OptiFit: a fast method for fitting amplicon sequences to existing OTUs

2021-06-02

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, Ann Arbor, MI 48109

2 Department of Microbiology and Immunology, University of Michigan, Ann Arbor, MI 48109

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

Abstract

Assigning amplicon sequences to Operational Taxonomic Units (OTUs) is an important step in characterizing the composition of microbial communities across large datasets. OptiClust, a *de novo* OTU clustering method in the mothur program, has been shown to produce higher quality OTU assignments than other methods and at comparable or faster speeds (1, 2). A notable difference between *de novo* clustering and database-dependent methods is that OTU assignments clustered with *de novo* methods are not stable when new sequences are added to a dataset (3). 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 deploying a machine learning model where OTUs are features (4). To provide an efficient and robust method to fit amplicon sequence data to existing OTUs, we developed the OptiFit algorithm as a new component of the mothur program.

TODO: summarize results & conclusion

Importance

TODO

Introduction

Amplicon sequencing has become a mainstay of microbial ecology and host-associated microbiome research. Researchers can affordably generate millions of sequences to characterize the composition of hundreds of samples from culture-independent microbial communities. In a typical analysis pipeline, 16S rRNA gene sequences are assigned to Operational Taxonomic Units (OTUs) to facilitate comparison of taxonomic composition between communities. A distance threshold of 3% (or sequence similarity of 97%) is commonly used to cluster sequences into OTUs based on either a reference database or 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, 5).

There are three main categories of OTU clustering algorithms: closed reference, open reference, and *de novo* clustering. Closed reference methods assign sequences to a set of pre-made OTUs generated from reference sequences. If a query sequence is not within the distance threshold to any of the reference sequences, it is discarded. 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 if they are not represented by a similar sequence in the database. *De novo* methods cluster sequences based on their distance to each other, without the use of an external reference. *De novo* clustering overcomes the limitations of reference databases by considering only sequences in the dataset, but is more computationally intensive and generates different OTU assignments when new sequences are introduced. Unstable OTU assignments make it difficult to use *de novo* clustering to compare taxonomic composition of communities between studies or to use machine learning models trained with *de novo* OTUs to make predictions on new data. Open reference methods take a hybrid approach, first performing closed reference clustering, then any sequences that cannot be assigned to reference OTUs are clustered *de novo* to create additional OTUs. Previous studies found that the OptiClust *de novo* clustering algorithm created the highest quality OTU assignments of all clustering methods based on the Matthews correlation coefficient (MCC) (1). As a result, we have recommended OptiClust as the preferred method for OTU clustering whenever OTU stability is not required.

- **TODO: current method for open/closed is vsearch against greengenes.**
- **TODO: use word "map" for what vsearch does, "fit" for what optifit does.**
- **TODO**: 2 categories of clustering: *de novo* and reference based. separate paragraphs. describe opticlust first in de novo paragraph. 2nd paragraph: ref methods are good cause they're fast and don't use much ram. dependent on order of db. people use greengenes, which are rep seqs from 3% otus from full length.
- reader should know what opticlust is, closed & open ref clustering is, strengths & weakness are of each. then we solve these problems.
- **TODO:** note that greengenes is defunct now?!

To overcome the limitations of *de novo* clustering while maintaining OTU quality, we developed OptiFit, a reference-based clustering algorithm in the mothur program which takes existing OTUs as the reference to fit new sequences to. **TODO: more words here?** Here, we tested the OptiFit algorithm with the reference as a database 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 (6), marine (7), mouse gut (8), and human gut (9) samples.

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). 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 assign to the reference OTUs, and the sequence pairs that are within the distance threshold (e.g. 0.03). Initially, query sequences are placed in singleton 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 assigned to a reference OTU. 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 assigned to references OTUs are discarded, and the results will 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) fit 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 fit the query sequences to the reference OTUs. We clustered sequences from four datasets isolated from soil (6), marine (7), mouse gut (8), and human gut (9) samples to test the performance of OptiFit with both of these strategies.

Reference clustering with public databases

While *de novo* clustering produces high 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 assigned to 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 fit each dataset to three databases of reference OTUs: the Greengenes database, the SILVA non-redundant database, and the Ribosomal Database Project (RDP) (10–12). Reference OTUs for each database were created by performing *de novo* clustering with OptiClust at a distance threshold of 3%. The *de novo* MCC scores were 0.72, 0.74, and 0.73 for gg, rdp, and silva, respectively. Fitting sequences to Greengenes and SILVA in closed reference mode performed similarly, with median MCC scores of 0.8 and 0.72 respectively, while the median MCC dropped to 0.33 when fitting to RDP. For comparison, clustering datasets *de novo* 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 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.

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

(36.4%) or 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 assign 100% of sequences to OTUs. The database chosen affects the final OTU assignments considerably in terms of both MCC score and fraction of query sequences that could be fit to 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. 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 method 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 map 35.1% more query sequences than OptiFit to the Greengenes reference database. This is because VSEARCH only considers the 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 resulted in fewer sequences being fit 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 is too unrelated to the samples of interest, such as when samples contain low abundant or 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 fit new sequences to existing OTUs when comparing OTUs across studies or when making predictions with machine learning models. To determine how well OptiFit performs for fitting 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 fit 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. OTU quality was highly similar to that from OptiClust regardless of mode (0.29% difference in median MCC). In closed reference mode, OptiFit was able to fit 81%

