

# **OptiClust: Improved method for assigning amplicon-based sequence data to operational taxonomic units**

Running title: OptiClust: Optimized Clustering

Sarah L. Westcott and Patrick D. Schloss<sup>†</sup>

<sup>†</sup> To whom correspondence should be addressed: [pschloss@umich.edu](mailto:pschloss@umich.edu)

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

## 1 **Abstract**

2 Assignment of 16S rRNA gene sequences to operational taxonomic units (OTUs) is a computational  
3 bottleneck in the process of analyzing microbial communities. Although this has been an active  
4 area of research, it has been difficult to overcome the time and memory demands while improving  
5 the quality of the OTU assignments. Here we developed a new OTU assignment algorithm that  
6 iteratively reassigns sequences to new OTUs to optimize the Matthews correlation coefficient  
7 (MCC), a measure of the quality of OTU assignments. To assess the new algorithm, OptiClust,  
8 we compared it to ten other algorithms using 16S rRNA gene sequences from two simulated and  
9 four natural communities. Using the OptiClust algorithm, the MCC values averaged 15.2 and  
10 16.5% higher than the OTUs generated when we used the average neighbor and distance-based  
11 greedy clustering with VSEARCH, respectively. Furthermore, on average, OptiClust was 94.6-times  
12 faster than the average neighbor algorithm and just as fast as distance-based greedy clustering  
13 with VSEARCH. An empirical analysis of the efficiency of the algorithms showed that the time  
14 and memory required to perform the algorithm scaled quadratically with the number of unique  
15 sequences in the dataset. The significant improvement in the quality of the OTU assignments over  
16 previously existing methods will significantly enhance downstream analysis by limiting the splitting  
17 of similar sequences into separate OTUs and merging of dissimilar sequences into the same OTU.  
18 The development of the OptiClust algorithm represents a significant advance that is likely to have  
19 numerous other applications.

## 20 **Importance**

21 The analysis of microbial communities from diverse environments using 16S rRNA gene sequencing  
22 has expanded our knowledge of the biogeography of microorganisms. An important step in  
23 this analysis is the assignment of sequences into taxonomic groups based on their similarity to  
24 sequences in a database or based on their similarity to each other, irrespective of a database. In  
25 this study, we present a new algorithm for the latter approach. The algorithm, OptiClust, seeks  
26 to optimize a metric of assignment quality by shuffling sequences between taxonomic groups.

27 We found that OptiClust produces more robust assignments and does so in a rapid and memory  
28 efficient manner. This advance will allow for a more robust analysis of microbial communities and  
29 the factors that shape them.

## 30 Introduction

31 Amplicon-based sequencing has provided incredible insights into Earth's microbial biodiversity (1,  
32 2). It has become common for studies to include sequencing millions of 16S rRNA gene sequences  
33 across hundreds of samples (3, 4). This is three to four orders of magnitude greater sequencing  
34 depth than was previously achieved using Sanger sequencing (5, 6). The increased sequencing  
35 depth has revealed novel taxonomic diversity that is not adequately represented in reference  
36 databases (1, 3). However, the advance has forced re-engineering of methods to overcome the  
37 rate and memory limiting steps in computational pipelines that process raw sequences through  
38 the generation of tables containing the number of sequences in different taxa for each sample  
39 (7–10). A critical component to these pipelines has been the assignment of amplicon sequences to  
40 taxonomic units that are either defined based on similarity to a reference or operationally based on  
41 the similarity of the sequences to each other within the dataset (11, 12).

42 A growing number of algorithms have been developed to cluster sequences into OTUs. These  
43 algorithms can be classified into three general categories. The first category of algorithms has been  
44 termed closed-reference or phylotyping (13, 14). Sequences are compared to a reference collection  
45 and clustered based on the reference sequences that they are similar to. This approach is fast;  
46 however, the method struggles when a sequence is similar to multiple reference sequences that  
47 may have different taxonomies and when it is not similar to sequences in the reference. The second  
48 category of algorithms has been called *de novo* because they assign sequences to OTUs without  
49 the use of a reference (14). These include hierarchical algorithms such as nearest, furthest, and  
50 average neighbor (15) and algorithms that employ heuristics such as abundance or distance-based  
51 greedy clustering as implemented in USEARCH (16) or VSEARCH (17), Sumacust, OTUCLUST  
52 (18), and Swarm (19). *De novo* methods tend to be more computationally intense and it has proven  
53 difficult to know which method generates the best assignments. A third category of algorithm  
54 is open-reference clustering, which is a hybrid approach (3, 14). Here sequences are assigned  
55 to OTUs using closed-reference clustering and sequences that are not within a threshold of a  
56 reference sequence are then clustered using a *de novo* approach. This category blends the  
57 strengths and weaknesses of the other method and adds the complication that closed-reference

58 and *de novo* clustering use different OTU definitions. These algorithms take different approaches to  
59 handling large datasets to minimize the time and memory requirements while attempting to assign  
60 sequences to meaningful OTUs.

61 Several metrics have emerged for assessing the quality of OTU assignment algorithms. These have  
62 included the time and memory required to run the algorithm (3, 19–21), agreement between OTU  
63 assignments and the sequences' taxonomy (19, 21–31), sensitivity of an algorithm to stochastic  
64 processes (32), the number of OTUs generated by the algorithm (22, 33), and the ability to  
65 regenerate the assignments made by other algorithms (3, 34). Unfortunately, these methods fail to  
66 directly quantify the quality of the OTU assignments. An algorithm may complete with minimal time  
67 and memory requirements or generate an idealized number of OTUs, but the composition of the  
68 OTUs could be incorrect. These metrics also tend to be subjective. For instance, a method may  
69 appear to be recapitulate the taxonomy of a synthetic community with known taxonomic structure,  
70 but do a poor job when applied to real communities with poorly defined taxonomic structure or for  
71 sequences that are prone to misclassification. As an alternative, we developed an approach to  
72 objectively benchmark the clustering quality of OTU assignments (13, 35, 36). This approach counts  
73 the number of true positives (TP), true negatives (TN), false positives (FP), and false negatives  
74 (FN) based on the pairwise distances. Sequence pairs that are within the user-specified threshold  
75 and are clustered together represent TPs and those in different OTUs are FNs. Those sequence  
76 pairs that have a distance larger than the threshold and are not clustered in the same OTU are TNs  
77 and those in the same OTU are FPs. These values can be synthesized into a single correlation  
78 coefficient, the Matthews' Correlation Coefficient (MCC), which measures the correlation between  
79 observed and predicted classifications and is robust to cases where there is an uneven distribution  
80 across the confusion matrix (37). Consistently, the average neighbor algorithm was identified as  
81 among the best or the best algorithm. The distance-based greedy clustering as implemented in  
82 VSEARCH has also performed well. The computational resources required to complete the average  
83 neighbor algorithm can be significant for large datasets and so there is a need for an algorithm that  
84 efficiently produces consistently high quality OTU assignments.

