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

2022-01-13

Kelly L. Sovacool[1], Sarah L. Westcott[2], M. Brodie Mumphrey[1], Gabrielle A. Dotson[1], Patrick D. Schloss[2,3,†]

1 Department of Computational Medicine and Bioinformatics, University of Michigan

2 Department of Microbiology and Immunology, University of Michigan

3 Center for Computational Medicine and Bioinformatics, 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~~ to previously clustered datasets without clustering all sequences again, 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 OTU~~ resulting in assignments that are ~~significantly~~ worse than those generated by *de novo* methods. To provide an efficient ~~and robust~~ method to fit ~~amplicon sequence data~~ sequences to existing OTUs, we developed the OptiFit algorithm. Inspired by ~~OptiClust~~the *de novo* OptiClust algorithm, OptiFit 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~~ strategies: 1) clustering to a reference database or ~~by~~ 2) splitting the dataset into a reference and query set~~and clustering the query sequences to the reference set after clustering it using OptiClust~~, clustering the references using OptiClust, then clustering the queries to the references. The result is an improved implementation of ~~closed and open-reference~~ reference-based clustering. OptiFit produces ~~OTUs of similar quality as OptiClust and~~ similar quality OTUs as OptiClust 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

28 option for users ~~who require~~ requiring consistent OTU assignments at the same quality

29 afforded by *de novo* clustering methods.

Importance

31 Advancements in DNA sequencing technology have allowed researchers to affordably

32 generate millions of sequence reads from microorganisms in diverse environments.

33 Efficient and robust software tools are needed to assign microbial sequences into

34 taxonomic groups for characterization and comparison of communities. The OptiClust

35 algorithm produces high quality groups by comparing sequences to each other, but the

36 assignments can change when new sequences are added to a dataset, making it difficult

37 to compare different studies. Other approaches assign sequences to groups by comparing

38 them to sequences in a reference database to produce consistent assignments, but the

39 quality of the groups produced is reduced compared to OptiClust. We developed OptiFit, a

40 new reference-based algorithm that produces consistent yet high quality assignments like

41 OptiClust. OptiFit allows researchers to compare microbial communities across different

42 studies or add new data to existing studies without sacrificing the quality of the group

43 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). OTU quality can be conceptualized as how well the OTU assignments match the definition set by the distance threshold, i.e. whether sequence pairs that are at least as similar as the distance threshold are assigned to the same OTU and sequence pairs that are more dissimilar than the distance threshold are assigned to different OTUs.

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 (4, 5). In methods employing greedy clustering algorithms, only the distance between each sequence and a representative centroid sequence in the OTU is considered while clustering. As a result, distances between pairs of sequences in the same OTU are frequently larger 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

70 positives.

71 A limitation of *de novo* clustering is that different OTU assignments will be produced

72 when new sequences are added to a dataset, making it difficult to use *de novo* clustering

73 to compare OTUs between different studies. Furthermore, since *de novo* clustering

74 requires calculating and comparing distances between all sequences in a dataset, the

75 execution time can be slow and memory requirements can be prohibitive for very large

76 datasets. Reference clustering attempts to overcome the limitations of *de novo* clustering

77 methods by using a representative set of sequences from a database, with each reference

78 sequence seeding an OTU. Commonly, the Greengenes set of representative full length

79 sequences clustered at 97% similarity is used as the reference with VSEARCH (5–7).

80 Query sequences are then clustered into OTUs based on their similarity to the reference

81 sequences. Any query sequences that are not within the distance threshold to any of

82 the reference sequences are either thrown out (closed reference clustering) or clustered

83 *de novo* to create additional OTUs (open reference clustering). While reference-based

84 clustering is generally fast, it is limited by the diversity of the reference database. Novel

85 sequences in the sample will be lost in closed reference mode if they are not represented

86 by a similar sequence in the database. ~~Previous studies~~ We previously found that the

87 OptiClust *de novo* clustering algorithm created the highest quality OTU assignments of all

88 clustering methods (1).

89 To overcome the limitations of current reference-based and *de novo* clustering algorithms

90 while maintaining OTU quality, we developed OptiFit, a reference-based clustering

91 algorithm. While other tools represent reference OTUs with a single sequence, OptiFit

92 uses ~~multiple~~ all sequences in existing OTUs as the reference and fits new sequences to

93 those reference OTUs. In contrast to other tools, OptiFit considers all pairwise distance

94 scores between reference and query sequences when assigning sequences to OTUs

95 in order to produce OTUs of the highest possible quality. Here, we tested the OptiFit

5

96 algorithm with the reference as a public database (e.g. Greengenes) or *de novo* OTUs

97 generated using a reference set from the full dataset and compared the performance to

98 existing tools. To evaluate the OptiFit algorithm and compare to existing methods, we used

99 four published datasets isolated from soil (8), marine (9), mouse gut (10), and human gut

100 (11) samples. OptiFit is available within the mothur software program.

