The line numbers referred to in our response to reviewers correspond to those in the version of the manuscript with the "track changes" noted (i.e. `track_changes.pdf`).

Reviewer 1

Major comments

Optimizing a criterion is a good idea. It would be useful to clarify if each of the other algorithms aims to optimize some other criterion and if so which one. There is some text about this in the discussion, but this information might work better earlier in the manuscript. It would also be helpful to see how each algorithm performs on these other metrics (e.g., minimizing FPs). You can argue that MCC is a better criterion to use, but it is also good to show if the methods that aim to optimize another criterion actually do so and how MCC-based OptiClust does with regards to these metrics. In addition, some further discussion about the relative costs of FPs and FNs in typical metagenomics applications and downstream studies would provide some context for your argument to use MCC.

We have added text to the Introduction to describe that the furthest and nearest neighbor algorithms no not allow FPs and FNs, respectively (L83-84). We have also added text to the Discussion (L244-251) pointing out that More FPs would increase the number of OTUs while more FNs would collapse OTUs together. It is difficult to know which is worse since both alpha and beta diversity metrics are tied to the number and composition of OTUs. As we point out, this is why it is important to jointly minimize the number of FPs and FNs.

For the simulations, it would be nice to show the true values of some of the output measures (e.g., number of clusters) versus the estimated values.

Unfortunately, this is a major problem with evaluating clustering algorithms. Except under highly contrived conditions, we do not know what the true values are for output measures such as the number of clusters. This problem is outlined in L62-88. This is why we have developed the concept of the MCC over our last several papers on the subject of OTU clustering.

A systematic evaluation of the influence of additional variables on performance (i.e., accuracy metrics, RAM, and speed) would provide better insight into expected performance on other data sets. The analyzed data sets do cover a broad range of environments with different types of communities. Even with those data, it would be nice to see performance metrics plotted versus phylogenetic diversity, percentage of sequences in the reference database (for pre-assignment analyses), and sequencing variables (e.g., read length, quality scores). In the simulations, you could additionally introduce realistic sequencing errors evaluate how performance changes.

As the reviewer correctly points out, there is no ideal dataset. This is why we selected four datasets that represent a variety of environments that are currently popular among microbial ecologists. We also added the simulated datasets (i.e. even and staggered) since these have been used in prior studies evaluating clustering methods. Given that the nature of the algorithm is to optimize the MCC coefficient (or any other coefficient) and we have demonstrated that it is effective at doing this, we don't think it is necessary to add additional datasets to reinfoce the point. To the reviewer's suggestion, we have added the number of OTUs that were found using the OptiClust algorithm and the number of pairwise distances that were equal to and smaller than 0.03 to the summary statistics in Table 1. Unfortunately, some of the datasets are too large and the sequencing data too noisy to permit the construction of the phylogenetic trees that would be needed to calculate the phylogenetic diversity for each study. As for comparing to percentage of sequences in a reference database, we feel that this is would be a misleading point as we have previously shown that reference-based methods are inferior to *de novo* clustering methods (L47; reference added) and the identity of the sequences does not affect the performance of the algorithm (i.e. we could have used synthetic DNA sequences and communities and reached the same conclusion). Finally, the the point regarding read length, quality scores, and sequencing errors we do feel that these analyses are beyond the scope of the current paper. OptiClust only uses pairwise distances as inputs to the clustering and data quality is a very important, but different problem. Given the data provided in Table 1 and Figure 2, we have demonstrated the effect of increasing the number of distances on the performance of the algorithm.

This may be beyond the scope, but it would be very interesting to see the performance

of OptiClust with other criteria as the target of optimization. In other words, how much of the performance is due to targeting MCC as the criterion versus the algorithm itself as an alternative to hierarchical clustering. You could investigate this by comparing the OptiClust algorithm minimizing FPs, for example, as compared to hierarchical clustering minimizing FPs. If this analysis is not performed, I recommend discussing that the algorithm for building clusters and the criterion that is being optimized are evaluated together in your analyses and not disentangled.

Because both reviewers inquired about this point, we added two supplementary figures. The first demonstrates the performance of the algorithm when we optimized clustering based on a variety of other metrics (Figure S1). The second, demonstrates how the MCC values improve when optimizing for the MCC value with each successive iteration (Figure S2).

Providing the analysis pipeline is great. Users who want to apply the method may like to have more description of the implementation and a stable link to the code (even if this is the same as mothur). To enable applications outside OTU clustering, you should also have a code base that is not embedded in mothur.