85 These previous efforts have assessed the quality of the clusters after the completion of the algorithm.  
86 In the current study we developed and benchmarked a new *de novo* clustering algorithm that uses

87 real time calculation of the MCC to direct the progress of the clustering. The result is the OptiClust  
88 algorithm, which produces significantly better sequence assignments while making efficient use of  
89 computational resources.

## 90 **Results**

91 ***OptiClust algorithm.*** The OptiClust algorithm uses the pairs of sequences that are within a desired  
92 threshold of each other (e.g. 0.03), a list of all sequence names in the dataset, and the metric that  
93 should be used to assess clustering quality. A detailed description of the algorithm is provided for a  
94 toy dataset in the Supplementary Material. Briefly, the algorithm starts by placing each sequence  
95 either within its own OTU or into a single OTU. The algorithm proceeds by interrogating each  
96 sequence and re-calculating the metric for the cases where the sequence stays in its current OTU,  
97 is moved to each of the other OTUs, or is moved into a new OTU. The location that results in  
98 the best clustering quality indicates whether the sequence should remain in its current OTU or  
99 be moved to a different or new OTU. Each iteration consists of interrogating every sequence in  
100 the dataset. Although numerous options are available within the mothur-based implementation  
101 of the algorithm (e.g. sensitivity, specificity, accuracy, F1 score, etc.), the default metric is MCC  
102 because it includes all four parameters from the confusion matrix. The algorithm continues until the  
103 optimization metric stabilizes or until it reaches a defined stopping criteria.

104 ***OptiClust-generated OTUs are more robust than those from other methods.*** To evaluate the  
105 OptiClust algorithm and compare its performance to other algorithms, we utilized six datasets  
106 including two synthetic communities and four previously published large datasets generated from  
107 soil, marine, human, and murine samples (Table 1). When we seeded the OptiClust algorithm with  
108 each sequence in a separate OTU and ran the algorithm until complete convergence, the MCC  
109 values averaged 15.2 and 16.5% higher than the OTUs using average neighbor and distance-based  
110 greedy clustering (DGC) with VSEARCH, respectively (Figure 1). The number of OTUs formed by  
111 the various methods was negatively correlated with their MCC value ( $\rho=-0.47$ ;  $p=0$ ). The OptiClust  
112 algorithm was considerably faster than the hierarchical algorithms and somewhat slower than  
113 the heuristic-based algorithms. Across the six datasets, the OptiClust algorithm was 94.6-times

114 faster than average neighbor and just as fast as DGC with VSEARCH. The human dataset was a  
115 challenge for a number of the algorithms. OTUCLUST and SumaClust were unable to cluster the  
116 human dataset in less than 50 hours and the average neighbor algorithm required more than 45 GB  
117 of RAM. The USEARCH-based methods were unable to cluster the human data using the 32-bit  
118 free version of the software that limits the amount of RAM to approximately 3.5 GB. These data  
119 demonstrate that OptiClust generated significantly more robust OTU assignments than existing  
120 methods across a diverse collection of datasets with performance that was comparable to popular  
121 methods.

122 ***OptiClust stopping criteria.*** By default, the mothur-based implementation of the algorithm stops  
123 when the optimization metric changes by less than 0.0001; however, this can be altered by the user.  
124 This implementation also allows the user to stop the algorithm if a maximum number of iterations is  
125 exceeded. By default mothur uses a maximum value of 100 iterations. The justification for allowing  
126 incomplete convergence was based on the observation that numerous iterations are performed  
127 that extend the time required to complete the clustering with minimal improvement in clustering.  
128 We evaluated the results of clustering to partial convergence (i.e. a change in the MCC value that  
129 was less than 0.0001) or until complete convergence of the MCC value (i.e. until it did not change  
130 between iterations) when seeding the algorithm with each sequence in a separate OTU (Figure 1).  
131 The small difference in MCC values between the output from partial and complete convergence  
132 resulted in a difference in the median number of OTUs that ranged between 1.5 and 17.0 OTUs.  
133 This represented a difference of less than 0.15%. Among the four natural datasets, between 3 and  
134 6 were needed to achieve partial convergence and between 8 and 12.50 iterations were needed to  
135 reach full convergence. The additional steps required between 1.4 and 1.7 times longer to complete  
136 the algorithm. These results suggest that achieving full convergence of the optimization metric  
137 adds computational effort; however, considering full convergence took between 2 and 17 minutes  
138 the extra effort was relatively small. Although the mothur's default setting is partial convergence,  
139 the remainder of our analysis used complete convergence to be more conservative.

140 ***Effect of seeding OTUs on OptiClust performance.*** By default the mothur implementation of  
141 the OptiClust algorithm starts with each sequence in a separate OTU. An alternative approach  
142 is to start with all of the sequences in a single OTU. We found that the MCC values for clusters

143 generated seeding OptiClust with the sequences as a single OTU were between 0 and 11.5% lower  
144 than when seeding the algorithm with sequences in separate OTUs (Figure 1). Interestingly, with  
145 the exception of the human dataset (0.2% more OTUs), the number of OTUs was as much as 7.0%  
146 lower (mice) than when the algorithm was seeded with sequence in separate OTUs. Finally, the  
147 amount of time required to cluster the data when the algorithm was seeded with a single OTU was  
148 between 1.5 and 2.9-times longer than if sequences were seeded as separate OTUs. This analysis  
149 demonstrates that seeding the algorithm with sequences as separate OTUs resulted in the best  
150 OTU assignments in the shortest amount of time.

151 ***OptiClust-generated OTUs are as stable as those from other algorithms.*** One concern that  
152 many have with *de novo* clustering algorithms is that their output is sensitive to the initial order of  
153 the sequences. An additional concern with the OptiClust algorithm is that it may stabilize at a local  
154 optimum. To evaluate these concerns we compared the results obtained using ten randomizations  
155 of the order that sequences were given to the algorithm. The median the coefficient of variation  
156 across the six datasets for MCC values obtained from the replicate clusterings using OptiClust was  
157 0.1% (Figure 1). We also measured the coefficient of variation for the number of OTUs across  
158 the six datasets for each method. The median coefficient of variation for the number of OTUs  
159 generated using OptiClust was 0.1%. Confirming our previous results, all of the methods we tested  
160 were stable to stochastic processes. Of the methods that involved randomization, the coefficient  
161 of variation for MCC values considerably smaller with OptiClust than the other methods and the  
162 coefficient of variation for the number of OTUs was comparable to the other methods. The variation  
163 observed in clustering quality suggested that the algorithm does not appear to converge to a locally  
164 optimum MCC value. More importantly, the random variation does yield output of a similarly high  
165 quality.