Results

The OptiFit algorithm

103 OptiFit leverages the method employed by OptiClust of iteratively assigning sequences

104 to OTUs to produce the highest quality OTUs possible, and extends this method for

105 reference-based clustering. OptiClust first seeds each sequence into its own OTU as a

106 singleton. Then for each sequence, OptiClust considers whether the sequence should

107 move to a different OTU or remain in its current OTU, choosing the option that results in a

108 better ~~Matthews correlation coefficient (MCC)~~ MCC score (1). The MCC uses all values

109 from a confusion matrix and ranges from negative one to one, with a score of one occurring

110 when all sequence pairs are true positives and true negatives~~and~~, a score of negative

111 one occurring when all pairs are false positives and false negatives~~.~~, and a score of zero

112 when there are equal numbers of true and false assignments (i.e. no better than random

113 guessing). Sequence pairs that are similar to each other (i.e. within the distance threshold)

114 are counted as true positives if they are clustered into the same OTU, and false negatives

115 if they are not in the the same OTU. Sequence pairs that are not similar to each other are

116 true negatives if they are not clustered into the same OTU, and false positives if they are

117 not in the same OTU. Thus, a pair of sequences is considered correctly assigned when

118 their OTU assignment matches the OTU definition set by the distance threshold. OptiClust

119 iterations continue until the MCC stabilizes or until a maximum number of iterations is

120 reached. This process produces *de novo* OTU assignments with the most optimal MCC

₁₂₁ given the input sequences.

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 of 3%. Previously, 50 reference sequences were clustered *de novo* with OptiClust (see the OptiClust supplemental text (1)). Reference sequences A through Q (colored orange) were within the distance threshold to at least one other reference sequence; the remaining reference sequences formed additional singleton OTUs (not shown). The goal of OptiFit is to assign the query sequences W through Z (colored green) to the reference OTUs. Here, there are 50 reference sequences and 4 query sequences which make 1,431 sequence pairs, of which 23 pairs are within the 3% distance threshold. Initially (step 1), OptiFit places each query sequence in its own OTU, resulting in 14 true positives, 9 false negatives, 0 false positives, and 1,408 true negatives for an MCC score of 0.78. 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 (steps 2-4). 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 was needed. Note that sequence Z was dissimilar from all other sequences and thus it remained a singleton. The final MCC score is 0.91 with 20 true positives, 3 false negatives, 1 false positive, and 1407 true negatives.

₁₂₂ OptiFit begins where OptiClust ends, starting with a list of reference OTUs and their

₁₂₃ sequences, a list of query sequences to cluster to the reference OTUs, and the sequence

₁₂₄ pairs that are within the distance threshold (e.g. 0.03) (Figure 1). Initially, all query

₁₂₅ sequences are placed into separate OTUs. Then, the algorithm iteratively reassigns the

₁₂₆ query sequences to the reference OTUs to optimize the MCC. Alternatively, a sequence

₁₂₇ will remain unassigned if the MCC value is maximized when the sequence is a singleton

₁₂₈ rather than clustered into a reference OTU. All query and reference sequence pairs are

₁₂₉ considered when calculating the MCC. This process is repeated until the MCC changes by

₁₃₀ no more than 0.0001 (default) or until a maximum number of iterations is reached (default:

₁₃₁ 100). In the closed reference mode, any query sequences that cannot be clustered into

₁₃₂ reference OTUs are discarded, and the results only contain OTUs that exist in the original

₁₃₃ reference. In the open reference mode, unassigned query sequences are clustered *de*

₁₃₄ *novo* using OptiClust to generate new OTUs. The final MCC is reported with the best

135 OTU assignments. There are two strategies for generating OTUs with OptiFit: 1) cluster

136 the query sequences to reference OTUs generated by *de novo* clustering an independent

137 database, or 2) split the dataset into a reference and query fraction, cluster the reference

138 sequences *de novo*, then cluster the query sequences to the reference OTUs.