of query sequences to reference OTUs with the split strategy, a great improvement over the average 43.1% of sequences fit to the greengenes database. In terms of run time, closed and open reference OptiFit performed faster than OptiClust on whole datasets by 29.0% and 20.2 respectively. The split dataset strategy also performed 11.9% faster than the database strategy in closed reference mode and 30.4% 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 an even split of the data into reference and query fractions, we then wanted to investigate whether there was an optimal reference fraction size. To test the best reference size, reference sizes from 10% to 80% of the sequences were created, with the remaining sequences used for the query. OTU quality was remarkably stable across reference fraction sizes. For example, splitting the human dataset 100 times yielded a coefficient of variation of 0.00063 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 762.8 with 10% of sequences in the reference and 470.9 with 90% of sequences in the reference. 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 0.88 with 10% of sequences in the reference and 0.96 with 90% of sequences in the reference. These trends hold for the other datasets as well (see supplemental figure **TODO**). Thus, the reference fraction does not affect OTU quality in terms of MCC score, but does affect the run time and the fraction of sequences that map during closed reference mode.

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 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 0.82 and 0.84 respectively), but worse for similarity-weighted sampling (median MCC of 0.71). In closed-reference clustering mode, the fraction of sequences that mapped were similar for simple and abundance-weighted sampling (median fraction mapped of 0.96 and 0.95 respectively), but worse for similarity-weighted sampling (median fraction mapped of 0.85). 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 113.1 seconds compared to 165.7 and 165.4 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 fitting 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 fitting 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 fit to OTUs across any dataset/database combination. This may reflect limitations of reference databases, which are unlikely to contain sequences from rare and novel microbes. This drop in quality was most notable with RDP, which contains only about 21,000 sequences compared to over 200,000 sequences in SILVA and Greengenes each at the time of this writing. We recommend that users who require an independent reference database opt for large databases with good coverage of microbial diversity. Since OptiClust 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 fitting to a database whenever stable OTUs are not required for the study at hand.

The OptiClust and OptiFit algorithms provided by mothur produced higher quality OTUs than VSEARCH in open reference, closed reference, or *de novo* modes. However, VSEARCH was able to map more sequences to OTUs than OptiFit in closed reference mode. While both mothur and VSEARCH use a distance or similarity threshold for determining how to assign sequences to OTUs, VSEARCH is more permissive than mothur. The OptiFit and OptiClust algorithms use all of the sequences to define an OTU, requiring that all pairs of sequences (including reference and query sequences) in an OTU are within the distance threshold without penalizing the MCC. In contrast, VSEARCH only requires each query sequence to be similar to the single sequence that seeded the OTU. In this way, VSEARCH sacrifices OTU quality in order to allow more sequences to fit to OTUs. Users who require closed reference clustering to the Greengenes database may prefer to use VSEARCH if they wish to maximize the fraction of sequences that can be fit at the cost of OTU quality. However, mothur's OptiClust or OptiFit are recommended for *de novo* or open reference clustering to produce OTUs of the highest possible quality.

When fitting 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 fit sequences to existing OTUs, such as when using already-trained machine learning models to make predictions on new data or comparing OTUs across studies. However, when stable 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.

TODO: big picture concluding paragraph. We have developed a new clustering algorithm that allows users to produce high quality OTUs using already existing OTUs as a reference. **TODO: Point to courtney's paper metaphorically. wow what a cool application someone should do *wink wink*.**

Materials and Methods

Data Processing Steps

We downloaded 16S rRNA gene amplicon sequences from four published datasets isolated from soil (6), marine (7), mouse gut (8), and human gut (9) samples. Raw sequences were processed using mothur according to the Schloss Lab MiSeq SOP as described in the mothur wiki and accompanying study by Kozich *et al.* (13, 14). These steps included trimming and filtering for quality, aligning to the SILVA reference alignment (11), discarding sequences that aligned outside the V4 region, removing chimeric reads with UCHIME (15), 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) (10), SILVA non-redundant database (v132) (11), and the Ribosomal Database Project (v16) (12). 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 fit directly to the unprocessed Greengenes reference alignment, since this method is how VSEARCH is typically used by the QIIME2 software reference-based clustering (16, 17).

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 80% of the dataset and for 100 random seeds. For each dataset split, the remaining sequences were assigned to 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 (18) and wrote scripts in R (19), Python (20), and GNU bash (21). Software used includes mothur v1.45.0 (2), VSEARCH v2.13.3 (22), numpy (23), the Tidyverse metapackage (24), R Markdown (25), the SRA toolkit (26), and the conda environment manager (27). The complete workflow, manuscript, and conda environment are available at **TODO: UPDATED REPO LINK**.

Acknowledgements

KLS received support from the NIH Training Program in Bioinformatics (T32 GM070449).

PDS received support from **TODO: Pat's grant(s)**.

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.

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**, **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.

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. **Topçuoğlu BD**, **Lesniak NA**, **Ruffin M**, **Wiens J**, **Schloss PD**. 2019. Effective application of machine learning to microbiome-based classification problems. bioRxiv 816090. doi:10.1101/816090.

5. **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.

6. **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.

7. **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.

8. **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.

9. **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.

10. **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.

11. **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.

12. **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.

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

14. **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.

15. **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.

16. **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.

17. Clustering sequences into OTUs using q2-vsearch QIIME 2 2021.2.0 documentation. https://docs.qiime2.org/2021.2/tutorials/otu-clustering/.

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

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

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

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

22. **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.

23. **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.

24. **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.

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

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

27. 2016. Anaconda Software Distribution. Anaconda Documentation. Anaconda Inc.

References