166 ***Time and memory required to complete Optimization-based clustering scales efficiently.***  
167 Although not as important as the quality of clustering, the amount of time and memory required  
168 to assign sequences to OTUs is a legitimate concern. To evaluate how the speed and memory  
169 usage scaled with the number of sequences in the dataset, we measured the time required and  
170 maximum RAM usage to cluster 20, 40, 60, 80, and 100% of the unique sequences from each of  
171 the natural datasets using the OptiClust algorithm (Figure 2). Within each iteration of the algorithm,

172 each sequence is compared to every other sequence and each comparison requires a recalculation  
173 of the confusion matrix. This would result in a worst case algorithmic complexity on the order of  
174  $N^3$ , where  $N$  is the number of unique sequences. Because the algorithm only needs to keep track  
175 of the sequence pairs that are within the threshold of each other, it is likely that the implementation  
176 of the algorithm is more efficient. To empirically determine the algorithmic complexity, we fit a power  
177 law function to the data in Figure 2A. We observed power coefficients between 1.7 and 2.5 for the  
178 marine and human datasets, respectively. The algorithm requires storing a matrix that contains the  
179 pairs of sequences that are close to each other as well as a matrix that indicates which sequences  
180 are clustered together. The memory required to store these matrices is on the order of  $N^2$ , where  
181  $N$  is the number of unique sequences. In fact, when we fit a power law function to the data in  
182 Figure 2B, the power coefficients were 1.9. This analysis suggests that doubling the number of  
183 sequences in a dataset would increase the time required to cluster the data by 4 to 8-fold and  
184 increase the RAM required by 4-fold. It is possible that future improvements to the implementation  
185 of the algorithm could improve this performance.

186 ***Cluster splitting heuristic generates OTUs that are as good as non-split approach.*** We  
187 previously described a heuristic to accelerate OTU assignments where sequences were first  
188 classified to taxonomic groups and within each taxon sequences were assigned to OTUs using  
189 the average neighbor clustering algorithm (13). This accelerated the clustering and reduce the  
190 memory requirements because the number of unique sequences is effectively reduced by splitting  
191 sequences across taxonomic groups. Furthermore, because sequences in different taxonomic  
192 groups are assumed to belong to different OTUs they are independent, which permits parallelization  
193 and additional reduction in computation time. Reduction in clustering quality are encountered in this  
194 approach if there are errors in classification or if two sequences within the desired threshold belong  
195 to different taxonomic groups. It is expected that these errors would increase as the taxonomic level  
196 goes from kingdom to genus. To characterize the clustering quality, we calculated the MCC values  
197 using OptiClust, average neighbor, and DGC with VSEARCH when splitting at each taxonomic level  
198 (Figure 3). For each method, the MCC values decreased as the taxonomic resolution increased;  
199 however, the decrease in MCC as not as large as the difference between clustering methods. As  
200 the resolution of the taxonomic levels increased, the clustering quality remained high, relative to

201 clusters formed from the entire dataset (i.e. kingdom-level). The MCC values when splitting the  
202 datasets at the class and genus levels were within 98.0 and 93.0%, respectively, of the MCC values  
203 obtained from the entire dataset. These decreases in MCC value resulted in the formation of as  
204 many as 4.7 and 22.5% more OTUs, respectively, than were observed from the entire dataset. For  
205 the datasets included in the current analysis, the use of the cluster splitting heuristic was probably  
206 not worth the loss in clustering quality. However, as datasets become larger, it may be necessary to  
207 use the heuristic to clustering the data into OTUs.

## 208 **Discussion**

209 Myriad methods have been proposed for assigning 16S rRNA gene sequences to OTUs that  
210 each claim improved performance based on speed, memory usage, representation of taxonomic  
211 information, and number of OTUs. Each of these metrics is subjective and do not actually indicate  
212 the quality of the clustering. This led us to propose using the MCC as a metric for assessing  
213 the quality of clustering, post hoc. Here, we described a new clustering method that seeks to  
214 optimize clustering based on an objective criterion that measures clustering quality in real time.  
215 In the OptiClust algorithm clustering is driven by optimizing a metric that assesses whether any  
216 two sequences should be grouped into the same OTU. The result is clusters that are significantly  
217 more robust and is efficient in the time and memory required to cluster the sequences into OTUs.  
218 This makes it more tractable to analyze large datasets without sacrificing clustering quality as was  
219 previously necessary using heuristic methods.

220 The cluster optimization procedure is dependent on the metric that is chosen for optimization. We  
221 employed the MCC because it includes the four values from a confusion matrix. Other algorithms  
222 such as the furthest neighbor and nearest neighbor algorithms minimize the number of FP and  
223 FN, respectively; however, these suffer because the number of FN and FP are not controlled (13,  
224 15). Alternatively, one could optimize based on the sensitivity, specificity, or accuracy, which are  
225 each based on two values from the confusion matrix or they could optimize based on the F1 score,  
226 which is based on three values from the confusion matrix. Because these metrics do not balance all  
227 four parameters equally, it is likely that one parameter will dominate in the optimization procedure.

228 For example, optimizing for sensitivity could lead to a large number of FPs. Since we would like  
229 to minimize both FPs and FNs and not just the total number of false assignments, we decided to  
230 optimize utilizing the MCC. It is possible that other metrics could be developed and employed for  
231 optimization of the clustering.

232 The OptiClust algorithm is relatively simple. For each sequence it effectively ask whether the MCC  
233 value will increase if the sequence is moved to a different OTU including creating a new OTU. If the  
234 value does not change, it remains in the current OTU. The algorithm repeats until the MCC value  
235 stabilizes. Assuming that the algorithm is seeded with each sequence in a separate OTU, it does  
236 not appear that the algorithm converges to a local optimum. Furthermore, execution of the algorithm  
237 with different random number generator seeds produces OTU assignments of consistently high  
238 quality. Future improvements to the implementation of the algorithm could provide optimization to  
239 further improve its speed and susceptibility to find a local optimum. Users are encourage to repeat  
240 the OTU assignment several times to confirm that they have found the best OTU assignments.