Reference clustering with public databases

140 To test how OptiFit performs for reference-based clustering, we clustered each dataset to

141 three databases of reference OTUs: the Greengenes database v13_8_99 (6), the SILVA

142 non-redundant database v132 (12), and the Ribosomal Database Project (RDP) v16 (,

143 13). Reference OTUs for each database were created by performing *de novo* clustering

144 with OptiClust at a distance threshold of 3% using the V4 region of each sequence (see

145 Figure 2). After trimming to the V4 region, the databases contained 174,979, 16,192, and

146 173,648 unique sequences and produced *de novo* MCC scores of 0.72, 0.74, and 0.73

147 for Greengenes, RDP, and SILVA, respectively. Clustering ~~sequences~~ query sequences

148 with OptiFit to Greengenes and SILVA in closed reference mode performed similarly,

149 with median MCC scores of 0.85 and 0.77 respectively, while the median MCC was

150 0.35 when clustering to RDP (Figure 3; "db: Greengenes", "db: SILVA", and "db: RDP").

151 For comparison, clustering datasets with OptiClust produced an average MCC score

152 of ~~0.87~~0.86 (Figure 3; "*de novo*"). This gap in OTU quality mostly disappeared when

153 clustering in open reference mode, which produced median MCCs of 0.86 with Greengenes,

154 ~~0.85~~ 0.86 with SILVA, and 0.86 with the RDP. Thus, open reference OptiFit produced OTUs

155 of very similar quality as *de novo* clustering with OptiClust, and closed reference OptiFit

156 followed closely behind as long as a suitable reference database was chosen.

157 Since closed reference clustering does not cluster query sequences that could not be

158 clustered into reference OTUs, an additional measure of clustering performance to consider

159 is the fraction of query sequences that were able to be clustered. On average, more

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.

160   sequences were clustered with Greengenes as the reference (~~59.1~~59%) than with SILVA

161   (~~50.0~~50%) or with the RDP (~~9.8~~9.7%) (Figure 3). This mirrored the result reported above

162   that Greengenes produced better OTUs in terms of MCC score than either SILVA or

163   RDP. Note that *de novo* and open reference clustering methods always cluster 100%

164   of sequences into OTUs. The database chosen affects the final closed reference OTU

165   assignments considerably in terms of both MCC score and fraction of query sequences

166   that could be clustered into the reference OTUs.

167   Despite the drawbacks, closed reference methods have been used when fast execution

168   speed is required, such as when using very large datasets (14). To compare performance

169   in terms of speed, we repeated each OptiFit and OptiClust run 100 times and measured

170   the execution time. Across all dataset and database combinations, closed reference OptiFit

171   outperformed both OptiClust and open reference OptiFit (Figure 3). For example, with

172   the human dataset fit to SILVA reference OTUs, the average run times in seconds were

173   406.8 for closed reference OptiFit, 455.3 for *de novo* clustering the dataset, and 559.4 for

174   open reference OptiFit. Thus, the OptiFit algorithm continues the precedent that closed

175   reference clustering sacrifices OTU quality for execution speed.

176   To compare to the reference clustering methods used by QIIME2, we clustered each

177   dataset with VSEARCH against the Greengenes database of OTUs previously clustered

178   at 97% sequence similarity. Each reference OTU from the Greengenes 97% database

9

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 three clustering strategies; 1) *de novo* clustering the whole dataset using OptiClust, 2) splitting the dataset with 50% of the sequences as a reference set and the other 50% as a query set, clustering the references using OptiClust, then clustering the query sequences to the reference OTUs with OptiFit, or 3) clustering the dataset to a reference database. 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.

179 contains one reference sequence, and VSEARCH maps sequences to the reference

180 based on each individual query sequence's similarity to the single reference sequence.

181 In contrast, OptiFit accepts reference OTUs which each may contain multiple sequences,

182 and the sequence similarity between all query and reference sequences is considered

183 when assigning sequences to OTUs. In closed reference mode, OptiFit produced 27.2%

184 higher quality OTUs than VSEARCH in terms of MCC score, but VSEARCH was able

185 to cluster ~~24.8~~24.9% more query sequences than OptiFit to the Greengenes reference

186 database (Figure 3). This is because VSEARCH only considers the distances between

187 each query sequence to the single reference sequence, while OptiFit considers the

188 distances between all pairs of reference and query sequences in an OTU. When open

189 reference clustering, OptiFit produced higher quality OTUs than VSEARCH against the

190 Greengenes database, with median MCC scores of 0.86 and 0.56, respectively. In terms

191 of run time, OptiFit outperformed VSEARCH in both closed and open reference mode by

