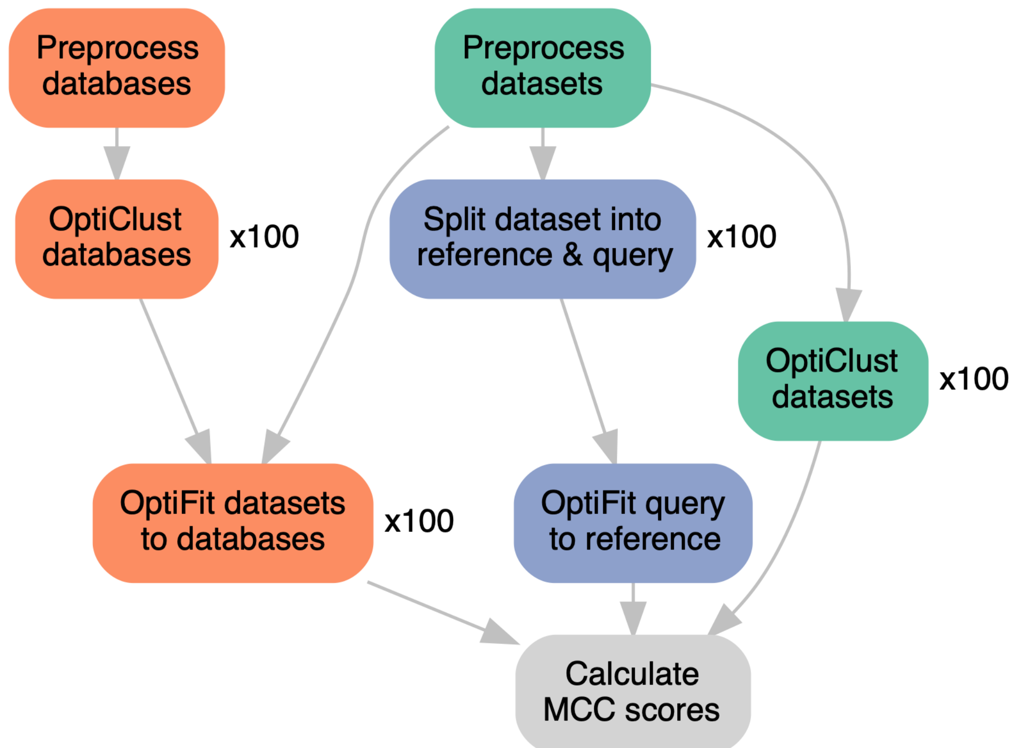

Figure 2: The Analysis Workflow. Reference sequences from Greengenes, the RDP, and SILVA were downloaded, preprocessed with mothur by trimming to the V4 region, and clustered *de novo* with OptiClust for 100 repetitions. Datasets from human, marine, mouse, and soil microbiomes were downloaded, preprocessed with mothur by aligning to the SILVA V4 reference alignment, then clustered *de novo* with OptiClust for 100 repetitions. Individual datasets were fit to reference databases with OptiFit; OptiFit was repeated 100 times for each dataset and database combination. Datasets were also randomly split into a reference and query fraction, and the query sequences were fit to the reference sequences with OptiFit for 100 repetitions. The final MCC score was reported for all OptiClust and OptiFit repetitions.

Since closed reference clustering does not cluster query sequences that could not be clustered into reference OTUs, an additional measure of clustering performance to consider is the fraction of query sequences that were able to be clustered. On average, more sequences were clustered with Greengenes as the reference (43.1%) than with SILVA

(36.4%) or with the RDP (7.1%). This mirrored the result reported above that Greengenes produced better OTUs in terms of MCC score than either SILVA or RDP. Note that *de novo* and open reference clustering methods always cluster 100% of sequences into OTUs. The database chosen affects the final OTU assignments considerably in terms of both MCC score and fraction of query sequences that could be clustered into the reference OTUs.

Despite the drawbacks, closed reference methods have been used when fast execution speed is required, such as when using very large datasets (14). To compare performance in terms of speed, we repeated each OptiFit and OptiClust run 100 times and measured the execution time. Across all dataset and database combinations, closed reference OptiFit outperformed both OptiClust and open reference OptiFit. For example, with the human dataset fit to SILVA reference OTUs, the average run times in seconds were 549.1 for closed reference OptiFit, 800.3 for *de novo* clustering the dataset, and 886.0 for open reference OptiFit. Thus, the OptiFit algorithm continues the precedent that closed reference clustering sacrifices OTU quality for execution speed.