241 Our previous MCC-based analysis of clustering algorithms indicated that the average neighbor  
242 algorithm consistently produced the best OTU assignments with the DGC-based method using  
243 USEARCH also producing robust OTU assignments. The challenge in using the average neighbor  
244 algorithm is that it requires a large amount of RAM and is computationally demanding. This led to  
245 the development of a splitting approach that divides the clustering across distinct taxonomic groups  
246 (13). The improved performance provided by the OptiClust algorithm likely makes such splitting  
247 unnecessary for most current datasets. We have demonstrated that although the OTU assignments  
248 made at the genus level are still better than that of other methods, the quality is not as good as that  
249 found without splitting. The loss of quality is likely due to misclassification because of limitations  
250 in the clustering algorithms and reference databases. The practical significance of such small  
251 differences in clustering quality remain to be determined; however, based on the current analysis, it  
252 does appear that the number of OTUs is artificially inflated. Regardless, the best clustering quality  
253 should be pursued given the available computer resources.

254 The time and memory required to execute the OptiClust algorithm scaled proportionally to the  
255 number of unique sequences raised to the second power. The power for the time requirement is

256 affected by the similarity of the sequences in the dataset with datasets containing more similar  
257 sequences having a higher power. Also, the number of unique sequences is the basis for both the  
258 amount of time and memory required to complete the algorithm. Both the similarity of sequences  
259 and number of unique sequences can be driven by the sequencing error since any errors will  
260 increase the number of unique sequences and these sequences will be closely related to the  
261 perfect sequence. This underscores the importance of reducing the noise in the sequence data (7).  
262 If sequencing errors are not remediated and are relatively randomly distributed, then it is likely that  
263 the algorithm will require an unnecessary amount of time and RAM to complete.

264 The rapid expansion in sequencing capacity has demanded that the algorithms used to assign  
265 16S rRNA gene sequences to OTUs be efficient while maintaining robust assignments. Although  
266 database-based approaches have been proposed to facilitate this analysis, they are limited by  
267 their limited coverage of bacterial taxonomy and by the inconsistent process used to name taxa.  
268 The ability to assign sequences to OTUs using an algorithm that optimizes clustering by directly  
269 measuring quality will significantly enhance downstream analysis. The development of the OptiClust  
270 algorithm represents a significant advance that is likely to have numerous other applications.

## 271 **Materials and Methods**

272 ***Sequence data and processing steps.*** To evaluate the OptiClust and the other algorithms we  
273 created two synthetic sequence collections and four sequence collections generated from previously  
274 published studies. The V4 region of the 16S rRNA gene was used from all datasets because it  
275 is a popular region that can be fully sequenced with two-fold coverage using the commonly used  
276 MiSeq sequencer from Illumina (7). The method for generating the simulated datasets followed  
277 the approach used by Kopylova et al. (33) and Schloss (35). Briefly, we randomly selected  
278 10,000 unique V4 fragments from 16S rRNA gene sequences that were unique from the SILVA  
279 non-redundant database (38). A community with an even relative abundance profile was generated  
280 by specifying that each sequence had a frequency of 100 reads. A community with a staggered  
281 relative abundance profile was generated by specifying that the abundance of each sequence was  
282 a randomly drawn integer sampled from a uniform distribution between 1 and 200. Sequence

283 collections collected from human feces (39), murine feces (40), soil (41), and seawater (42) were  
284 used to characterize the algorithms' performance with natural communities. These sequence  
285 collections were all generated using paired 150 or 250 nt reads of the V4 region. We re-processed  
286 all of the reads using a common analysis pipeline that included quality score-based error correction  
287 (7), alignment against a SILVA reference database (38, 43), screening for chimeras using UCHIME  
288 (9), and classification using a naive Bayesian classifier with the RDP training set requiring an 80%  
289 confidence score (10).

290 **Implementation of clustering algorithms.** In addition to the OptiClust algorithm we evaluated  
291 ten different *de novo* clustering algorithms. These included three hierarchical algorithms, average  
292 neighbor, nearest neighbor, and furthest neighbor, which are implemented in mothur (v.1.39.0) (11).  
293 Seven heuristic methods were also used including abundance-based greedy clustering (AGC) and  
294 (distance-based greedy clustering) DGC as implemented in USEARCH (v.6.1) (16) and VSEARCH  
295 (v.2.3.3) ((17)], OTUCLUST (v.0.1) (18), SumaClust (v.1.0.20), and Swarm (v.2.1.9) (19). With  
296 the exception of Swarm each of these methods uses distance-based thresholds to report OTU  
297 assignments.

298 **Benchmarking.** We evaluated the quality of the sequence clustering, reproducibility of the  
299 clustering, the speed of clustering, and the amount of memory required to complete the clustering.  
300 To assess the quality of the clusters generated by each method, we counted the cells within a  
301 confusion matrix that indicated how well the clusterings represented the distances between the pair  
302 of sequences (13). Pairs of sequences that were in the same OTU and had a distance less than  
303 3% were true positives (TPs), those that were in different OTUs and had a distance greater than  
304 3% were true negatives (TNs), those that were in the same OTU and had a distance greater than  
305 3% were false positives (FPs), and those that were in different OTUs and had a distance less than  
306 3% were false negatives (FNs). To synthesize the matrix into a single metric we used the Matthews  
307 Correlation Coefficient using the `sens.spec` command in mothur using the following equations.

$$MCC = \frac{TP \times TN - FP \times FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}}$$

308 To assess the reproducibility of the algorithms we randomized the starting order of each sequence  
309 collection ten times and ran each algorithm on each randomized collection. We then measured the  
310 MCC for each randomization and quantified their percent coefficient of variation (% CV; 100 times  
311 the ratio of the standard deviation to the mean).

312 To assess how the the memory and time requirements scaled with the number of sequences  
313 included in each sequence collection, we randomly subsampled 20, 40, 60, or 80% of the unique  
314 sequences in each collection. We obtained 10 subsamples at each depth for each dataset and ran  
315 each collection (N= 50 = 5 sequencing depths x 10 replicates) through each of the algorithms. We  
316 used the timeout script to quantify the maximum RAM used and the amount of time required to  
317 process each sequence collection (<https://github.com/pshved/timeout>). We limited each algorithm  
318 to 45 GB of RAM and 50 hours using a single processor.