The code for the OptiClust algorithm is available as C++ source code within mothur (e.g. opticluster.cpp at https://github.com/mothur/mothur/tree/master/source). We are reluctant to separate the OptiClust source code from mothur to make it a stand alone tool. This makes maintenance of the software cumbersome and is contrary to the already successful development strategy that we have had for mothur. It should be noted that mothur is wrapped within other tools such as QIIME. Also, mothur can be run from the command line or using scripts. Because of these points, we do not feel it is necessary to separate the code out of the main package.

Minor comments

Line 69: Typo: "to be recapitulate"

This has been fixed.

Line 132: Not clear why the number of OTUs is mentioned here

This was meant to indicate that the small difference in MCC values (0.0001) had a relatively small effect on the number of OTUs that result from the algorithm. Although we do not feel that the number of OTUs is a particularly relevant metric of clustering quality, we did feel that it was important to show the effect of small differences in MCC on measures of community richness.

Line 138: With partial convergence, does the output include the last value for change in MCC?

Yes, the MCC values are provided as part of the output of running the command.

Lines 140-150: Mention if agglomerative hierarchical methods are also better than divisive (cite literature or own data).

We have added a sentence to the Introduction (L52) indicating that all of the *de novo* methods used by microbial ecologists are agglomerative.

Lines 151-153: Processing the sequences/OTUs deterministically is a good option for alleviating dependence on data order.

We have added a clause to this sentence to indicate that this really isn't possible since the algorithms must find a way to break ties when a sequence is equally similar to multiple OTUs.

Line 161: Typo: "values considerably smaller"

This has been fixed.

Line 170: Is disk or CPU also useful to measure?

In our experience users are limited by RAM and want the methods to complete as quickly as possible. This is shown by the failure of the USEARCH methods to load the datasets in the available RAM or other algorithms to finish in 50 hrs with 48 GB of RAM. The availability of disk space is rarely a problem.

Lines 182-185: Clarify that this is assuming the pairwise similarities are similar to the

analyzed data set. Can you say something about what happens as diversity increases or decreases?

We have added text (L175-177) to indicate that speed was proportionate to the number of distances smaller than 0.03. It does not necessarily appear to be related to the number of sequences or the number of OTUs.

Line 187: Remind the reader that this is "open reference"

A sentence has been added to clarify the differences between our method and open reference clustering (L198-201). Briefly, in open reference clustering sequences are classified and any that do not map to a reference are clustered using a de novo approach. In our approach, all sequences are classified (not mapped) to a broad taxonomic level and then all sequences are clustered using a de novo approach.

Line 189: Typo: "reduce" should be past tense

This has been fixed.

Line 197: Are all of these with taxonomic assignment first? Clarify.

This has been clarified.

Line 204: Explain why (i.e., what types of errors the extra clusters represent).

This has been clarified.

Line 232: Typo: "effectively ask"

This has been fixed.

Line 239: Typo: "are encourage"

This has been fixed.

Reviewer 2

Major comments

This is a well described, self-enclosed, well implemented piece of work that does not require major revision. However, there are minor issues which should be addressed.

Minor comments

On page 6 and 7 the authors compare the speed of OptiClust to other methods; however, I do not think Figure 1 really conveys this, mostly due to the log-scaled y-axis for the time panel. Whilst I recognise why the log-scale has been used, the authors may consider a different way to visually present the data that better conveys the message "The OptiClust algorithm was considerably faster than the hierarchical algorithms and somewhat slower than the heuristic-based algorithms"

We tried several methods for improving the visualization. First, we made all speed values relative to those in the first column. The problem with this was that the range of values was still considerable and obscured the ability to see the differences. Second, we added grid lines that overlapped with the values from the first column and extended to the right along the plot. This ended up making the figure look overly busy. Third, we attempted to group the methods within the sample types (we had grouped the samples within the methods type), but it was difficult to clearly represent the 13 different methods and 6 samples efficiently. Ultimately, we decided to add a thin gridline at the major y-axis values to give the reader something to anchor their perspective as they looked across the plot to compare the different methods.

The comparison of RAM used (at the top of page 7) may also be better demonstrated visually.

At L120-123, our goal wasn't to compare RAM usage between methods. We merely wanted to point out that the USEARCH-based approaches could not handle large datasets without purchasing the

64-bit version of the software and that OTUCLUST and SumClust could not store the data within 45 GB of RAM.

It wasn't clear to me which software implementation was used for "Nearest Neighbour" and "Furthest Neighbour".

These were run using the implementations found in mothur (L313).

The authors describe the potential implications of optimising a metric other than MCC - I wonder if they have actually done this?

We have added text and two supplemental figures to the manuscript in response to this comment and that of the first reviewer. We addressed this point above.