**Point-for-point response to reviewers for manuscript number MBIO-D-18-00153**

**Title**: "MetaWRAP - a flexible pipeline for genome-resolved metagenomic data analysis"

**Authors**: G. Uritskiy, J. DiRuggiero, and J. Taylor

\*\*We are grateful to the referees for their time in reviewing our manuscript and for their positive assessment. We appreciate their insightful comments and suggestions on improvements to the work. We have addressed all of their specific comments below and in the submitted manuscript. Line numbers presented below refer to track changes in the “**Marked Up Manuscript File**” document.

**Reviewer #1 (Comments for the Authors):**

* *I have only one comment. Please describe in more details the datasets used in the paper. As example, which are the characteristics of the three datasets (water, gut, soil) shown in figures 3 and 7?*
  + This is an important point, as the original MS did not give give sufficient background on these data sets. To address this, we introduced a new section early in the MS called “Benchmarking metaWRAP in real data”, where we describe in more detail the source of the metagenomic datasets, as well as what type of community they represent. [Lines 279-296]

**Reviewer #2: (Comments for the Authors):**

* *One key issue really requires more discussion in the manuscript. The background to this is that metagenomic "assembly" is basically almost never assembly in the true sense, but clustering of similar sequences which may represent several closely-related organisms. Indeed, the authors address this point when contrasting the approaches of 3rd-party tools: the more inclusive "collapsing" of similar bins into more complete, but less pure genomes; versus the more exclusive "purity" prioritization (lines 88-96).*
  + Indeed, the contigs resulting from a metagenomic assembly are the averaged sequences from closely related species/strains. This remains to be one of the big challenges in metagenomics. This is also an issue at the binning level, where contigs from varying levels of relatedness may or may not be binned together depending on the stringency of the parameters. Unfortunately, it is out of the scope of this project to address this at the assembly level, however we can indirectly evaluate the “collapsing” affects at the binning level by looking closer at the CAMI dataset.
* *Is it possible to comment on this given that one of the benchmarking exercises used various CAMI gold-standard assemblies?* 
  + As was pointed out, the CAMI challenge offers a unique opportunity to explore this in more detail by evaluating the binning performance of metaWRAP genomes that are closely or distantly related to each other. The results suggested that metaWRAP significantly outperformed other binning approaches when binning closely-related genomes, suggesting that it overcomes the “collapsing” issue to a better degree. The resulting analysis is addressed in the MS [lines 262-271] and show in in a new supplementary table (Additional file 14).
* *The important question therefore from the reader/user, which is not really mentioned at all is: what do the resulting genomes represent? Species, strains?*
  + Unfortunately, it is impossible to answer this question without knowing more about the specifics of the dataset in question. This is left intentionally ambiguous in the MS, as metaWRAP is meant to be a broadly applicable data processing tool. The interpretation of the data will ultimately be up to the biologists. Furthermore, there is currently a lot of ambiguity about the specific definition of species and strains (especially in the metagenomic context), and discussing this is out of the scope of this work.
* *The authors perform a thorough analysis of changes in 2 parameters (relating to completeness as measured by a 3rd-party tool CheckM, contamination) when performing the bin refinement; and emphasize that the user can set their own thresholds for what they consider to be a good genome thus obtained. But how should the user view these metrics in terms of biological/phylotypic meaning - how should the user view say, a contamination metric of zero or near-zero; would that give the expectation that for instance different bacterial strains would be separated out? Intra-species differences can be critical, so could a user reasonably expect these to be resolved with correct specification of parameters?*
  + This is an important point, and one that was not directly addressed in the MS. The desired maximum contamination and minimum completion thresholding will vary greatly depending context of the analysis and the down-stream applications of the extracted bins. Because we cannot anticipate the applications of metaWRAP, these parameters are left up user. We added this clarification [lines 357-359].
* *Perhaps related to (1) is the question of how the bin\_refinement module, which appears to perform impressively, actually works. It seems to rest on the "hybridization" of results (bin) produced by combinations of 2 or 3 of the 3rd-party binning tools, but I don't think this hybridization is explained well, either in the manuscript or in Additional File 1. The manuscript especially (lines 148-150 and 239-241) is confusing. Again this makes it difficult for the user to have a detailed perspective of what their results actually mean.*
  + Indeed, we did not properly explain how the Bin\_refiner and Reassemble\_bins modules work, especially in the main MS. We added two sections under “Implementation” in the MS [lines 143-167], describing how these two algorithms work.
* *On a similar note, the description of one of the 3rd-party refiners is not all that clear (lines 91-92).*
  + We provided a more detailed and more clear explanation on how DAS\_Tool and Binning\_refiner work [lines 96-107].
* *There are some details which I feel should be included in the main manuscript. It was unclear why one assembly tool (MegaHIT) was used for the initial, full-metagenome assembly, but a different one (meta-SPAdes) used for the reassembly of single-bin reads. I realize that this is stated in the supplementary material but I don't think it's in the main text, and without it it looks like an arbitrary choice.* 
  + The individual bin reassembly is actually done with SPAdes – a non-metagenomic assembler. This is now better explained in the “Implementation” section [lines 161-163].
* *Also, given that a lot of the benchmarking rests on assessment of genome completeness by CheckM, it may be worth briefly indicating how that metric is calculated, as it's not from complete reference genomes but a rather small number of marker genes as far as I'm aware (this does not detract from its use).*