To compare to the reference clustering methods used by QIIME2, we clustered each dataset with VSEARCH against the Greengenes database of OTUs previously clustered at 97% sequence similarity. Each reference OTU from the Greengenes 97% database contains one reference sequence, and VSEARCH maps sequences to the reference based on each individual query sequence's similarity to the single reference sequence. In contrast, OptiFit accepts reference OTUs which each may contain multiple sequences, and the sequence similarity between all query and reference sequences is considered when assigning sequences to OTUs. *De novo* clustering with OptiClust produced 56.0% higher quality OTUs than VSEARCH in terms of MCC, but performed 39.6% slower than VSEARCH. In closed reference mode, OptiFit produced 25.9% higher quality OTUs than VSEARCH, but VSEARCH was able to cluster 35.1% more query sequences than OptiFit to the Greengenes reference database. This is because VSEARCH only considers the

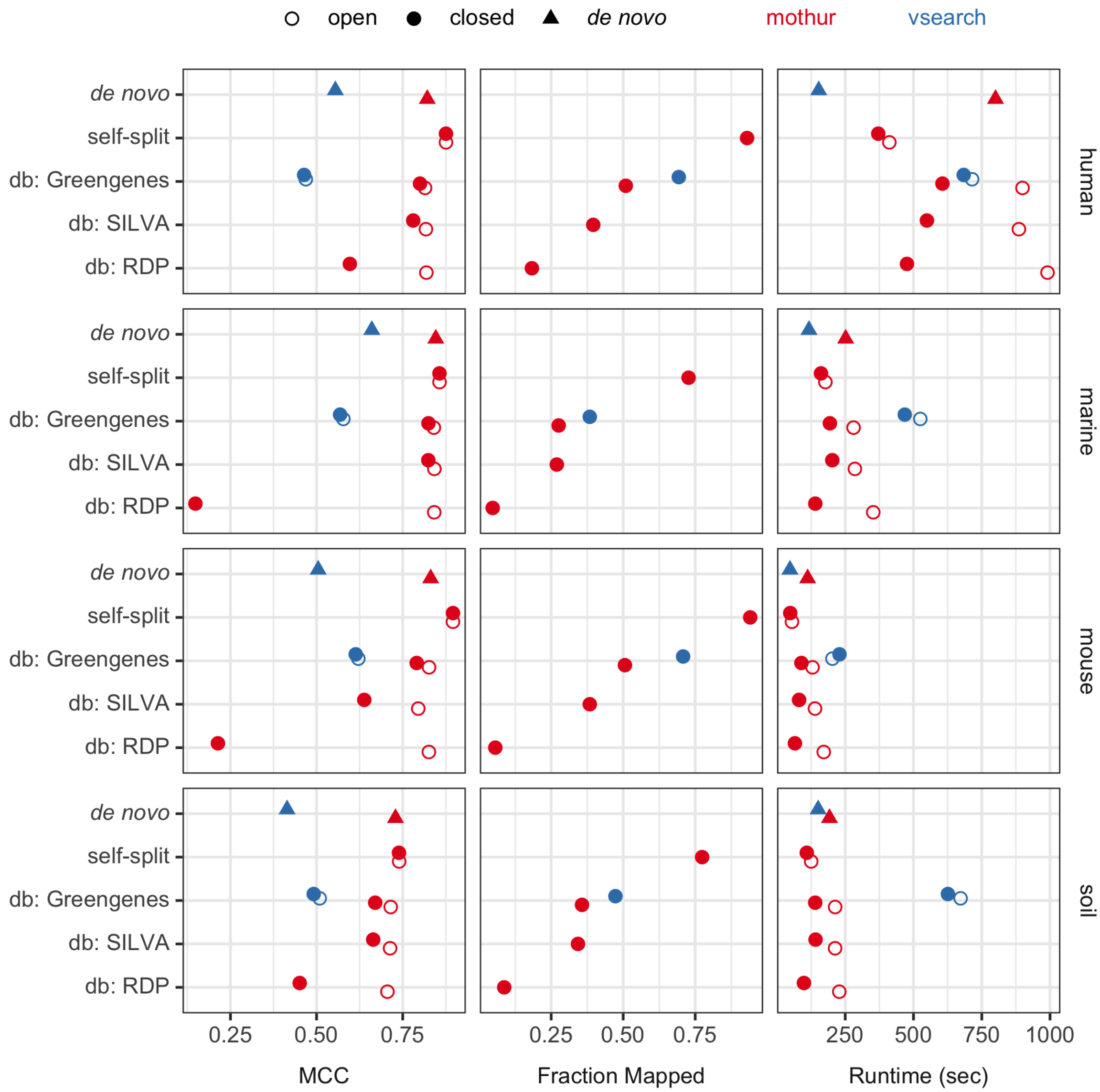

Figure 3: Benchmarking Results. The median MCC score, fraction of query sequences that mapped in closed-reference clustering, and runtime in seconds from repeating each clustering method 100 times. Each dataset underwent *de novo* clustering using OptiClust or reference-based clustering using OptiFit with one of two strategies; splitting the dataset and fitting 50% the sequences to the other 50%, or fitting the dataset to a reference database (Greengenes, SILVA, or RDP). Reference-based clustering was repeated with open and closed mode. For additional comparison, VSEARCH was used for *de novo* and reference-based clustering against the Greengenes database.

distances between each query sequence to the single reference sequence, while OptiFit considers the distances between all pairs of sequences in an OTU. When open reference clustering, OptiFit produced higher quality OTUs than VSEARCH against the Greengenes database, with median MCC scores of 0.82 and 0.54, respectively). In terms of run time, OptiFit outperformed VSEARCH in both closed and open reference mode by 74.3% and 135.3% on average respectively. Thus, the more stringent OTU definition employed by OptiFit, which prefers the query sequence to be similar to all other sequences in the OTU rather than to only one sequence, resulted in fewer sequences being clustered to reference OTUs than when using VSEARCH, but caused OptiFit to outperform VSEARCH in terms of both OTU quality and execution time.

Reference clustering with split datasets