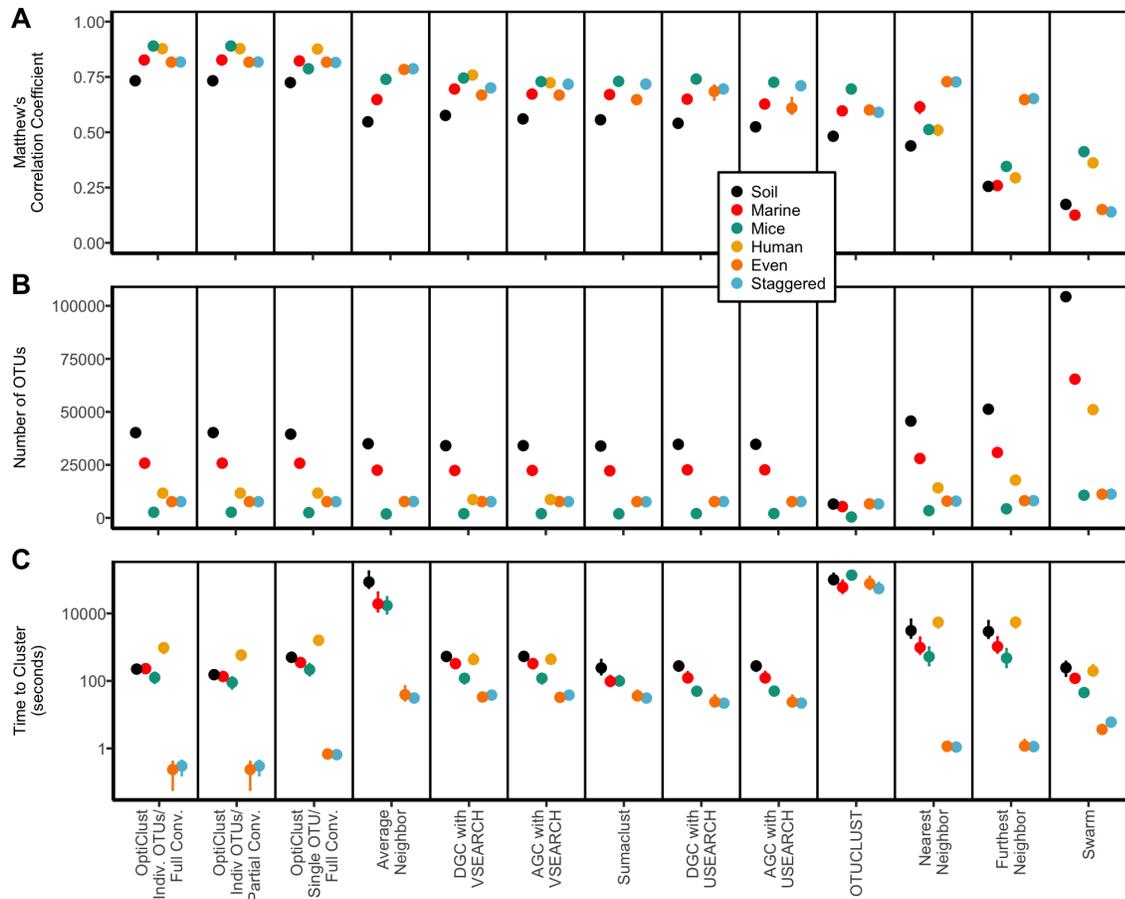
319 **Data and code availability.** The workflow utilized commands in GNU make (v.3.81), GNU bash  
320 (v.4.1.2), mothur (v.1.39.0) (11), and R (v.3.3.2) (44). Within R we utilized the wesanderson (v.0.3.2)  
321 (45), dplyr (v.0.5.0) (46), tidyr (v.0.6.0) (47), cowplot (v.0.6.9990) (48), and ggplot2 (v.2.1.0.9001)  
322 (49) packages. A reproducible version of this manuscript and analysis is available at [https://github.](https://github.com/SchlossLab/Westcott_OptiClust_mSphere_2017)  
323 [com/SchlossLab/Westcott\\_OptiClust\\_mSphere\\_2017](https://github.com/SchlossLab/Westcott_OptiClust_mSphere_2017).

## 324 **Acknowledgements**

325 This work was supported through funding from the National Institutes of Health to PDS  
326 (P30DK034933). SLW designed, implemented, and evaluated the algorithm. PDS designed and  
327 evaluated the algorithm. Both authors wrote and edited the manuscript.

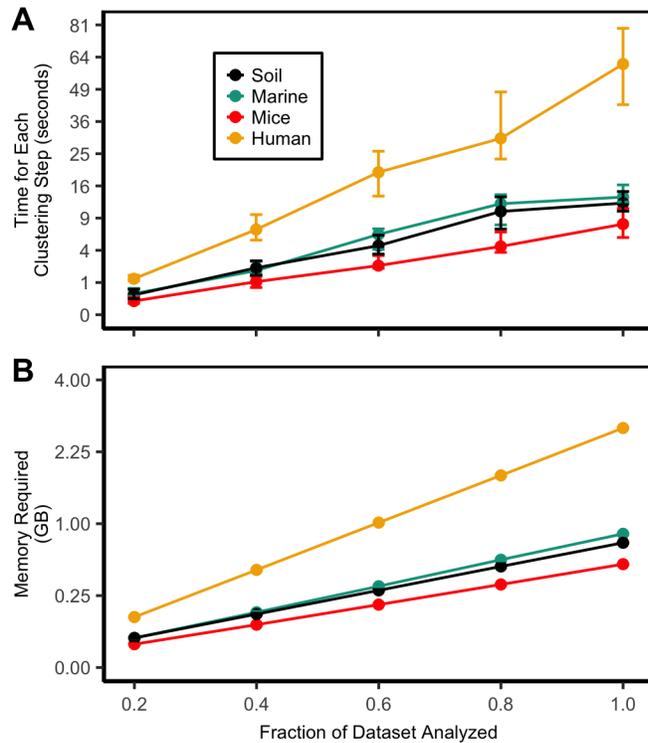
328 **Table 1. Description of datasets used to evaluate the OptiClust algorithm and compare its**  
329 **performance to other algorithms.** Each dataset contains sequences from the V4 region of the  
330 16S rRNA gene. The even and staggered datasets were generated by extracting the V4 region from  
331 full length reference sequences and the datasets from the natural communities were generated by  
332 sequencing the V4 region using a Illumina MiSeq with either paired 150 or 250 nt reads.