  + To clarify this, we described the concept behind the CheckM algorithm in more detail in the Introduction [lines 84-91].
* *It is unclear why Kraken is used in the early step. Kraken is designed for speed, using k-mer frequencies to identify taxa, and unsurprisingly, a relatively high frequency of false-positives can result in my experience. At worst, Kraken output could detract and distract from the much more reliable results obtained by the assembly/refinement followed by taxon-assignment by Taxator-Tk. At the least, should some sort of health-warning about the results in the Kraken-derived krona-plot be included?*
* We agree on the limitations of Kraken. However, because metaWRAP does not place much emphasis in taxonomic profiling, the Kraken module is meant to be a quick pass on the data to get an approximate idea of what the community is comprised of. To address this limitation, we direct the readers to Bracken – a Kraken post-classification standardization software that would allow for more accurate taxonomic composition estimation [lines 190-192].
* *Figure 3 shows the bins colored by Phylum. However Lines 220-222 describe lower-level ranks (and Phyla); Clostridia and Alphaproteobacteria are not depicted on Fig 3.*
* This was a mistake in the text descriptions, and fixed in lines 293-296.
* *A positive major comment is that there is a large amount of documentation, scripts and tutorials at the various github URLs. In my opinion this is at a much more detailed level than is often seen, and I thank the authors. This includes some scripts for producing the figures in the paper. However, I could find the scripts for creating the plots, but not for the analyses themselves, e.g. CAMI benchmarking. Could the authors confirm if these are available?* 
  + These commands are now available under the metawrap\_paper github directory <https://github.com/bxlab/metawrap_paper> in “running\_metawrap\_on\_cami\_data.sh” and “running\_metawrap\_on\_real\_data.sh”.
* *Relating to this, it is unclear if all of the parameters used for 3rd-party tools are stated (just as one example - use of BWA to extract single bin-specific reads, line 403).*
  + We admit that this information was scattered between the Implementation section, Methods section, and Additional\_file\_1, making it confusing to find. Because metaWRAP uses dozens of third-party software, we felt it would be uncessessarily verbose to include all the details in the main MS. Instead we re-wrote the Methods section [lines 489-635] (and added the Implementation sections [lines 125-168]) to include only surface and conceptual detail about what the different software used, while explicitly referring to the Supplementary Methods and and the script files in <https://github.com/bxlab/metawrap_paper/> for extra details. We still kept the parameterization details for software not described in the metaWRAP module descriptions. The Supplementary methods (Additional\_file\_1) now contains greater degree of detail about the software used by metaWRAP, their versions, and their parameterization.
* *Line 68: MetaWRAP may score highly for convenience of use, but it's not necessary to use the existing tools on a cluster, and this statement might be taken to imply that MetaWRAP is saving the bother of doing this or saving compute time. It's perfectly feasible to assemble hundreds of 3rd-party metagenomics samples with existing tools on a single Linux server in a tractable time, and the compute times the authors report (lines 161 etc) when running their own software on a Linux server are in line with this.*
  + This was a miscommunication. We removed the words “on a cluster” to keep it more inclusive [line 71].
* *68: This should be "environment variables" I think.*
  + This was a typo. Fixed [line 71].
* *89: first word "bin" should be pluralized.*
  + This was a typo. This section was re-written [lines 86-89].
* *119: Strictly should be Bioconda, not Anaconda?*
  + At this time, metaWRAP is not part of Bioconda (which is a popular channel for Anaconda packages). So it only distributed through the Anaconda cloud system (https://anaconda.org/ursky/metawrap-mg).
* *139: "produced" should be "produces"*
  + Typo. Fixed [line 183].
* *288: "water and bin set" should be "water and gut bin set"?*
  + Typo. Fixed [line 401].
* *452-454: states NCBI but the URL is for the EBI equivalent for these SRA data*
  + This was a mistake. Fixed with proper NCBI link [line 762].
* *454: the URL is incorrect as it includes a session-specific tag at the end.*
  + No longer applies, as it was replaced with the proper NCBI linke[line 762].
* *The instructions for installing and configuring the prerequisite databases are thorough. I noticed one important typo at this URL* [*https://github.com/bxlab/metaWRAP/blob/master/installation/database\_installation.md*](https://github.com/bxlab/metaWRAP/blob/master/installation/database_installation.md) *specify the BMTAGGER\_DB variable in the contig-metawrap file" - should be "config-metawrap", the name of the file the user changes.*
  + - * The database instruction have been updated to include this fix.
* *I managed to follow most steps of the tutorial, but unfortunately time constraints have prevented me from following this through to the end. I would like to note one issue, which was that after installation I ended up with 2 config-metawrap files, one in the bin/ directory of the metaWRAP installation, and another under Miniconda (~/miniconda2/bin/config-metawrap) - and it was the latter which needed editing to take effect, not the former, which caused some confusion at first.*
  + *The only way I can see this happening is if metaWRAP was installed twice: once manually and once with conda. Regardless, the config-metawrap file that matters is the one that has priority in the PATH, i.e the one that comes up when the user runs “which metawrap”. This is emphasized in the installation instructions.*
* *Reference 20 looks incorrect - it appears unrelated to the manuscript in general and to its cited context in particular.*
  + This was a mistake. This reference was removed.
* *References 14, 28, 34, 40: Please check the journal guidelines for a definitive view, but I think inclusion of a DOI for these preprint /unreviewed publications is preferable.*
  + All BioRxiv references now have a DOI, and all software references without a proper publication now have a URL.
* *Reference 45: a URL could be useful here.*
  + Fixed.