When performing reference clustering against public databases, the database chosen greatly affects the quality of OTUs produced. OTU quality may be poor when the reference database consists of sequences that are too unrelated to the samples of interest, such as when samples contain novel populations. While *de novo* clustering overcomes the quality limitations of reference clustering to databases, OTU assignments are not consistent when new sequences are added. Researchers may wish to cluster new sequences to existing OTUs or to compare OTUs across studies. To determine how well OptiFit performs for clustering new sequences to existing OTUs, we employed a split dataset strategy, where each dataset was randomly split into a reference fraction and a query fraction. Reference sequences were clustered *de novo* with OptiClust, then query sequences were clustered to the *de novo* OTUs with OptiFit.

First, we tested whether OptiFit performed as well as *de novo* clustering when using the split dataset strategy with half of the sequences selected for the reference by a simple random sample (a 50% split). OTU quality was highly similar to that from OptiClust regardless of mode (-4.62% difference in median MCC). In closed reference mode, OptiFit was able to cluster 85.2% of query sequences to reference OTUs with the split strategy, a great improvement over the average 43.1% of sequences clustered to the Greengenes database. In terms of run time, closed and open reference OptiFit performed faster than OptiClust on whole datasets by 39.1% and 31.8 respectively. The split dataset strategy also performed 4.0% faster than the database strategy in closed reference mode and 40.5% faster in open reference mode. Thus, reference clustering with the split dataset strategy creates as high quality OTUs as *de novo* clustering yet at a faster run time, and fits far more query sequences than the database strategy.

While we initially tested this strategy using a 50% split of the data into reference and query fractions, we next investigated whether there was an optimal reference fraction size. To identify the best reference size, reference sets with 10% to 90% of the sequences were created, with the remaining sequences used for the query. OTU quality was remarkably consistent across reference fraction sizes. For example, splitting the human dataset 100 times yielded a coefficient of variation of 0.00045 for the MCC score across all fractions. Run time generally decreased as the reference fraction increased; for the human dataset, the median run time was 470.1 with 10% of sequences in the reference and 305.8 with 90% of sequences in the reference (Figure 4). In closed reference mode, the fraction of sequences that mapped increased as the reference size increased; for the human dataset, the median fraction mapped was 0.92 with 10% of sequences in the reference and 0.97 with 90% of sequences in the reference. These trends held for the other datasets as well (Figure 4). Thus, the reference fraction did not affect OTU quality in terms of MCC score, but did affect the run time and the fraction of sequences that mapped during the closed reference clustering.