192 ~~54.6% and 49.5~~53.6% and 44.0% on average, respectively. Thus, the more stringent OTU

193 definition employed by OptiFit, which prefers the query sequence to be similar to all other

194 sequences in the OTU rather than to only one sequence, resulted in fewer sequences

195 being clustered to reference OTUs than when using VSEARCH, but caused OptiFit to

196 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) (Figure 3; "self-split"). OTU quality was similar to that from OptiClust regardless of mode (~~0.029~~0.031% difference in median MCC). In closed reference mode, OptiFit was able to cluster ~~84.8~~84.9% of query sequences to reference OTUs with the split strategy, a great improvement over the average ~~59.1~~59% 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 ~~34.7% and 33.5~~39.6% and 36.8%, respectively. Random access memory (RAM) usage was similar, with OptiFit requiring slightly more RAM in gigabytes than OptiClust. Open and closed reference OptiFit required 1.8% and 1.2% more RAM than OptiClust, respectively (data not shown). The split dataset strategy also performed ~~13.5~~6.7% faster than the database strategy in closed reference mode and ~~43.5~~65.5% 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

₂₂₃ fits far more query sequences than the database strategy.

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. Reference 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 with OptiClust is also shown for comparison.

₂₂₄ While we initially tested this strategy using a 50% split of the data into reference and

₂₂₅ query fractions, we next investigated whether there was an optimal reference fraction size.

₂₂₆ To identify the best reference size, reference sets with 10% to 90% of the sequences

₂₂₇ were created, with the remaining sequences used for the query (Figure 4). OTU quality

₂₂₈ was remarkably consistent across reference fraction sizes. For example, splitting the

₂₂₉ human dataset 100 times yielded a coefficient of variation (i.e. the standard deviation

₂₃₀ divided by the mean) of ~~0.00022~~ 0.0018 for the MCC score across all fractions. Run time

₂₃₁ generally decreased as the reference fraction increased; for the human dataset, the median

₂₃₂ run time was ~~364.1~~ 364.0 seconds with 10% of sequences in the reference and ~~291.3~~

₂₃₃ 290.8 seconds with 90% of sequences in the reference. The RAM usage was virtually

₂₃₄ the same across reference fraction sizes, with a coefficient of variation of 0.00089 for the

₂₃₅ human dataset (data not shown). In closed reference mode, the fraction of sequences that

₂₃₆ mapped increased as the reference size increased; for the human dataset, the median

₂₃₇ fraction mapped was 0.85 with 10% of sequences in the reference and 0.95 with 90% of

₂₃₈ sequences in the reference. These trends held for the other datasets as well. Thus, the

₂₃₉ reference fraction did not affect OTU quality in terms of MCC score nor the memory usage,

₂₄₀ but did affect the run time and the fraction of sequences that mapped during the closed

₂₄₁ reference clustering.

₂₄₂ After testing the split strategy using a simple random sample to select the reference

243 sequences, we then investigated other methods of splitting the data. We tested three

244 methods for selecting the fraction of sequences to be used as the reference at a size of

245 50%: a simple random sample, weighting sequences by relative abundance, and weighting

246 by similarity to other sequences in the dataset (Figure 4). OTU quality in terms of MCC was

247 similar across all three sampling methods (median MCC of ~~0.87~~0.86). In closed-reference

248 clustering mode, the fraction of sequences that mapped were similar for simple and

249 abundance-weighted sampling (median fraction mapped of 0.85 and 0.84, respectively),

250 but worse for similarity-weighted sampling (median fraction mapped of 0.56). While simple

251 and abundance-weighted sampling produced better quality OTUs than similarity-weighted

252 sampling, OptiFit performed faster on similarity-weighted samples with a median runtime of

253 ~~93.8~~ 103.9 seconds compared to ~~123.2 and 122.6~~ 135.4 and 134.8 seconds for simple and

254 abundance-weighted sampling, respectively. Thus, employing more complicated sampling

255 strategies such as abundance-weighted and similarity-weighted sampling did not confer

256 any advantages over selecting the reference via a simple random sample, and in fact

257 decreased OTU quality in the case of similarity-weighted sampling.

Discussion

259 We developed a new algorithm for clustering sequences to existing OTUs and have

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

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

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

263 reference mode, regardless of strategy employed. Open reference OptiFit performs slower

264 than OptiClust due to the additional *de novo* clustering step, so users may prefer OptiClust

265 for tasks that do not require reference OTUs.

