# Summary of Changes to the Manuscript

## Changes to text

In response to both reviewers, we have shortened the paper and improved flow, which led to many minor changes throughout. Of particular note

1. We changed the description of the programs in the software pipeline from (riboSelect.py, riboSeed.py, etc), to reflect their revised implementation as subcommands (select, seed, etc). We believe these changes aid readability and provide a streamlined user interface to the riboSeed tool.

2. We condensed results tables from the GAGE-B benchmarking, hybrid assembly, and case study into a single table.

3. We added a section to supplementary data justifying our decision to remove one of the GAGE-B sample from the analysis due to likely contamination.

## Changes to Data

* A new set of 25 *E. coli* isolates was selected. e While addressing the reviewers’ comments we discovered a bug in our strain randomization that compromised proper seeded, repeatable pseudorandom selection.
* We omitted the GAGE-B *B. cereus* HiSeq dataset from the benchmarking analysis, as we suspect it to be contaminated; an explanatory section has been added to the supplementary data

## Changes to Analysis

* Software versions; as a result of using the conda ecosystem for installation, we have chosen to alter third-party software dependencies: .
  + Barrnap 0.7 -> 0.8
  + MAFFT 7.215 -> 7.310
  + EMBOSS 6.6.0 -> 6.5.7
* Updates to the riboSeed codebase are described in the GitHub commit logs (exclusion of inappropriate kmers, changed handling of initial reference rDNA region, etc.)
* As noted above, we corrected the issue with pseudorandom number generation for strain selection.
* We increased the *E. coli* and *K. pneumoniae* artificial read simulation replicates from 5 to 9

## Changes to results

* The average substitution rate in the 25 *E. coli* sequences calculated from Parsnp was changed from 0.0062 to 0.0035. This was a reflection of two changes:
  + This resulted in a new set of 25 strains being selected. See above.
  + We now use Parsnp’s “curated” (“-c”) parameter to force all sequences to be considered rather than using MUMi (maximal unique matches index) to exclude more divergent genomes. This elevated the average substitution rate.
* We amended the results from the *Methanobacterium formicicum* JCM10132 dataset, as

we are investigating our observation that SPAdes performance is modified when the chosen kmer length parameter exceeds the length of input reads. The updated version (0.4.35) of riboSeed used in this study now prevents users from choosing this combination of parameters, and we note the effect this has in that section of the manuscript.

# Referee: 1

## Comments for the Author

#### 1. *The manuscript need to be shorten considerably and proper flow to be maintained.*

As per the reviewer’s suggestion we have attempted to shorten the manuscript, but find it difficult to shorten substantially. We would welcome specific suggestions for where we might achieve a shorter manuscript with no loss of clarity.

The riboSeed method relies on two findings regarding the rDNA flanking regions: within-genome uniqueness, and between-genome conservation. We have not found previous reports of these observations with detail pertinent to our implementation, which we believe that we need to establish in order to describe and justify the method. That being said, we have attempted to streamline the implementation and descriptions of the software components for better flow, and revised the paper throughout.

#### 2. *Reproduction of the method is difficult from the written method. Needs the method to be simplified.*

The pipeline has been reconfigured to be run with a single script instead of three steps of preprocessing, operon prediction, and *de fere* novo assembly. Further, it can now be installed using the "conda" ecosystem, which removes the need for the user to satisfy additional third-party software dependencies

#### 3. *It is known fact that flanking regions of rDNA are conserved in a genome, so any other novelty used in this strategy?*

While it is certainly a truism and widely-held belief that this is the case, we have not found a specific report of this observation in the literature. We believe that we need to establish this as fact to justify the methodology, and we would be happy to cite any prior work that supports the relevant claims directly. We have also characterised these flanking regions across a range of bacterial species. We believe this to be central to supporting the applicability of our reported method but, as before, have been unable to identify prior work establishing the same observation. We are the first (to our knowledge) to report specific use of this characteristic to improve genome assembly.

#### 4. *The article is already available online: http://www.biorxiv.org/content/early/2017/07/14/159798 (doi.org/10.1101/159798) at bioRxiv. Therefore I am not sure if it violates the NAR policy.*

NAR's policies do not conflict with preprints. (https://academic.oup.com/nar/pages/Policies)