<b>Dataset (Ref.)</b>	<b>Read Length</b>	<b>Samples</b>	<b>Total Seqs.</b>	<b>Unique Seqs.</b>
Soil (41)	150	18	948,243	143,677
Marine (42)	250	7	1,384,988	75,923
Mice (40)	250	360	2,825,495	32,447
Human (39)	250	489	20,951,841	121,281
Even (33, 35)	NA	NA	1,155,800	11,558
Staggered (33, 35)	NA	NA	1,156,550	11,558



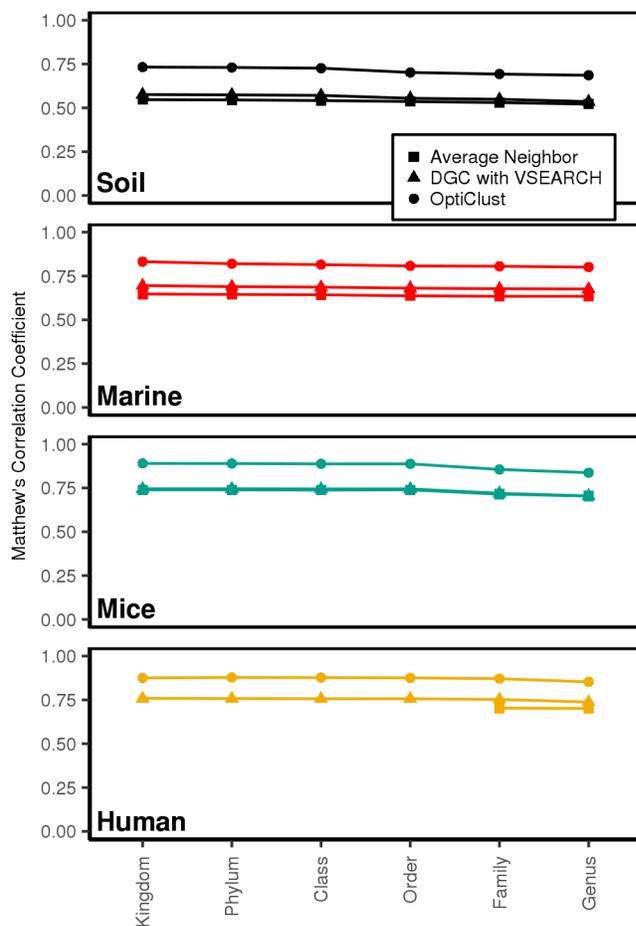
333

334 **Figure 1. Comparison of de novo clustering algorithms.** Plot of MCC (A), number of OTUs  
 335 (B), and execution times (C) for the comparison of *de novo* clustering algorithms when applied to  
 336 four natural and two synthetic datasets. The first three columns of each figure contain the results  
 337 of clustering the datasets (i) seeding the algorithm with one sequence per OTU and allowing the  
 338 algorithm to proceed until the MCC value no longer changed; (ii) seeding the algorithm with one  
 339 sequence per OTU and allowing the algorithm to proceed until the MCC changed by less than  
 340 0.0001; (iii) seeding the algorithm with all of the sequences in one OTU and allowing the algorithm  
 341 to proceed until the MCC value no longer changed. The human dataset could not be clustered by  
 342 the average neighbor, Sumaclus, USEARCH, or OTUCLUST with less than 45 GB of RAM or 50  
 343 hours of execution time. The median of 10 re-orderings of the data is presented for each method  
 344 and dataset. The range of observed values is indicated by the error bars, which are typically smaller  
 345 than the plotting symbol.



346

347 **Figure 2. OptiClust performance** The average execution time (A) and memory usage (B) required  
348 to cluster the four natural datasets. The confidence intervals indicate the range between the  
349 minimum and maximum values. The y-axis is scaled by the square root to demonstrate the  
350 relationship between the time and memory requirements relative to the number of unique sequences  
351 squared.



352

353 **Figure 3. Effects of taxonomically splitting the datasets on clustering quality.** The datasets  
354 were split at each taxonomic level based on their classification using a naive Bayesian classifier  
355 and clustered using average neighbor, VSEARCH-based DGC, and OptiClust.

356 **Supplemental text.** Worked example of how OptiClust algorithm clusters sequences into OTUs.

## 357 **References**

- 358 1. **Schloss PD, Girard RA, Martin T, Edwards J, Thrash JC.** 2016. Status of the archaeal and  
359 bacterial census: An update. *mBio* **7**:e00201–16. doi:<http://doi.org/10.1128/mbio.00201-16>.
- 360 2. **Locey KJ, Lennon JT.** 2016. Scaling laws predict global microbial diversity. *Proceedings of the*  
361 *National Academy of Sciences* **113**:5970–5975. doi:<http://doi.org/10.1073/pnas.1521291113>.
- 362 3. **Rideout JR, He Y, Navas-Molina JA, Walters WA, Ursell LK, Gibbons SM, Chase J,**  
363 **McDonald D, Gonzalez A, Robbins-Pianka A, Clemente JC, Gilbert JA, Huse SM, Zhou**  
364 **H-W, Knight R, Caporaso JG.** 2014. Subsampled open-reference clustering creates consistent,  
365 comprehensive OTU definitions and scales to billions of sequences. *PeerJ* **2**:e545. doi:<http://doi.org/10.7717/peerj.545>.
- 367 4. **Consortium THMP.** 2012. Structure, function and diversity of the healthy human microbiome.  
368 *Nature* **486**:207–214. doi:<http://doi.org/10.1038/nature11234>.
- 369 5. **Eckburg PB, Bik EM, Bernstein CN, Purdom E, Dethlefsen L, Sargent M, Gill SR, Nelson**  
370 **KE, Relman DA.** 2005. Diversity of the human intestinal microbial flora. *Science* **308**:1635–1638.
- 371 6. **Elshahed MS, Youssef NH, Spain AM, Sheik C, Najjar FZ, Sukharnikov LO, Roe BA, Davis**  
372 **JP, Schloss PD, Bailey VL, Krumholz LR.** 2008. Novelty and uniqueness patterns of rare  
373 members of the soil biosphere. *Applied and Environmental Microbiology* **74**:5422–5428. doi:<http://doi.org/10.1128/aem.00410-08>.
- 375 7. **Kozich JJ, Westcott SL, Baxter NT, Highlander SK, Schloss PD.** 2013. Development of a  
376 dual-index sequencing strategy and curation pipeline for analyzing amplicon sequence data on  
377 the MiSeq Illumina sequencing platform. *Applied and Environmental Microbiology* **79**:5112–5120.  
378 doi:<http://doi.org/10.1128/aem.01043-13>.
- 379 8. **Schloss PD.** 2009. A high-throughput DNA sequence aligner for microbial ecology studies.  
380 *PLOS ONE* **4**:e8230. doi:<http://doi.org/10.1371/journal.pone.0008230>.
- 381 9. **Edgar RC, Haas BJ, Clemente JC, Quince C, Knight R.** 2011. UCHIME improves

382 sensitivity and speed of chimera detection. *Bioinformatics* **27**:2194–2200. doi:<http://doi.org/10.1093/bioinformatics/btr381>.  
383

384 10. **Wang Q, Garrity GM, Tiedje JM, Cole JR.** 2007. Naive bayesian classifier for rapid assignment  
385 of rRNA sequences into the new bacterial taxonomy. *Applied and Environmental Microbiology*  
386 **73**:5261–5267. doi:<http://doi.org/10.1128/aem.00062-07>.