266 When clustering to public databases, OTU quality dropped in closed reference mode to

267 different degrees depending on the database and dataset source, and no more than half

₂₆₈ of query sequences were able to be clustered into OTUs across any dataset/database

₂₆₉ combination. This may reflect limitations of reference databases, which are unlikely

₂₇₀ to contain sequences from novel microbes. This drop in quality was most notable

₂₇₁ with the RDP reference, which contained only 16,192 sequences compared to 173,648

₂₇₂ sequences in SILVA and 174,979 in Greengenes. Note that Greengenes has not been

₂₇₃ updated since 2013 at the time of this writing, while SILVA and the RDP are updated

₂₇₄ regularly. We recommend that users who require an independent reference database

₂₇₅ opt for large databases with regular updates and good coverage of microbial diversity for

₂₇₆ their environment. Since OptiClust still performs faster than open reference OptiFit and

₂₇₇ creates higher quality OTUs than closed reference OptiFit with the database strategy, we

₂₇₈ recommend using OptiClust rather than clustering to a database whenever consistent

₂₇₉ OTUs are not required.

₂₈₀ The OptiClust and OptiFit algorithms produced higher quality OTUs than VSEARCH in

₂₈₁ open reference, closed reference, or *de novo* modes. However, VSEARCH was able

₂₈₂ to cluster more sequences to OTUs than OptiFit in closed reference mode. While both

₂₈₃ OptiFit and VSEARCH use a distance or similarity threshold for determining how to cluster

₂₈₄ sequences into OTUs, VSEARCH is more permissive than OptiFit regardless of mode.

₂₈₅ The OptiFit and OptiClust algorithms use all of the sequences to define an OTU, preferring

₂₈₆ that all pairs of sequences (including reference and query sequences) in an OTU are within

₂₈₇ the distance threshold in order to maximize the MCC. In contrast, VSEARCH only requires

₂₈₈ each query sequence to be similar to the single centroid sequence that seeded the OTU,

₂₈₉ thus allowing pairs of query sequences to be less similar to each other than the threshold

₂₉₀ specified. Because of this, VSEARCH sacrifices OTU quality by allowing more dissimilar

₂₉₁ sequences to be clustered into the same OTUs.

₂₉₂ When clustering with the split dataset strategy, OTU quality was remarkably similar when

₂₉₃ reference sequences were selected by a simple random sample or weighted by abundance,

14

²⁹⁴ but quality was slightly worse when sequences were weighted by similarity. We recommend

²⁹⁵ using a simple random sample since the more sophisticated reference selection methods

²⁹⁶ do not offer any benefit. The similarity in OTU quality between OptiClust and OptiFit with

²⁹⁷ this strategy demonstrates the suitability of using OptiFit to cluster sequences to existing

²⁹⁸ OTUs, such as when comparing OTUs across studies. However, when consistent OTUs

²⁹⁹ are not required, we recommend using OptiClust for *de novo* clustering over the split

³⁰⁰ strategy with OptiFit since OptiClust is simpler to execute but performs similarly in terms of

³⁰¹ both run time and OTU quality.

³⁰² Unlike existing reference-based methods that cluster query sequences to a single centroid

³⁰³ sequence in each reference OTU, OptiFit considers all sequences in each reference OTU

³⁰⁴ when clustering query sequences, resulting in OTUs of a similar high quality as those

³⁰⁵ produced by the *de novo* OptiClust algorithm. Potential applications include clustering

³⁰⁶ sequences to reference databases, comparing taxonomic composition of microbiomes

³⁰⁷ across different studies, or using OTU-based machine learning models to make predictions

³⁰⁸ on new data. OptiFit fills the missing option for clustering query sequences to existing

³⁰⁹ OTUs that does not sacrifice OTU quality for consistency of OTU assignments.

Materials and Methods

Data Processing Steps

³¹² We downloaded 16S rRNA gene amplicon sequences from four published datasets isolated

³¹³ from soil (8), marine (9), mouse gut (10), and human gut (11) samples. These datasets

³¹⁴ contain sequences from the V4 region of the 16S rRNA gene and represent a selection

³¹⁵ of the broad types of natural communities that microbial ecologists study. We processed

³¹⁶ the raw sequences using mothur according to the Schloss Lab MiSeq SOP (15) and

³¹⁷ accompanying study by Kozich *et al.* (16). These steps included trimming and filtering

³¹⁸ for quality, aligning to the SILVA reference alignment (12), discarding sequences that

₃₁₉ aligned outside the V4 region, removing chimeric reads with UCHIME (17), and calculating

₃₂₀ distances between all pairs of sequences within each dataset prior to clustering.