# Referee: 2

## Comments for the Author

### General Comments:

#### 1. *This manuscript describes a DNA sequence assembly pipeline, riboSeed, that addresses a key problem in de novo microbial genome assembly with short read sequences — repeat regions. riboSeed implements a so-called “de fere novo” method of sequence assembly by first recruiting reads to the ribosomal operons of a closely related (and already completed) prokaryotic genome. Reads that recruit to the ribosomal operons and flanking regions (1 Kbp to the left and right of each operon) are then co-assembled with that region. The resulting contigs from each region are then stitched together with 5 Kbp of ambiguous (N) bases creating a “pseudocontig”. The pseudocontig is then refined by iteratively recruiting reads and reassembling. Once (if) the pseudocontig is good enough it is then co-assembled with the raw reads as either a SPAdes “trusted” or “untrusted” contig, depending on the quality of the pseudocontig.*

#### *In simulated and real data the riboSeed de fere novo method out-performed the de novo assembly method. While that result is good the see, it isn’t necessarily surprising, as reference-based assemblies should always out-perform de novo assemblies in repeat regions.*

We agree that the result is not surprising, and we were not surprised to obtain improved assembly through repeat regions using our method. In this manuscript we are reporting methodology and software to obtain usefully improved assembly in genomic regions that are traditionally difficult to resolve, rather than making a claim to a surprising outcome.

#### 2. *Overall I think the manuscript is well-written, there are however a few tables and figures I would like to see revised and improved (see specific comments). Also, the first two subsections of the Results section seem to be more Materials and Methods than actually reporting results. I appreciate that the authors report other genome polishing tools for gap closure, but I wish some tool comparisons were done so I could understand how much better riboSeed is doing compared to the other tools. I think the concept of “de fere novo” is humorous and makes sense to me, but I don’t see that sticking in the community, as the term “reference-based” is already widely accepted for this methodology (granted this method is a bit more de novo than most reference-based methods).*

We thank the reviewer for their kind words and useful comments!

We think that that characterisation of rDNA flanking region variation within and between genomes is not previously reported in the literature - particularly not in this detail - so, as a novel observation we believe that this belongs in the results section, though we are open to argument and pointers to prior work we may have missed, that we could cite. The subsequent findings reported in the paper require the substantiation of these characteristics; we believe that removing this section would undermine the reliability of those later claims.

We have addressed the comments on the tables and figures below. "*De fere novo*" may not catch on, but we think it helps distinguish riboSeed’s approach from *de novo* assembly, for the purpose of this paper. As the reviewer notes, there is some clear water between this method and the conventional view of “reference-based” assembly, and it may be that referring to riboSeed as ‘reference-based’ might prime a reader to expect a different methodology or outcome.

#### 3. *I regret that I do not have sufficient time to download and use the riboSeed code on a test dataset, but I will commend the authors for putting together a clear and well-documented GitHub repository. As a casual observer I’m a bit intimidated by how many parameters and scripts there are in the code base, but from the perspective of a researcher who is desperate to close a bacterial genome I think I would be willing to forgo time spent tinkering with all of the parameters if it helped the assembly. riboSeed seems less automated than most reference-based assembly pipelines.*

We appreciate the reviewer’s appraisal of the Documentation found on the GitHub repo. We take on board their comments regarding the complexity of the interface, and have attempted to streamline this. As of version 0.4.35, he user interface now has a simplified subcommand structure that we hope the reviewer might find more intuitive. For parameters choice, we simplified execution of the script by exposing the most commonly used parameters through a ‘runner’ script. All the functionality is still available to users, but this simplifies execution of the pipeline considerably

Additionally, we have made use of Readthedocs.io (<http://riboseed.readthedocs.io/en/latest/>) to host more, and clearer, documentation.

We welcome continuous feedback on usability and performance, and specific comments can also be left at the GitHub repository: (https://github.com/nickp60/riboSeed/issues)

#### 4. *Lastly, I have left several specific comments below, some of which are minor, others are a little more important. The largest conflicts I could find in the methods/implementation of riboSeed is:*

#### [A] *Likely the most important decision a riboSeed user has is “which reference should I use” and the authors don’t give a great deal of advice on that subject (though KGCAK is recommended). They do demonstrate that selecting the (obviously wrong) reference can lead to a poor assembly (Fig. 3), but I would like to see a more specific set of instructions for users to follow to get a good reference sequence (maybe using phylogeny?). Also, how easy is it to know that a genome has all rDNA regions assembled and known to be in the right context/order? Seems like there’s a need for tools/methods to verify that more easily (though the authors do mention 16Stimator).*

This is an important critique and we thank the reviewer for raising it. We have added suggestions in the materials and methods section to help the user make this decision, but we recognise that the situation may be quite different in a well-sequenced taxon such as *E. coli* than it is for a less well-studied bacterium, and it is difficult to be prescriptive about references with confidence.

In short, we propose two protocols, a robust method and a quicker, easier method. The robust protocol relies on Kraken to find the most similar sequenced strain, and the quick method uses the Reads2Type web interface at https://cge.cbs.dtu.dk/services/Reads2Type/ to find related sequenced strains.