After testing the split strategy using a simple random sample to select the reference sequences, we then investigated other methods of splitting the data. We tested three

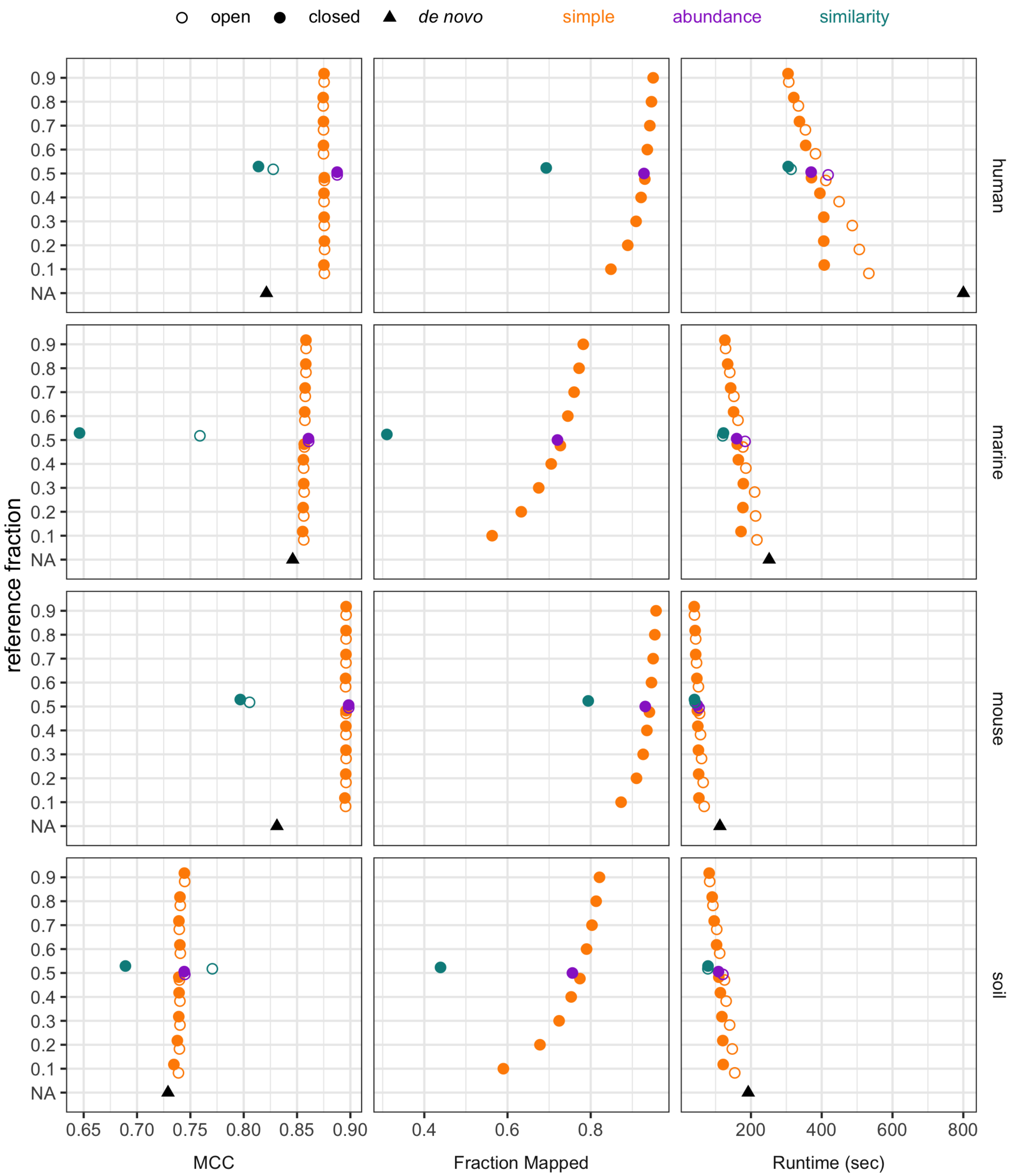

Figure 4: Split dataset strategy. The median MCC score, fraction of query sequences that mapped in closed-reference clustering, and runtime in seconds from repeating each clustering method 100 times. Each dataset was split into a reference and query fraction. References sequences were selected via a simple random sample, weighting sequences by relative abundance, or weighting by similarity to other sequences in the dataset. With the simple random sample method, dataset splitting was repeated with reference fractions ranging from 10% to 90% of the dataset and for 100 random seeds. *De novo* clustering each dataset is also shown for comparison.

methods for selecting the fraction of sequences to be used as the reference at a size of 50%: a simple random sample, weighting sequences by relative abundance, and weighting by similarity to other sequences in the dataset. OTU quality in terms of MCC was similar with the simple and abundance-weighted sampling (median MCCs of 0.87 and 0.87, respectively), but worse for similarity-weighted sampling (median MCC of 0.78). In closed-reference clustering mode, the fraction of sequences that mapped were similar for simple and abundance-weighted sampling (median fraction mapped of 0.97 and 0.97 respectively), but worse for similarity-weighted sampling (median fraction mapped of 0.90). While simple and abundance-weighted sampling produced better quality OTUs than similarity-weighted sampling, OptiFit performed faster on similarity-weighted samples with a median runtime of 99.4 seconds compared to 143.3 and 140.2 seconds for simple and abundance-weighted sampling, respectively. Thus, employing more complicated sampling strategies such as abundance-weighted and similarity-weighted sampling did not confer any advantages over selecting the reference via a simple random sample, and in fact decreased OTU quality in the case of similarity-weighted sampling.