387 11. **Schloss PD, Westcott SL, Ryabin T, Hall JR, Hartmann M, Hollister EB, Lesniewski RA,**  
388 **Oakley BB, Parks DH, Robinson CJ, Sahl JW, Stres B, Thallinger GG, Horn DJV, Weber CF.**  
389 2009. Introducing mothur: Open-source, platform-independent, community-supported software  
390 for describing and comparing microbial communities. *Applied and Environmental Microbiology*  
391 **75**:7537–7541. doi:<http://doi.org/10.1128/aem.01541-09>.

392 12. **Caporaso JG, Kuczynski J, Stombaugh J, Bittinger K, Bushman FD, Costello EK, Fierer**  
393 **N, Peña AG, Goodrich JK, Gordon JI, Huttley GA, Kelley ST, Knights D, Koenig JE, Ley RE,**  
394 **Lozupone CA, McDonald D, Muegge BD, Pirrung M, Reeder J, Sevinsky JR, Turnbaugh PJ,**  
395 **Walters WA, Widmann J, Yatsunenko T, Zaneveld J, Knight R.** 2010. QIIME allows analysis of  
396 high-throughput community sequencing data. *Nature Methods* **7**:335–336. doi:<http://doi.org/10.1038/nmeth.f.303>.  
397

398 13. **Schloss PD, Westcott SL.** 2011. Assessing and improving methods used in operational  
399 taxonomic unit-based approaches for 16S rRNA gene sequence analysis. *Applied and*  
400 *Environmental Microbiology* **77**:3219–3226. doi:<http://doi.org/10.1128/aem.02810-10>.

401 14. **Navas-Molina JA, Peralta-Sánchez JM, González A, McMurdie PJ, Vázquez-Baeza Y, Xu**  
402 **Z, Ursell LK, Lauber C, Zhou H, Song SJ, Huntley J, Ackermann GL, Berg-Lyons D, Holmes**  
403 **S, Caporaso JG, Knight R.** 2013. Advancing our understanding of the human microbiome using  
404 QIIME, pp. 371–444. *In* *Methods in enzymology*. Elsevier BV.

405 15. **Schloss PD, Handelsman J.** 2005. Introducing DOTUR, a computer program for defining  
406 operational taxonomic units and estimating species richness. *Applied and Environmental*  
407 *microbiology* **71**:1501–1506.

- 408 16. **Edgar RC**. 2010. Search and clustering orders of magnitude faster than BLAST. *Bioinformatics*  
409 **26**:2460–2461. doi:<http://doi.org/10.1093/bioinformatics/btq461>.
- 410 17. **Rognes T, Flouri T, Nichols B, Quince C, Mahé F**. 2016. VSEARCH: A versatile open source  
411 tool for metagenomics. *PeerJ* **4**:e2584. doi:<http://doi.org/10.7717/peerj.2584>.
- 412 18. **Albanese D, Fontana P, Filippo CD, Cavalieri D, Donati C**. 2015. MICCA: A complete  
413 and accurate software for taxonomic profiling of metagenomic data. *Scientific Reports* **5**:9743.  
414 doi:<http://doi.org/10.1038/srep09743>.
- 415 19. **Mahé F, Rognes T, Quince C, Vargas C de, Dunthorn M**. 2014. Swarm: Robust and fast  
416 clustering method for amplicon-based studies. *PeerJ* **2**:e593. doi:<http://doi.org/10.7717/peerj.593>.
- 417 20. **Sun Y, Cai Y, Liu L, Yu F, Farrell ML, McKendree W, Farmerie W**. 2009. ESPRIT: Estimating  
418 species richness using large collections of 16S rRNA pyrosequences. *Nucleic Acids Research*  
419 **37**:e76–e76. doi:<http://doi.org/10.1093/nar/gkp285>.
- 420 21. **Cai Y, Sun Y**. 2011. ESPRIT-tree: Hierarchical clustering analysis of millions of 16S rRNA  
421 pyrosequences in quasilinear computational time. *Nucleic Acids Research* **39**:e95–e95. doi:<http://doi.org/10.1093/nar/gkr349>.
- 422 [//doi.org/10.1093/nar/gkr349](http://doi.org/10.1093/nar/gkr349).
- 423 22. **Edgar RC**. 2013. UPARSE: Highly accurate OTU sequences from microbial amplicon reads.  
424 *Nature Methods* **10**:996–998. doi:<http://doi.org/10.1038/nmeth.2604>.
- 425 23. **Mahé F, Rognes T, Quince C, Vargas C de, Dunthorn M**. 2015. Swarm v2: Highly-scalable  
426 and high-resolution amplicon clustering. *PeerJ* **3**:e1420. doi:<http://doi.org/10.7717/peerj.1420>.
- 427 24. **Barriuso J, Valverde JR, Mellado RP**. 2011. Estimation of bacterial diversity using next  
428 generation sequencing of 16S rDNA: A comparison of different workflows. *BMC Bioinformatics*  
429 **12**:473. doi:<http://doi.org/10.1186/1471-2105-12-473>.
- 430 25. **Bonder MJ, Abeln S, Zaura E, Brandt BW**. 2012. Comparing clustering and pre-processing in  
431 taxonomy analysis. *Bioinformatics* **28**:2891–2897. doi:<http://doi.org/10.1093/bioinformatics/bts552>.
- 432 26. **Chen W, Zhang CK, Cheng Y, Zhang S, Zhao H**. 2013. A comparison of methods for

433 clustering 16S rRNA sequences into OTUs. PLOS ONE **8**:e70837. doi:[http://doi.org/10.1371/](http://doi.org/10.1371/journal.pone.0070837)  
434 [journal.pone.0070837](http://doi.org/10.1371/journal.pone.0070837).