We acknowledge that, unless long reads were available that read through a complete rDNA region when assembling the reference, there will always be some doubt concerning their assembly/context/order, and therefore its trustworthiness s . We attempt to increase confidence in correct assembly by our *de fere novo* method of using independent assemblies to reconstruct the same rDNA regions. A more detailed survey of the applicability of complete assemblies submitted to public repositories to assess the consistency of copy number and rDNA context is beyond the scope of this methods paper, but is pencilled in for future work.

We are able to propose some guidelines. We are, from the work reported here, confident in asserting that there is a limiting threshold of flanking region identity beyond which riboSeed fails to work well; this approximates to ≈5% difference in nucleotide sequence identity across the flanking region. Assuming that the flanking region is representative of coding regions/the bulk of the genome, this implies that useful references will have <≈5% sequence difference across the complete genome. To the extent that bacterial species boundaries imply ≈5% sequence difference across the complete genome, identification of a candidate reference genome from the same species with Kraken/reads2type etc. should be sufficient to identify good reference genome candidates. Where several candidates are available, the genome with greatest sequence identity to a draft genome of the organism being assembled may well be the most appropriate choice - this could be identified from the subset of complete genomes of that species with, e.g. MiSi or ANI.

#### 5 *[B] The use of the reference genome’s ribosomal operon as a “trusted contig” in the initial SPAdes assembly. It’s still unclear to me how SPAdes handles trusted (and untrusted) contigs and I’m also confused by why they recommend users not to feed in contigs of “related species”, which is what the pipeline does. Maybe they’re worried that with a related reference the assembly will drift away from the sample’s genome and toward that reference. That might not be an issue for riboSeed because the pseudocontigs are later re-assembled iteratively by only the reads. Would want to see how (or even if) using the reference genome as a trusted contig helps or hurts the results.*

We thank the reviewer for this insightful observation, and have added an explanatory note to the text, as well as a more detailed note to the supplementary data “Parameters” section.

We hope that by showing the effect of using "trusted-contigs", users will feel sufficiently informed to choose the appropriate option. The reasons for using reference as either "trusted or untrusted" by default include the difficulty spades has with assembling reads mapping to rDNAs de novo. One of the issues is that of the coverage depth at the rDNA. Because the mapping scheme allows multiple mappings, the non-polymorphic regions of the rDNA will have higher coverage than the regions that differentiate between rDNAs. Because of this, coverage correction is not enabled for the subassemblies, as SPAdes often returns an error; at the advice of the authors, the parameters for subassembly spades runs are to use single-cell mode (as single cell data shares some of the coverage issues). As an alternative, we have also implemented an alternate consensus assembly option for this first iteration, where reads mapping to the rDNA regions are used to create a consensus sequence, which is then used as the pseudocontig in the second iteration. We have found that this produces comparable “long reads” after the iterations are complete, indicating that for an average dataset, using the reference rDNA region in the first iteration’s subassembly is comparable to using a consensus-based approach.

### Specific Comments:

#### *1. Pg. 1 Line 30 (abstract): Need to italicize “de novo”*

The NAR LaTeX template does not appear to render italics in the abstract environment; perhaps this is rectified during publication?.

#### 2. P*g. 1 Line 34 (intro): “Sequencing of the 16S ribosomal region is widely used to identify bacteria and explore […]” “Bacteria” should probably be “prokaryotes”.*

Corrected

#### 3. *Table 1: Why are only two dates shown? They’re relatively close to one another, too. Might be better to include annual counts that go back to, say, 2007. This should demonstrate that the gap between the number of complete genomes and total genomes sequenced begins to rapidly increase after short read sequencing comes into play.*

We have changed table one to be a small figure showing the counts of genome assembly levels between 2000 and 2017.

#### 4. *Fig. 1: I’m assuming when the pseudocontigs are being stitched together that the order of the rDNA operons will be correct. No mention of the order of the operons. Is that not important?*

We have made a note in the “seed“ subsection confirming the reviewers assumption about the importance of the order of the pseudocontigs.

#### 5. *Pg. 3 Line 48: Not sure what “default size 1 Kbp” is referring to. The flanking region? Need to be a bit more clear here.*

The sentence now reads "Reads that map to each annotated rDNA and its flanking regions (where the flanking regions consist of 1kb upstream and 1kb downstream of the rDNA, by default) are ..."

