MICHIGAN STATE
UNIVERSITY

May 1st, 2012

Dear Editors,

We thank the editor and three reviewers for their excellent and detailed comments.

In response, we have made the appropriate corrections and revisions to the paper; the methods, results, and conclusions have not changed significantly.

We have rewritten the abstract and introduction to emphasize that we believe that this paper presents both a theoretical advance in data structures as well as a specific useful implementation. While we unfortunately do not have the space to discuss the application to larger data sets, we have applied this approach to numerous other (*much* larger) data sets and are in the process of writing up those results; unfortunately, validation of those results requires considerable effort and discussion, and could not possibly fit into this paper. In this paper we make the key advance, namely that the data structure is accurate for a useful regime of parameter space, and can be used for partitioning in that regime.

We have also updated several references in response to recent publications (e.g. the Cortex assembler).

In response to the editor, we have eliminated mention of "divide and conquer" and sharpened the discussion of partitioning in the introduction.

With respect to missing edges, we represent the graph as an implicit de Bruijn graph, as is used in several extant assemblers. When reads abut but do not overlap, this can lead to false positive connections; the paper demonstrates that, as with the Bloom filter false positives, they do not lead to erroneous connections in the large-scale graph structure. We have amended the first paragraph in the results section accordingly.

In response to Reviewer 1's concern that our comparison of memory usage between an exact and inexact representation was inadequate, we have included a reference to the theoretical lower bound of an exact representation (see Figure 5). In Figure 5, we show that for a large range of values, our inexact representation is significantly lower memory than the theoretical lower bound. We believe that this is preferable to a discussion of actual

COLLEGE OF ENGINEERING

Department of Computer Science and Engineering

Michigan State University
3115 Engineering Building
East Lansing, Michigan
48824-1226

(517) 353-3148
FAX: (517) 432-1061

implementations, which vary from assembler to assembler and release to release.

We have added the correct accession number for the MSB2 data set.

As Reviewer 1 noted, this graph representation is indeed useful for many other purposes! We are in fact deploying it for a wide range of purposes, including Hidden Markov Model-based traversal and assembly, graph-based error correction, and artifact detection in large data sets. We have amended the concluding paragraph to discuss this. However, we are concerned that a stronger forward-looking statement would be unacceptable given the PNAS requirement that we not reference unpublished work or personal communications.

In response to Reviewer 2's concern about the diameter experiments and sparsity, we have addressed these points by breaking up the paragraph for clarity, and noting that our results demonstrate the lack of long "bridges" falsely appearing in graphs residing in sparsely occupied k-mer spaces. To help illustrate this, we have colored the false positive k-mers in Figure 2 in red and updated our description of the figure.

We have also corrected the digraph statement as per Reviewer 2.

We have unfortunately not yet been able to demonstrate that the critical threshold property holds for *all* k, although the convergence for larger k is implied in Figure S1. Because we can show that it holds for k sizes that are useful for partitioning, however, the effective use of the data structure for partitioning does not depend on it. Nonetheless this is an active area of research for us.

With respect to the software, we have been working intensively on making this a useful component of an overall pipeline for metagenome assembly (see http://ged.msu.edu/angus/nih-hmp-2012/). However, because this relies on several additional unpublished approaches and evaluation methods, we do not include this in the paper.

Reviewer 3 makes an excellent point regarding both the novelty of the algorithm and the generalizability. The algorithm for partitioning is neither novel nor particularly clever, and we did not intend to highlight it as such; in fact, partitioning has been applied to both metagenomic data sets (MetaVelvet and MetaIDBA) and mRNAseq (Trinity). Rather, the novelty is in the use of the low memory data structure *for* partitioning as a way to reduce overall assembly requirements. Prior to our effort, partitioning the graph based on e.g. coverage (MetaVelvet) was used for improving assembly, but not for decreasing overall memory requirements.

With respect to the generalizability of partitioning, we too have observed that mRNAseq data sets cluster into a giant component, in part because of poly-A tails; disentangling this into components is only possible with approximate or heuristic approaches such as those used by Trinity. Trinity could certainly make use of our data structure for this purpose, and we have discussed this with the Trinity developers. However, in practice mRNAseq

data sets do not currently stress computational resources as much as soil metagenome assembly, and so we have chosen to focus our initial application of the graph structure on the bigger problem of metagenome assembly. In response to Reviewer 3's comment, we have integrated a discussion of partitioning in Trinity in two places in the manuscript.

Reviewer 3 also correctly points out that the MSB2 data set is very low coverage and assembles into a very small set of contigs. This highlights the challenges with these data sets: even with 35m reads, comprising approximately 2.5 GB of sequence, we can sample only the highest abundance members of microbial populations in agricultural soil and hence obtain only a small amount of assembled sequence. We believe that this is still an important (although limited) demonstration of the basic concept of partitioning on real data, and have altered the text in the discussion to emphasize that this conclusion is limited.

In response to Reviewer 3's specific comments,

1. We have altered the structure so as to simplify our main argument.
2. We have altered the abstract to emphasize that we are not altering the assembly algorithm.
3. The original "512 GB" is the correct number from the ALLPATHS-LG paper. The high memory use is from the inclusion of all the sequencing errors in the 100x human genome sequencing necessary to obtain a good assembly.
4. We have changed the text to clarify that the "20-fold decrease in memory usage" applies to the maximum memory usage across the entire assembly process, which may or may not include partitioning. That is, if we can apply partitioning in 1 GB of RAM and assemble the partitioned data in 500 MB of RAM, the 1 GB is the appropriate comparison relative to assembling unpartitioned data in 32 GB of memory.
5. We have replaced "constant" with "fixed" throughout.
6. We have replaced "efficiently" with "accurately", and emphasized the random/uncorrelated nature of false positives.
7. The random sequence is generated with 50/50 AT/GC, and we have added this information to the Methods. Reviewer 3 is absolutely correct that real DNA is quite different from random, which is why we also applied partitioning to the MSB2 data set, which represents not only real DNA but real Illumina sequencing.
8. Figure 2 displays a flattened visualization of the multidimensional graph structure for the same sequence loaded at different false positive rates, with both real and connected false-positive k-mers indicated. We have updated the description of Figure 2 to make this clearer, and colored the false positive k-mers in red. The strategy used to generate the graphs (breadth-first traversal from "real" k-mers) is emphasized only to indicate that we are only plotting false k-mers *connected* to real k-mers; we have updated the text to make this clearer.
9. We are using percentile bootstrap because over the range of experiments, we are sampling from multiple different (non-normal) distributions. We believe this is an appropriate use.
10. We have included error bars on Figure S1; our apologies for omitting them!

11. We have corrected the text to explain that we used a 50 bp sequence with the initial 8 bp postpended, so as to create a circular graph.
12. We have updated the heading to be more precise: it is now “Erroneous k-mers from sequencing eclipse graph false positives”.
13. We have updated the heading to be “False connected k-mers”.
14. Our initial implementation of the software had an inefficiency with respect to storing false positives that led to higher memory usage at high false positive rates. This has been corrected in the most recent revisions, and we have updated the results in the paper accordingly.
15. This sentence (“This feature…”) was indeed confusing and has been removed.
16. We have corrected this sentence in the Methods – thank you!

We believe that these revisions represent a substantial improvement in clarity and precision over the original submission, and we would like to thank the editor and reviewers for their comments.

Sincerely,

C. Titus Brown, for the authors.