Discussion

We developed a new algorithm for clustering sequences to existing OTUs and have demonstrated its suitability for reference-based clustering. OptiFit makes the iterative method employed by OptiClust available for tasks where reference-based clustering is required. We have shown that OTU quality is similar between OptiClust and OptiFit in open reference mode, regardless of strategy employed. Open reference OptiFit performs slower than OptiClust due to the additional *de novo* clustering step, so users may prefer OptiClust for tasks that do not require reference OTUs.

When clustering to public databases, OTU quality dropped in closed reference mode to different degrees depending on the database and dataset source, and no more than half of query sequences were able to be clustered into OTUs across any dataset/database combination. This may reflect limitations of reference databases, which are unlikely to contain sequences from novel microbes. This drop in quality was most notable with the RDP reference, which contained only 16,192 sequences compared to 173,648 sequences in SILVA and 174,979 in Greengenes. Note that Greengenes has not been updated since 2013 at the time of this writing, while SILVA and the RDP are updated regularly. We recommend that users who require an independent reference database opt for large databases with regular updates and good coverage of microbial diversity for their environment. Since OptiClust still performs faster than open reference OptiFit and creates higher quality OTUs than closed reference OptiFit with the database strategy, we recommend using OptiClust rather than clustering to a database whenever consistent OTUs are not required.

The OptiClust and OptiFit algorithms produced higher quality OTUs than VSEARCH in open reference, closed reference, or *de novo* modes. However, VSEARCH was able to cluster more sequences to OTUs than OptiFit in closed reference mode. While both OptiFit and VSEARCH use a distance or similarity threshold for determining how to cluster sequences into OTUs, VSEARCH is more permissive than OptiFit. The OptiFit and OptiClust algorithms use all of the sequences to define an OTU, preferring that all pairs of sequences (including reference and query sequences) in an OTU are within the distance threshold in order to maximize the MCC. In contrast, VSEARCH only requires each query sequence to be similar to the single centroid sequence that seeded the OTU. Because of this, VSEARCH sacrifices OTU quality by allowing more sequences to be clustered into OTUs.

When clustering with the split dataset strategy, OTU quality was remarkably similar when reference sequences were selected by a simple random sample or weighted by abundance, but quality was slightly worse when sequences were weighted by similarity. We recommend using a simple random sample since the more sophisticated reference selection methods do not offer any benefit. The similarity in OTU quality between OptiClust and OptiFit with this strategy demonstrates the suitability of using OptiFit to cluster sequences to existing OTUs, such as when comparing OTUs across studies. However, when consistent OTUs are not required, we recommend using OptiClust for *de novo* clustering over the split strategy with OptiFit since OptiClust is simpler to execute but performs similarly in terms of both run time and OTU quality.

We have developed a new clustering algorithm that allows users to produce high quality OTUs using already existing OTUs as a reference. Unlike existing reference-based methods that cluster query sequences to a single centroid sequence in each reference OTU, OptiFit considers all sequences in each reference OTU when clustering query sequences, resulting in OTUs of a similar high quality as those produced by the *de novo* OptiClust algorithm. Potential applications include clustering sequences to reference databases, comparing taxonomic composition of microbiomes across different studies, or using OTU-based machine learning models to make predictions on new data. OptiFit fills the missing option for clustering query sequences to existing OTUs that does not sacrifice OTU quality for consistency of OTU assignments.

Materials and Methods

Data Processing Steps

We downloaded 16S rRNA gene amplicon sequences from four published datasets isolated from soil (8), marine (9), mouse gut (10), and human gut (11) samples. These datasets contain sequences from the V4 region of the 16S rRNA gene and represent a selection of the broad types of natural communities that microbial ecologists study. We processed the raw sequences using mothur according to the Schloss Lab MiSeq SOP (15) and accompanying study by Kozich *et al.* (16). These steps included trimming and filtering for quality, aligning to the SILVA reference alignment (12), discarding sequences that aligned outside the V4 region, removing chimeric reads with UCHIME (17), and calculating distances between all pairs of sequences within each dataset prior to clustering.