#### 6. *Pg. 3 Line 51: Why use the reference rDNA as trusted contigs? Did you see beter results when testing with/without this method? It’s still unclear to me how the trusted contigs are utilized by SPAdes. I assume k-mers are generated by the trusted contigs and just thrown in the pool of available k-mers during de novo assembly. Interestingly the SPAdes manual says not to use contigs of related species: “--trusted-contigs: Reliable contigs of the same genome, which are likely to have no misassemblies and small rate of other errors (e.g. mismatches and indels). This option is not intended for contigs of the related species.” I'm not entirely certain why they have this disclaimer in the manual. I’d like to see how well riboSeed performs without the trusted contigs.*

We address this in General Comments #5 above.

#### 7. *Pg. 3 Line 55: This isn’t particularly important, but why a 5 Kbp stretch of N’s? Seems excessive. Wouldn’t a 500 bp stretch of N’s suffice? Also, be consistent, to this point you typically write “kbp” but here you write “kb”.*

We have changed the spacer length to 1kb. The reviewer’s comment is noted, but the authors feel that using 1kb while being in effect no different from a 500bp spacer, helps readers understand purpose of the spacers to be exceedingly longer than spannable by a read from illumina or other short read technology. Further, wikipedia seems to prefer the suffix "kb", and the manuscript has been changed to reflect that throughout.

#### 8. *Pg. 3 Line 52: Go ahead and specify “BLASTn” here.*

Corrected.

#### 9*. Fig. 2: In part (b) what are “scannedScaffolds” in the title of the plot? I think you want to have more precise titles for both of these plots. Or just drop them and let the (a) and (b) labels explain what they are.*

The titles have been dropped for better readability, leaving the labels to explain.

#### 10. Pg. 4 Line 60: “softare” -> “software”

Corrected.

#### 11. *Pg. 6 Line 20: “adoped” -> “adopted”*

Corrected.

#### 12. *Fig. 5: What is the y-axis? Number of rDNA’s assembled? Need to label this better. Also the y-axis limits are different between (a) and (b). Both should be ylim=c(0,7).*

We have added a clarification to the legend, noting the differing numbers of rDNAs in the respective genomes of E coli and K. pneumoniae. The y-axes were not modified to avoid misrepresenting the two genomes as containing the same number of rDNAs.

#### 13. *Table 3: What does “skipped assembly” mean? Also, I would recommend re-configuring this table (a table with only one row isn’t much of a table). Maybe row one is de novo and row two is de fere novo?*

We have tried to clarify on our usage of the word "skipped". The reviewer was right to note that we were not consistent in our language, as we used "failure" and "skipped" to mean the same thing (an rDNA that was neither successfully resolved not incorrectly assembled). We have revised the "Validating Assembly across rDNA regions" section, changing all instances to to word “unassembled”.

As for the table structure, we agree that having single-row tables is not optimal. In order to keep the structure uniform between the three experiments (UAMS-1, hybrid assembly, and benchmarking with GAGE-B), we have combined all three tables into a single table with subsections for each of the experiments.

#### 14. *Table 4: You display the SNP type in the table, but don’t discuss the significance of them. Is it worth reporting?*

We do not believe the actual nature of the substitutions to be of significance, and report them only for completeness. The main point we wish to make is that the number of SNPs (9) is very low compared to 108 SNPs between the reference and the true sequence in the rDNA regions. If the detail of each SNP is distracting, we are open to removing it.

#### 15. *Pg. 7 Line 47: What is USA200?*

We added the word "lineage" to clarify; while not necessarily a common practice , *S. aureus* researchers frequently discuss strains by means of their lineage, similar to, say, the *E. coli* Clermont phylotypes.

#### 16. *Pg. 7 Line 27: How low for GC content and why? If low-GC content is a challenge, wouldn’t high-GC content be a challenge as well because of the low-complexity?*

The issues relating to GC content occur both prior to (or during) sequencing (namely PCR bias and cluster formation on the flowcell (<https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3188800/>

)), and at the data analysis stage (<http://journals.plos.org/plosone/article?id=10.1371/journal.pone.0062856>). The issue with low GC samples isn't solely the actual low complexity of the sequence, but also the "noisiness" of the data generated from those samples. The reviewer is correct that regions of low complexity, regardless of base composition, are problematic.

We anticipate that low-complexity (e.g. low GC) flanking regions and tandem rDNA repeats may pose problems for correct assembly using riboSeed, but we have not further explored performance on such examples past those encountered in the GAGE-B dataset. We observe that riboSeed does not perform as well with low GC genomes (*Staphylococcus*, *Campylobacter*) but we have not acquired enough data to confirm that this is a general property of low-GC organisms.

#### 17. *Table 5: Would recommend re-configuring this table as well.*

See response above

#### 18. *Pg. 8 Line 22: “where the rDNA regions to act as […]” -> “where the rDNA regions act as […]”*

Corrected