Reference database clustering

₃₂₂ To generate reference OTUs from public databases, we downloaded sequences from the

₃₂₃ Greengenes database (v13_8_99) (6), SILVA non-redundant database (v132) (12), and the

₃₂₄ Ribosomal Database Project (v16) (13). These sequences were processed using the same

₃₂₅ steps outlined above followed by clustering sequences into *de novo* OTUs with OptiClust.

₃₂₆ Processed reads from each of the four datasets were clustered with OptiFit to the reference

₃₂₇ OTUs generated from each of the three databases. When reference clustering with

₃₂₈ VSEARCH, processed datasets were clustered directly to the unprocessed Greengenes

₃₂₉ 97% OTU reference alignment, since this method is how VSEARCH is typically used by

₃₃₀ the QIIME2 software for reference-based clustering (7, 18).

Split dataset clustering

₃₃₂ For each dataset, half of the sequences were selected to be clustered *de novo* into

₃₃₃ reference OTUs with OptiClust. We used three methods for selecting the subset of

₃₃₄ sequences to be used as the reference: a simple random sample, weighting sequences by

₃₃₅ relative abundance, and weighting by similarity to other sequences in the dataset. Dataset

₃₃₆ splitting was repeated with 100 random seeds. With the simple random sampling method,

₃₃₇ dataset splitting was also repeated with reference fractions ranging from 10% to 90% of

₃₃₈ the dataset. For each dataset split, the remaining query sequences were clustered into the

₃₃₉ reference OTUs with OptiFit.

Benchmarking

₃₄₁ OptiClust and OptiFit randomize the order of query sequences prior to clustering and

₃₄₂ employ a random number generator to break ties when OTU assignments are of equal

16

343 quality. As a result, they produce slightly different OTU assignments when repeated

344 with different random seeds. To capture any variation in OTU quality or execution time,

345 clustering was repeated with 100 random seeds for each combination of parameters and

346 input datasets. We used the benchmark feature provided by Snakemake to measure the

347 run time of every clustering job. We calculated the MCC on each set of OTUs to quantify

348 the quality of clustering, as described by Westcott *et al.* (1).

Data and Code Availability

350 We implemented the analysis workflow in Snakemake (19) and wrote scripts in R (20),

351 Python (21), and GNU bash (22). Software used includes mothur v1.47.0 (23), VSEARCH

352 v2.15.2 (5), the tidyverse metapackage (24), R Markdown (25), ggraph (26), ggtext (27),

353 numpy (28), the SRA toolkit (29), and conda (30). The complete workflow and supporting

354 files required to reproduce this manuscript are available at https://github.com/SchlossLab/

355 Sovacool_OptiFit_mSphere_2022.

Acknowledgements

Author Contributions

363 KLS wrote the analysis code, evaluated the algorithm, and wrote the original draft of

364 the manuscript. SLW designed and implemented the OptiFit algorithm and assisted in

365 debugging the analysis code. MBM and GAD contributed analysis code. PDS conceived

366 the study, supervised the project, and assisted in debugging the analysis code. All authors

367 reviewed and edited the manuscript.

References

369 1.   **Westcott SL**, **Schloss PD**. 2017. OptiClust, an Improved Method for Assigning
Amplicon-Based Sequence Data to Operational Taxonomic Units.   mSphere
370   **2**:e00073–17. doi:10.1128/mSphereDirect.00073-17.

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

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

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

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

379 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.
380   doi:10.1128/AEM.03006-05.

381 7.   Clustering sequences into OTUs using Q2-vsearch — QIIME 2 2021.2.0
382   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.

395 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

396 QIIME, p. 371–444. *In* DeLong, EF (ed.), Methods in Enzymology. Academic Press.

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

398

399 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.

400 Appl Environ Microbiol **79**:5112–5120. doi:10.1128/AEM.01043-13.

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

402 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.

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

412

413 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.
414   doi:10.1128/AEM.01541-09.

415 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
416   of Open Source Software **4**:1686. doi:10.21105/joss.01686.

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

419 26.   **Pedersen TL**. 2021. Ggraph: An implementation of grammar of graphics for graphs
420   and networks.

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

422

423 28.   **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.
424   doi:10.1038/s41586-020-2649-2.

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

426

427 30.   2016. Anaconda Software Distribution. Anaconda Documentation. Anaconda Inc.

428