**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 and we agree that the original MS did not provide 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.
  + Excerpt 1 [lines 252-263]: “The water data set was from a brackish water survey, which investigated the seasonal dynamics and biogeography of the surface bacterioplankton in the Baltic Sea [42]. This dataset included 36 samples for a total of 196Gbp of sequencing data. The gut data set came from the Metagenomic of the Human Intestinal Tract (MetaHIT) survey, which sequenced the gut microbiota from volunteers across Europe to explore the diversity and drivers in individual gut microbiome composition. [40]. The benchmarking dataset consisted of 50 samples for a total of 144Gbp of sequencing data. The soil data came from sequencing the highly diverse grassland soil microbial communities from Angelo Coastal Reserve, CA [27]. This dataset consisted of 6 samples for a total of 481Gbp of sequencing data.”
  + Excerpt 2 [lines 269-274]: “Notably, contigs from the soil microbiomes had much higher GC content compared to those of gut and water. Also, soil contigs did not form as many defined clusters on the GC vs Abundance plot, suggesting that the communities were comprised of multiple closely related taxa (Figure 3). Due to the high GC content and high taxonomic similarity of soil microbiota, this dataset posed significant binning challenges compared to the water and gut microbiomes.”

**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.*
  + Thank you for raising this important issue, which we failed to discuss in appropriate detail in the initial submission. Indeed, contigs resulting from a metagenomic assembly are averaged sequences from closely related species/strains. This remains one of the big challenges in metagenomics. We have added a section in the MS to clarify and discuss the underlying assumptions behind metagenomic assembly, as well as the relationship between bins and genomes [lines 362-392].
  + Excerpt 1 [lines 331-344]: “It is important to note that while refinement of binning predictions results in high quality bins when evaluated with single-copy gene numbers, they do not represent the genomes of single individuals in a community, or even individual strains. In this context, a bin is simply the optimized taxonomic clustering of contigs, which themselves are representative consensus resulting from the clustering of reads belonging to closely related taxa. In the context of phylogeny, bins may represent individual strains, species, or even higher-order averaged taxa, depending on the level of heterogeneity of the community in question. In the literature, bins are sometimes referred to as population genomes [43], underlying the complex nature of bins. As described in the context of the CAMI challenge, the analysis of a community with mostly “unique strains”, i.e. distantly related organisms, will result in bins potentially representing species or even strains, whereas the analysis of a community with mostly ‘common strains’, i.e. closely related organisms, will results in more hybrid bins. In reality, most communities are an assemblage of both closely and distantly related taxa resulting in a range of bin qualities.”
  + Excerpt 2 [lines 376-379]: “Just as with the binning process, it is important to note that the bins resulting from the reassembly do not represent the true genomes of individual organisms found in the community, but are rather consensus backbones for reads coming from 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). 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 point 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 result of this analysis is addressed in the MS [lines 262-271] and shown in in a new supplementary table (Additional file 14).
  + Excerpt [lines 234-244]: “The CAMI challenge consisted of genomes of varying degree of similarity, and categorized the genomes into two broad categories depending on their average nucleotide identity (ANI) to other genomes in the mix. “Unique strains” are defined as genomes with <95% ANI to any other genome and “common strains” as genomes with ≥95% ANI to another genome in the data set. [9] This gave us an opportunity to benchmark metaWRAP at recovering genomes from contig clusters of varying complexity. We found that metaWRAP outperformed all other binning methods in reconstituting both closely and distantly related genomes (Additional file 14: CAMI binning summary table). Interestingly, we found that Binning\_refiner performed almost as well as metaWRAP in distantly related genomes, but performed poorly in closely related genomes. On the other hand, DAS\_Tool recovered almost as many closely related genomes as metaWRAP, but performed relatively poorly in more discrete genomes.”
* *The important question therefore from the reader/user, which is not really mentioned at all is: what do the resulting genomes represent? Species, strains?*
  + This is an important point that we addressed in our answers above by describing the processes of metagenome assembly and binning and how at each step, the resulting products, i.e. contigs or bins, are consensus of sequences or genomes.
  + Excerpt [lines 335-344]: “In the context of phylogeny, bins may represent individual strains, species, or even higher-order averaged taxa, depending on the level of heterogeneity of the community in question. In the literature, bins are sometimes referred to as population genomes [43], underlying the complex nature of bins. As described in the context of the CAMI challenge, the analysis of a community with mostly “unique strains”, i.e. distantly related organisms, will result in bins potentially representing species or even strains, whereas the analysis of a community with mostly “common strains”, i.e. closely related organisms, will results in more hybrid bins. In reality, most communities are an assemblage of both closely and distantly related taxa resulting in a range of bin qualities.”

* *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. We added further discussion at the end of the section to give the readers a better understanding about viable parameterization of metaWRAP depending on context and the down-stream applications of the extracted bins and the microbiome in question.
  + Excerpt [lines 345-356]: “Because of this, contamination resulting from strain heterogeneity is expected [44], and the desired bin quality can be tailored to the requirements of the down-stream applications. For accurate taxonomic assignment of bins, a low contamination is important (1-5%), but a high completion may not be (20-50% may be sufficient). For accurate reconstruction of metabolic potential on the other hand, it is more important to reconstruct the genome with a higher completion (90-98%), even at the expense of introducing contamination (5-10%), as long as the user understands that the resulting bins represent the averaging of closely-related taxa. The parameterization will also be constrained by the characteristics of the microbiome in question. Communities with relatively low diversity, low strain heterogeneity, and low GC content (such as gut microbiomes) will yield bins with lower contamination and higher completion than those extracted with from a community with high diversity, heterogeneity, and average GC content (such as soil microbiomes).”