Reference database clustering

To generate reference OTUs from independent databases, we downloaded sequences from the Greengenes database (v13_8_99) (6), SILVA non-redundant database (v132) (12), and the Ribosomal Database Project (v16) (13). These sequences were processed using the same steps outlined above followed by clustering sequences into *de novo* OTUs with OptiClust. Processed reads from each of the four datasets were clustered with OptiFit to the reference OTUs generated from each of the three databases. When reference clustering with VSEARCH, processed datasets were clustered directly to the unprocessed Greengenes 97% OTU reference alignment, since this method is how VSEARCH is typically used by the QIIME2 software reference-based clustering (7, 18).

Split dataset clustering

For each dataset, a fraction of the sequences was selected to be clustered *de novo* into reference OTUs with OptiClust. We used three methods for selecting the fraction of sequences to be used as the reference: a simple random sample, weighting sequences by relative abundance, and weighting by similarity to other sequences in the dataset. Dataset splitting was repeated with reference fractions ranging from 10% to 90% of the dataset and for 100 random seeds. For each dataset split, the remaining query sequences were clustered into the reference OTUs with OptiFit.

Benchmarking

Since OptiClust and OptiFit employ a random number generator to break ties when OTU assignments are of equal quality, they produce slightly different OTU assignments when repeated with different random seeds. To capture any variation in OTU quality or execution time, clustering was repeated with 100 random seeds for each combination of parameters and input datasets. We used the benchmark feature provided by Snakemake to measure the run time of every clustering job. We calculated the MCC on each set of OTUs to quantify the quality of clustering, as described by Westcott *et al.* (1).

Data and Code Availability

We implemented the analysis workflow in Snakemake (19) and wrote scripts in R (20), Python (21), and GNU bash (22). Software used includes mothur v1.45.0 (23), VSEARCH v2.13.3 (5), numpy (24), the tidyverse metapackage (25), R Markdown (26), ggtext (27), the SRA toolkit (28), and the conda environment manager (29). The complete workflow, manuscript, and conda environment are available at https://github.com/SchlossLab/Sovacool_OptiFit_2021.

Acknowledgements

We thank members of the Schloss Lab for their feedback on the figures.

KLS received support from the NIH Training Program in Bioinformatics (T32 GM070449). Salary support for PDS came from NIH grants R01CA215574 and U01AI124255. The funders had no role in study design, data collection and interpretation, or the decision to submit the work for publication.

Author Contributions

KLS wrote the analysis code, evaluated the algorithm, and wrote the original draft of the manuscript. SLW designed and implemented the OptiFit algorithm and assisted in debugging the analysis code. MBM and GAD contributed analysis code. PDS conceived the study, supervised the project, and assisted in debugging the analysis code. All authors reviewed and edited the manuscript.

References

1. **Westcott SL**, **Schloss PD**. 2017. OptiClust, an Improved Method for Assigning Amplicon-Based Sequence Data to Operational Taxonomic Units. mSphere **2**:e00073–17. doi:10.1128/mSphereDirect.00073-17.

2. **Schloss PD**. 2016. Application of a Database-Independent Approach To Assess the Quality of Operational Taxonomic Unit Picking Methods. mSystems **1**:e00027–16. doi:10.1128/mSystems.00027-16.

3. **Westcott SL**, **Schloss PD**. 2015. De novo clustering methods outperform reference-based methods for assigning 16S rRNA gene sequences to operational taxonomic units. PeerJ **3**:e1487. doi:10.7717/peerj.1487.

4. **Edgar RC**. 2010. Search and clustering orders of magnitude faster than BLAST. Bioinformatics **26**:2460–2461. doi:10.1093/bioinformatics/btq461.

5. **Rognes T**, **Flouri T**, **Nichols B**, **Quince C**, **Mahé F**. 2016. VSEARCH: A versatile open source tool for metagenomics. PeerJ **4**:e2584. doi:10.7717/peerj.2584.

6. **DeSantis TZ**, **Hugenholtz P**, **Larsen N**, **Rojas M**, **Brodie EL**, **Keller K**, **Huber T**, **Dalevi D**, **Hu P**, **Andersen GL**. 2006. Greengenes, a Chimera-Checked 16S rRNA Gene Database and Workbench Compatible with ARB. AEM **72**:5069–5072. doi:10.1128/AEM.03006-05.

7. Clustering sequences into OTUs using Q2-vsearch — QIIME 2 2021.2.0 documentation. https://docs.qiime2.org/2021.2/tutorials/otu-clustering/.