435 27. **Huse SM, Welch DM, Morrison HG, Sogin ML.** 2010. Ironing out the wrinkles in the rare  
436 biosphere through improved OTU clustering. Environmental Microbiology **12**:1889–1898. doi:[http:](http://doi.org/10.1111/j.1462-2920.2010.02193.x)  
437 [//doi.org/10.1111/j.1462-2920.2010.02193.x](http://doi.org/10.1111/j.1462-2920.2010.02193.x).

438 28. **May A, Abeln S, Crielaard W, Heringa J, Brandt BW.** 2014. Unraveling the outcome of 16S  
439 rDNA-based taxonomy analysis through mock data and simulations. Bioinformatics **30**:1530–1538.  
440 doi:<http://doi.org/10.1093/bioinformatics/btu085>.

441 29. **Sun Y, Cai Y, Huse SM, Knight R, Farmerie WG, Wang X, Mai V.** 2011. A large-scale  
442 benchmark study of existing algorithms for taxonomy-independent microbial community analysis.  
443 Briefings in Bioinformatics **13**:107–121. doi:<http://doi.org/10.1093/bib/bbr009>.

444 30. **White JR, Navlakha S, Nagarajan N, Ghodsi M-R, Kingsford C, Pop M.** 2010. Alignment and  
445 clustering of phylogenetic markers - implications for microbial diversity studies. BMC Bioinformatics  
446 **11**:152. doi:<http://doi.org/10.1186/1471-2105-11-152>.

447 31. **Al-Ghalith GA, Montassier E, Ward HN, Knights D.** 2016. NINJA-OPS: Fast accurate  
448 marker gene alignment using concatenated ribosomes. PLOS Computational Biology **12**:e1004658.  
449 doi:<http://doi.org/10.1371/journal.pcbi.1004658>.

450 32. **He Y, Caporaso JG, Jiang X-T, Sheng H-F, Huse SM, Rideout JR, Edgar RC, Kopylova E,**  
451 **Walters WA, Knight R, Zhou H-W.** 2015. Stability of operational taxonomic units: An important  
452 but neglected property for analyzing microbial diversity. Microbiome **3**. doi:[http://doi.org/10.1186/](http://doi.org/10.1186/s40168-015-0081-x)  
453 [s40168-015-0081-x](http://doi.org/10.1186/s40168-015-0081-x).

454 33. **Kopylova E, Navas-Molina JA, Mercier C, Xu ZZ, Mahé F, He Y, Zhou H-W, Rognes T,**  
455 **Caporaso JG, Knight R.** 2016. Open-source sequence clustering methods improve the state of  
456 the art. mSystems **1**:e00003–15. doi:<http://doi.org/10.1128/msystems.00003-15>.

457 34. **Schmidt TSB, Rodrigues JFM, Mering C von.** 2014. Limits to robustness and reproducibility  
458 in the demarcation of operational taxonomic units. Environ Microbiol **17**:1689–1706. doi:[http:](http://doi.org/10.1111/1365-3113.12444)

459 [//doi.org/10.1111/1462-2920.12610](https://doi.org/10.1111/1462-2920.12610).

460 35. **Schloss PD**. 2016. Application of a database-independent approach to assess the quality of  
461 operational taxonomic unit picking methods. *mSystems* **1**:e00027–16. doi:[http://doi.org/10.1128/  
462 msystems.00027-16](http://doi.org/10.1128/msystems.00027-16).

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

466 37. **Matthews B**. 1975. Comparison of the predicted and observed secondary structure of t4  
467 phage lysozyme. *Biochimica et Biophysica Acta (BBA) - Protein Structure* **405**:442–451. doi:[http:  
468 //doi.org/10.1016/0005-2795\(75\)90109-9](http://doi.org/10.1016/0005-2795(75)90109-9).

469 38. **Pruesse E, Quast C, Knittel K, Fuchs BM, Ludwig W, Peplies J, Glockner FO**. 2007.  
470 SILVA: A comprehensive online resource for quality checked and aligned ribosomal RNA sequence  
471 data compatible with ARB. *Nucleic Acids Research* **35**:7188–7196. doi:[http://doi.org/10.1093/nar/  
472 gkm864](http://doi.org/10.1093/nar/gkm864).

473 39. **Baxter NT, Ruffin MT, Rogers MAM, Schloss PD**. 2016. Microbiota-based model improves  
474 the sensitivity of fecal immunochemical test for detecting colonic lesions. *Genome Medicine* **8**.  
475 doi:<http://doi.org/10.1186/s13073-016-0290-3>.

476 40. **Schloss PD, Schubert AM, Zackular JP, Iverson KD, Young VB, Petrosino JF**. 2012.  
477 Stabilization of the murine gut microbiome following weaning. *Gut Microbes* **3**:383–393. doi:[http:  
478 //doi.org/10.4161/gmic.21008](http://doi.org/10.4161/gmic.21008).

479 41. **Johnston ER, Rodriguez-R LM, Luo C, Yuan MM, Wu L, He Z, Schuur EAG, Luo Y, Tiedje  
480 JM, Zhou J, Konstantinidis KT**. 2016. Metagenomics reveals pervasive bacterial populations  
481 and reduced community diversity across the alaska tundra ecosystem. *Front Microbiol* **7**. doi:[http:  
482 //doi.org/10.3389/fmicb.2016.00579](http://doi.org/10.3389/fmicb.2016.00579).

483 42. **Henson MW, Pitre DM, Weckhorst JL, Lanclos VC, Webber AT, Thrash JC**. 2016. Artificial  
484 seawater media facilitate cultivating members of the microbial majority from the gulf of mexico.

485 mSphere 1:e00028–16. doi:<http://doi.org/10.1128/msphere.00028-16>.

486 43. **Schloss PD**. 2010. The effects of alignment quality, distance calculation method, sequence  
487 filtering, and region on the analysis of 16S rRNA gene-based studies. PLOS Comput Biol  
488 6:e1000844. doi:<http://doi.org/10.1371/journal.pcbi.1000844>.

489 44. **R Core Team**. 2015. R: A language and environment for statistical computing. R Foundation  
490 for Statistical Computing, Vienna, Austria.

491 45. **Ram K, Wickham H**. 2015. wesanderson: A wes anderson palette generator.

492 46. **Wickham H, Francois R**. 2016. dplyr: A grammar of data manipulation.

493 47. **Wickham H**. 2016. tidyr: Easily tidy data with 'spread()' and 'gather()' functions.

494 48. **Wilke CO**. cowplot: Streamlined plot theme and plot annotations for 'ggplot2'.

495 49. **Wickham H**. 2009. ggplot2: Elegant graphics for data analysis. Springer-Verlag New York.