* *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.*
  + We agree and added two sections under “Implementation” in the MS [lines 143-167] to properly explain how the Bin\_refiner and Reassemble\_bins modules work.
  + Excerpt 1 [lines 134-144]: “The metaWRAP-Bin\_refinement module produces a superior bin set from multiple original binning predictions. First, hybrid bin sets are produced with Binning\_refiner [28], which splits the contigs such that no two contigs are together if they were in different bins in any of the original sets. Then the module goes over the different variants of each bin found in the original and hybridized bin sets, and choses its best version based on completion and contamination metrics estimated with CheckM [24]. The decision of the “best bin” is based on the user-provided minimum completion and maximum contamination parameters. The contigs in the final bin set are then de-replicated, and a report of their completion, contamination, and other metrics is produced (Additional file 3: Figure S2). See Supplementary Methods (Additional file 1) for more details on the Bin\_refinement module.”
  + Excerpt 2 [lines 146-157]: “The metaWRAP-Reassemble\_bins module improves a set of bins by individually re-assembling each bin (Additional file 4: Figure S3). Reads are mapped to the bins with BWA v0.7.15 [32] strictly (no mismatches) and permissively (<5 mismatches) and stored into their respective FastQ files. Importantly, read pairs will be pulled out even if only one read aligned to the bin. Each read set is then reassembled with SPAdes [34], which produces more contiguous sequences compared to metagenomic assemblers such as MegaHit [35] and metaSPAdes [36] used in the Assembly modules. CheckM [24] is used to evaluate the completion and contamination of each of the three versions of each bin – the original bin, the “strict” re-assembled bin, and “permissive” re-assembled bin and the best version of each bin is chosen for the final bin set based on the user-defined desired bin quality. See Supplementary Methods (Additional file 1) for more details on the Reassemble\_bins module.”
* *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 clear explanation on how DAS\_Tool and Binning\_refiner work.
  + Excerpt [lines 91-98]: “DAS\_Tool predicts single-copy genes in all the provided bin sets, aggregates bins from different binning predictions, and extracts a more complete consensus bin from each aggregate such that the resulting bin has the most single-copy genes while having a reasonably low number of duplicate genes [27]. This collapsing approach significantly improves the completion of the bins. Binning\_refiner, on the other hand, splits the contigs into more bins such that no two contigs are in the same bin if they were in different bins in any of the original bin sets. This breaks the contigs into many more bins, reducing contamination[28].”
* *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 was actually done with SPAdes – a non-metagenomic assembler. This is now better explained in the “Implementation” section of the MS.
  + Excerpt [lines 150-153]: “Each read set is then reassembled with SPAdes [34], which produces more contiguous sequences compared to metagenomic assemblers such as MegaHit [35] and metaSPAdes [36] used in the Assembly modules.”
* *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.
  + Excerpt [lines 81-86]: “These metrics can be estimated by counting universal single-copy genes within each bin [22, 23]. CheckM improves on this by checking for single-copy genes that a genome of the bin’s taxonomy is expected to have [24]. The percentage of expected single-copy genes that are found in a bin is interpreted as its completion, while the contamination is estimated from the percentage of single-copy genes that are found in duplicate.”
* *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 and find it a useful and fast tool for a first pass of the data in order to give the user an approximate idea of the composition of the community. For more accurate taxonomic composition estimates, we have added a section redirecting the reader to Bracken – a Kraken post-classification standardization software.
* Excerpt [lines 175-177]: “It should be noted that while KRAKEN is fast, post-classification standardization may be needed to obtain a more accurate community composition estimate [38].”
* *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 and it is now fixed.
* Excerpt [lines 266-269]: “The water samples were dominated by *Proteobacteria*, the gut samples were dominated by *Bacteroidetes* and *Firmicutes*, and the soil samples comprised of a wide variety of *Proteobacteria* and *Actinobacteria* (Additional file 8)”
* *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 difficult to find. Because metaWRAP uses dozens of third-party software, we felt it would be unnecessarily verbose to include all the details in the main MS. Instead, we re-wrote the Methods section [lines 447-521] and added the Implementation sections [lines 134-157] to include conceptual detail about what the different software used while explicitly referring to the Supplementary Methods and the script files in <https://github.com/bxlab/metawrap_paper/> for details. The Supplementary methods (Additional file 1) now contain a 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 is a good and justified point and we removed the words “on a cluster” [line 68].
* *68: This should be "environment variables" I think.*
  + This was a typo. Fixed [line 68].
* *89: first word "bin" should be pluralized.*
  + This was a typo. This section was re-written [lines 81-86].
* *119: Strictly should be Bioconda, not Anaconda?*
  + At this time, metaWRAP is not part of Bioconda (which is a popular channel for Anaconda packages). It is only distributed through the Anaconda cloud system (https://anaconda.org/ursky/metawrap-mg).
* *139: "produced" should be "produces"*
  + Typo. Fixed [line 171].
* *288: "water and bin set" should be "water and gut bin set"?*
  + Typo. Fixed [line 368].
* *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.
  + Excerpt [line 546]: “National Centre for Biotechnology Information under SRA numbers SRR2053273–SRR2053308 ([https://www.ncbi.nlm.nih.gov/bioproject/PRJNA273799)](https://www.ebi.ac.uk/metagenomics/projects/SRP058493;jsessionid=4DFCCEDFE404206D16E02ECC78D63F8A)) for the Central Baltic Surface Water Metagenome.”
* *454: the URL is incorrect as it includes a session-specific tag at the end.*
  + This no longer applies and was replaced with the proper NCBI link [line 546].
* *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 instructions 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 this could happen 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.