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

2020-12-15

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

Importance

Introduction

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: Describe OptiClust and vsearch.

Results

- TODO: brief description of OptiFit algorithm. the method we came up with is the result. OptiClust paper had toy example to walk through the algorithm.
- TODO: Paragraph at beginning to describe datasets: why picked, how processed.

We used four published datasets isolated from soil (5), marine (6), mouse (7), and human (8) samples.

- 16S rRNA gene sequence datasets from human gut, mouse gut, marine, and soil environments as well as the greengenes, silva, and RDP public databases were processed with mothur and clustered *de novo* with mothur's OptiClust algorithm.
- All clustering was performed at a distance threshold of 0.03 and repeated with 100

different random seeds.

- The Matthews Correlation Coefficient was calculated to serve as a measure of OTU quality. OTU quality was evaluated using the Matthews Correlation Coefficient (MCC) with a sequence dissimilarity threshold of 0.03% as described previously (3, 9).

Reference clustering with public databases

- Public reference databases were clustered *de novo* using OptiClust, then query datasets were fit to the *de novo* OTUs using OptiFit.
  - 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. OptiFit was able to map 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 out-performed open-reference OptiFit.

Reference clustering with split datasets

- TODO: compare optifit performance for split dataset vs public database

- TODO: double-check with Sarah that MCCs are from full OTU dataset or just query sequnce OTUs

- TODO: double-check reference fractions aren't flipped (fraction mapped plot looks weird)

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

  - OTU quality from fitting split datasets was highly similar to that from *de novo* clustering the whole dataset.
  - Closed-reference OptiFit with split datasets was faster than OptiClust on whole datasets.
    * OptiClust performed faster than open-reference OptiFit only when the OptiFit reference fraction was 30% or less.
  - Different methods for selecting the sequences to be used as the reference were tested; a simple random sample, weighting sequences by relative abundance, and weighting by similarity to other sequences in the dataset.
    * OTU quality was similar with the simple and abundance-weighted sampling, but slightly worse with similarity-weighted sampling.
    * The fraction of query sequences that are able to be fit to the reference OTUs in closed-reference mode decreases as the reference fraction increases.

Comparison to vsearch

- TODO: move this into first section with public database.

- vsearch is more permissive than mothur (radius vs diameter with 0.03 dissimilarity threshold).

- 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 genearlly performed faster than vsearch when reference clustering, while vsearch *de novo* clustering outperformed OptiClust.

Discussion

Materials and Methods

Sequence 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. Dependencies include 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). A reproducible version of the workflow, manuscript, and conda environment is 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. **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**, **Yutani H**. 2019. Welcome to the Tidyverse. Journal of Open Source Software **4**:1686. doi:10.21105/joss.01686.

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

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

References