8. **Johnston ER**, **Rodriguez-R LM**, **Luo C**, **Yuan MM**, **Wu L**, **He Z**, **Schuur EAG**, **Luo Y**, **Tiedje JM**, **Zhou J**, **Konstantinidis KT**. 2016. Metagenomics Reveals Pervasive Bacterial Populations and Reduced Community Diversity across the Alaska Tundra Ecosystem. Front Microbiol **7**. doi:10.3389/fmicb.2016.00579.

9. **Henson MW**, **Pitre DM**, **Weckhorst JL**, **Lanclos VC**, **Webber AT**, **Thrash JC**. 2016. Artificial Seawater Media Facilitate Cultivating Members of the Microbial Majority from the Gulf of Mexico. mSphere **1**. doi:10.1128/mSphere.00028-16.

10. **Schloss PD**, **Schubert AM**, **Zackular JP**, **Iverson KD**, **Young VB**, **Petrosino JF**. 2012. Stabilization of the murine gut microbiome following weaning. Gut Microbes **3**:383–393. doi:10.4161/gmic.21008.

11. **Baxter NT**, **Ruffin MT**, **Rogers MAM**, **Schloss PD**. 2016. Microbiota-based model improves the sensitivity of fecal immunochemical test for detecting colonic lesions. Genome Med **8**:37. doi:10.1186/s13073-016-0290-3.

12. **Quast C**, **Pruesse E**, **Yilmaz P**, **Gerken J**, **Schweer T**, **Yarza P**, **Peplies J**, **Glöckner FO**. 2013. The SILVA ribosomal RNA gene database project: Improved data processing and web-based tools. Nucleic Acids Research **41**:D590–D596. doi:10.1093/nar/gks1219.

13. **Cole JR**, **Wang Q**, **Fish JA**, **Chai B**, **McGarrell DM**, **Sun Y**, **Brown CT**, **Porras-Alfaro A**, **Kuske CR**, **Tiedje JM**. 2014. Ribosomal Database Project: Data and tools for high throughput rRNA analysis. Nucl Acids Res **42**:D633–D642. doi:10.1093/nar/gkt1244.

14. **Navas-Molina JA**, **Peralta-Sánchez JM**, **González A**, **McMurdie PJ**, **Vázquez-Baeza Y**, **Xu Z**, **Ursell LK**, **Lauber C**, **Zhou H**, **Song SJ**, **Huntley J**, **Ackermann GL**, **Berg-Lyons D**, **Holmes S**, **Caporaso JG**, **Knight R**. 2013. Chapter Nineteen - Advancing Our Understanding of the Human Microbiome Using QIIME, p. 371–444. *In* DeLong, EF (ed.), Methods in Enzymology. Academic Press.

15. **Schloss PD**, **Westcott SL**. MiSeq SOP. https://mothur.org/MiSeq_SOP.

16. **Kozich JJ**, **Westcott SL**, **Baxter NT**, **Highlander SK**, **Schloss PD**. 2013. Development of a Dual-Index Sequencing Strategy and Curation Pipeline for Analyzing Amplicon Sequence Data on the MiSeq Illumina Sequencing Platform. Appl Environ Microbiol **79**:5112–5120. doi:10.1128/AEM.01043-13.

17. **Edgar RC**, **Haas BJ**, **Clemente JC**, **Quince C**, **Knight R**. 2011. UCHIME improves sensitivity and speed of chimera detection. Bioinformatics **27**:2194–2200. doi:10.1093/bioinformatics/btr381.

