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

2021-01-22

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

2

Introduction

- **TODO: broad context; importance of amplicon sequencing**
- **TODO: describe reference vs de novo clustering. trade-offs. machine learning use-case.**
- **TODO: mention OptiClust and vsearch.**

Results

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.

- **TODO: brief description of OptiFit algorithm**
- **TODO: open & closed reference modes**
- **TODO: mcc score; only considers query seqs (when printref=f)**

To evaluate the OptiFit algorithm and compare to existing methods, we used four published datasets isolated from soil (5), marine (6), mouse gut (7), and human gut (8) samples. There are two strategies for generating OTUs with OptiFit: 1) fit sequences to reference OTUs of an independent database, or 2) split the dataset into a reference and query fraction, then fit the query sequences to OTUs generated by clustering the reference sequences *de novo.* For each dataset repeated with 100 random seeds, we generated OTUs with OptiFit using both strategies, and also clustered *de novo* OTUs with OptiClust for comparison. All clustering was performed at a sequence dissimilarity threshold of 0.03 and OTU quality was evaluated using the Matthews Correlation Coefficient (MCC) as described previously (3, 9). We calculated the fraction of query sequences that were fit to existing OTUs in closed-reference mode as an additional measure of quality for this mode.

3

Reference clustering with public databases

To evaluate reference-based clustering with independent databases, we fit each dataset to reference OTUs generated by *de novo* clustering the Greengenes database (v13_8_99), Silva non-redundant database (v132), and the Ribosomal Database Project (RDP; v16). In open-reference mode, OTU quality was similar between fitting the datasets to reference OTUs with OptiFit and clustering the datasets *de novo* with OptiClust. This held true for all datasets and reference databases. However, in closed-reference mode, OTU quality was slightly worse when fitting to greengenes and silva, and much worse when fitting to RDP as compared to OptiClust. No more than half of query sequences were fit to reference OTUs in closed-reference mode across any dataset/database combination. OptiFit was able to fit more query sequences to reference OTUs created with the greengenes and silva databases than with RDP. In terms of runtime, closed-reference OptiFit outperformed OptiClust, while OptiClust outperformed open-reference OptiFit.

To compare to existing software, vsearch was used to cluster OTUs *de novo* or with reference-based clustering to the greengenes database. For all datasets and clustering methods (*de novo*, open reference, and closed reference), mothur's clustering algorithms produced higher quality OTUs than vsearch. When closed-reference clustering against the greengenes database, vsearch was able to map more query sequences to the reference than mothur's OptiFit algorithm. In terms of runtime, OptiFit generally performed faster than vsearch when reference clustering, while vsearch *de novo* clustering outperformed OptiClust.

Reference clustering with split datasets

Datasets were randomly split into a reference fraction and a query fraction. Reference sizes from 10% to 80% of the sequences were created, with the remaining sequences used for the query. Reference sequences were clustered *de novo* with OptiClust, then

4

₆₆ query sequences were fit to the *de novo* OTUs with OptiFit.

₆₇ OTU quality from the split dataset strategy with OptiFit was highly similar to that from *de*

₆₈ *novo* clustering the whole dataset with OptiClust regardless of mode. OTU quality was

₆₉ remarkably stable across 100 different random seeds. In terms of runtime, closed-reference

₇₀ OptiFit performed faster than OptiClust on whole datasets. In open-reference mode,

₇₁ OptiClust performed faster than OptiFit only when the OptiFit reference fraction was

₇₂ 30% or less. The split dataset strategy performed just as well as the database strategy

₇₃ in open-reference mode regardless of database used, and outperformed the database

₇₄ strategy in closed-reference mode.

₇₅ We also tested three methods for selecting the sequences to be used as the reference;

₇₆ a simple random sample, weighting sequences by relative abundance, and weighting by

₇₇ similarity to other sequences in the dataset. OTU quality was similar with the simple and

₇₈ abundance-weighted sampling, but slightly worse with similarity-weighted sampling. In

₇₉ closed-reference mode, The fraction of query sequences that can be fit to the reference

₈₀ OTUs decreases as the reference fraction increases.

Discussion

₈₂ We developed a new algorithm for fitting sequences to existing OTUs and have

₈₃ demonstrated its suitability for reference-based clustering. OptiFit makes the iterative

₈₄ method employed by OptiClust available for tasks where reference-based clustering is

₈₅ required. We have shown that OTU quality is similar between OptiClust and OptiFit in

₈₆ open-reference mode, regardless of strategy employed. Open-reference OptiFit does

₈₇ perform slower than OptiClust due to the additional *de novo* clustering step, so users may

₈₈ prefer OptiClust for tasks that do not require reference OTUs.

₈₉ 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. We recommend users who require an independent reference database opt for Greengenes or Silva instead. 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 where possible.

The mothur algorithms produced higher quality OTUs than vsearch in open-reference, closed-reference, or *de novo* modes. However, vsearch was able to fit more sequence into OTUs than OptiFit in closed-reference mode. While both mothur and vsearch use a dissimilarity threshold for determining how to assign sequences into OTUs, vsearch is more permissive than mothur. Mothur requires that all pairs of sequences in an OTU are within the dissimilarity threshold without penalizing the MCC, while vsearch only requires sequences to be similar to one other sequence in 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 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 preferred for *de novo* or open-reference clustering.

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.

- **TODO: big picture concluding paragraph**

Materials and Methods

Data Processing Steps

Benchmarking

Data and Code Availability

We implemented the analysis workflow in Snakemake (10) and relied on Python (11), R (12), and GNU bash. Software used includes mothur v1.45.0 (2), vsearch v2.13.3 (13), numpy (14), the Tidyverse metapackage (15), R Markdown (16), and the conda environment manager (17). The complete workflow, manuscript, and conda environment are available at **TODO: UPDATED REPO LINK**.

Acknowledgements

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

7

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. **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.

6. **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.

7. **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.

8. **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.

9. **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.

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

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

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

13. **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.

14. **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.

15. **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**,

183 **Yutani H**. 2019. Welcome to the Tidyverse. Journal of Open Source Software **4**:1686.

184 doi:10.21105/joss.01686.

185 16. **Xie Y**, **Allaire JJ**, **Grolemund G**. 2018. R Markdown: The Definitive Guide. Taylor &

186 Francis, CRC Press.

187 17. 2016. Anaconda Software Distribution. Anaconda Documentation. Anaconda Inc.

188 ## References