18. **Bolyen E**, **Rideout JR**, **Dillon MR**, **Bokulich NA**, **Abnet CC**, **Al-Ghalith GA**, **Alexander H**, **Alm EJ**, **Arumugam M**, **Asnicar F**, **Bai Y**, **Bisanz JE**, **Bittinger K**, **Brejnrod A**, **Brislawn CJ**, **Brown CT**, **Callahan BJ**, **Caraballo-Rodríguez AM**, **Chase J**, **Cope EK**, **Da Silva R**, **Diener C**, **Dorrestein PC**, **Douglas GM**, **Durall DM**, **Duvallet C**, **Edwardson CF**, **Ernst M**, **Estaki M**, **Fouquier J**, **Gauglitz JM**, **Gibbons SM**, **Gibson DL**, **Gonzalez A**, **Gorlick K**, **Guo J**, **Hillmann B**, **Holmes S**, **Holste H**, **Huttenhower C**, **Huttley GA**, **Janssen S**, **Jarmusch AK**, **Jiang L**, **Kaehler BD**, **Kang KB**, **Keefe CR**, **Keim P**, **Kelley ST**, **Knights D**, **Koester I**, **Kosciolek T**, **Kreps J**, **Langille MGI**, **Lee J**, **Ley R**, **Liu Y-X**, **Loftfield E**, **Lozupone C**, **Maher M**, **Marotz C**, **Martin BD**, **McDonald D**, **McIver LJ**, **Melnik AV**, **Metcalf JL**, **Morgan SC**, **Morton JT**, **Naimey AT**, **Navas-Molina JA**, **Nothias LF**, **Orchanian SB**, **Pearson T**, **Peoples SL**, **Petras D**, **Preuss ML**, **Pruesse E**, **Rasmussen LB**, **Rivers A**, **Robeson MS**, **Rosenthal P**, **Segata N**, **Shaffer M**, **Shiffer A**, **Sinha R**, **Song SJ**, **Spear JR**, **Swafford AD**, **Thompson LR**, **Torres PJ**, **Trinh P**, **Tripathi A**, **Turnbaugh PJ**, **Ul-Hasan S**, **van der Hooft JJJ**, **Vargas F**, **Vázquez-Baeza Y**, **Vogtmann E**, **von Hippel M**, **Walters W**, **Wan Y**, **Wang M**, **Warren J**, **Weber KC**, **Williamson CHD**, **Willis AD**, **Xu ZZ**, **Zaneveld JR**, **Zhang Y**, **Zhu Q**, **Knight R**, **Caporaso JG**. 2019. Reproducible, interactive, scalable and extensible microbiome data science using QIIME 2. Nat Biotechnol **37**:852–857. doi:10.1038/s41587-019-0209-9.

19. **Köster J**, **Rahmann S**. 2012. Snakemake — a scalable bioinformatics workflow engine. Bioinformatics **28**:2520–2522. doi:10.1093/bioinformatics/bts480.

20. **R Core Team**. 2020. R: A language and environment for statistical computing. Manual, R Foundation for Statistical Computing, Vienna, Austria.

21. **Van Rossum G**, **Drake FL**. 2009. Python 3 Reference Manual | Guide books.

22. Bash Reference Manual. https://www.gnu.org/software/bash/manual/bash.html.

23. **Schloss PD**, **Westcott SL**, **Ryabin T**, **Hall JR**, **Hartmann M**, **Hollister EB**, **Lesniewski RA**, **Oakley BB**, **Parks DH**, **Robinson CJ**, **Sahl JW**, **Stres B**, **Thallinger GG**, **Van Horn DJ**, **Weber CF**. 2009. Introducing mothur: Open-source, platform-independent, community-supported software for describing and comparing microbial communities. Applied and Environmental Microbiology **75**:7537–7541. doi:10.1128/AEM.01541-09.

24. **Harris CR**, **Millman KJ**, **van der Walt SJ**, **Gommers R**, **Virtanen P**, **Cournapeau D**, **Wieser E**, **Taylor J**, **Berg S**, **Smith NJ**, **Kern R**, **Picus M**, **Hoyer S**, **van Kerkwijk MH**, **Brett M**, **Haldane A**, **del Río JF**, **Wiebe M**, **Peterson P**, **Gérard-Marchant P**, **Sheppard K**, **Reddy T**, **Weckesser W**, **Abbasi H**, **Gohlke C**, **Oliphant TE**. 2020. Array programming with NumPy. Nature **585**:357–362. doi:10.1038/s41586-020-2649-2.

25. **Wickham H**, **Averick M**, **Bryan J**, **Chang W**, **McGowan LD**, **François R**, **Grolemund G**, **Hayes A**, **Henry L**, **Hester J**, **Kuhn M**, **Pedersen TL**, **Miller E**, **Bache SM**, **Müller K**, **Ooms J**, **Robinson D**, **Seidel DP**, **Spinu V**, **Takahashi K**, **Vaughan D**, **Wilke C**, **Woo K**, **Yutani H**. 2019. Welcome to the Tidyverse. Journal of Open Source Software **4**:1686. doi:10.21105/joss.01686.

26. **Xie Y**, **Allaire JJ**, **Grolemund G**. 2018. R Markdown: The Definitive Guide. Taylor & Francis, CRC Press.

27. **Wilke CO**. 2020. Ggtext: Improved text rendering support for 'Ggplot2'. Manual.

28. SRA-Tools - NCBI. http://ncbi.github.io/sra-tools/.

29. 2016. Anaconda Software Distribution. Anaconda Documentation. Anaconda Inc.