MetaWRAP - a modular pipeline for whole genome metagenomic data analysis and draft genome extraction

ABSTRACT

**Background**

De-convoluting metagenomic data by extracting high-quality draft genomes allows for analysis of uncultivated microbial populations that may have important roles in their microbiomes. As software and pipelines for such analysis are becoming more diverse and sophisticated, it is also becoming increasingly burdensome for biologists to access and use. To further complicate matters, the performance of various approaches varies between microbial communities.

**Results**

To address some of these challenges in present-day metagenomics, we created MetaWRAP, which is an easy-to-install and easy-to-use modular pipeline software that deploys state-of-the-art software and handles common tasks in metagenomic data processing starting from raw sequencing reads, and ending in metagenomic bins and their analysis. In addition to being a tool wrapper, metaWRAP also offers a powerful hybrid approach for extracting high-quality bins by using a variety of binning software and utilizing their individual strengths and minimizing their weaknesses. This approach outperforms not only individual binning approaches, but also other bin consolidation programs in both synthetic and real datasets. MetaWRAP also comes with a bin reassembly algorithm, which further improves the completeness and purity of the draft genomes. Finally, metaWRAP has numerous components that are dedicated to the analysis of metagenomic bins, including taxonomy assignment, abundance estimation, functional annotation, and visualization.

**Conclusions**

We present metaWRAP - an easy-to-use modular pipeline software that accomplishes the core tasks in metagenomic analysis, while also contributing significant improvements to the extraction of high-quality metagenomic bins. The bin refinement and reassembly modules of metaWRAP consistently outperform other currently available binning approaches. Each module of metaWRAP is also a standalone component, making it a flexible and versatile tool for tackling whole metagenomic sequencing data.

BACKGROUND

The study of microbial communities through whole metagenome (WMG) sequencing opens new opportunities for the study of not only the taxonomic composition of microbiomes, but also their metabolic potential. However, because all the sequencing reads from different community members are mixed together, this poses a unique and difficult challenge for analysis and interpretation of such sequencing data. As the number of sequenced metagenomes rapidly grows, the need to more sophisticated analysis software readily available for microbiologists is becoming obvious.

There are now thousands of available software to analyze metagenomic data, and this number is rapidly growing. In order to perform meaningful analysis of metagenomic data, microbiologists have to use dozens of software and in-house scripts to take the raw sequencing reads through quality trimming, de-contamination, assembly, taxonomic profiling, binning, functional annotation, and many other functions that they may be interested in. The establishment of a such a pipeline is a difficult task for a variety of reasons. First, the investigator needs to research the best currently available tool for their task. Second, the software needs to be installed and configured, which is a major challenge faced even by bioinformaticians. Additionally, conflicting libraries and environmental variables need to be addressed. Finally, running each software and script one after another is a lot of work. Each output needs to be converted into the correct format to the input for the next step, and errors need to be detected, identified, and handled.

Together, these challenges present a major burden on anyone attempting metagenomic analysis, especially for microbial biologists without significant computational experience. This prevents many biologists from attempting this on their own, forcing them to outsource sequencing data analysis or perform only surface-level inspection of the data. The barriers preventing biologists from properly utilizing the wealth of available algorithmic and computational power significantly slows the progress of microbiology as a field. With this in mind, metaWRAP was created in order to make metagenomic analysis easily accessible.

One aspect of metagenomic analysis is de-convoluting assembled WMGs by extracting the single genomes of its community members through a process is called binning. Genome-resolved metagenomics allows for inspection of the metabolic pathways and abundance shifts of individual organisms, increasing the resolution of metagenomic analysis. Metagenomic binning poses a great computational challenge, as the goal is to predict scaffolds belonging to the same organisms based on abundance and sequence qualities alone. Many sophisticated tools such as CONCOCT, MaxBin, and metaBAT have been developed to tackle this problem, each one using different approaches to make the bin predictions.

Most good metagenomic binning tools rely on a combination of qualities of the contigs to cluster them together. One such quality which was used in early binning attempts is, is the k-mer composition and codon usage properties of scaffolds, which are assumed to be similar throughout the genome of a given organism. Another useful assumption that most binners make is that scaffolds from the same organism are expected to have similar read coverages in any given sample. This concept greatly improves bin prediction by clustering together scaffolds that have similar abundances across multiple samples. Finally, as the number of reference microbial sequences grows, some tools also incorporate the phylogenetic information that can be derived from sequences and their genes.

Because they use a variety of approaches, there is no single binning software that will extract the best version of each bin in every case. To combine the strengths and minimize weaknesses of different binning software, a couple bin consolidation software were created. DAS\_Tool predicts single-copy genes in all the provided bin, aggregates bins with overlapping genes, and extracts a more complete consensus bin from each aggregate. This aggressive bin collapsing approach significantly improves the completion of the bin predictions with minimal increase of contamination. Binning\_refiner takes the opposite approach – it splits the contigs into bins such that all the contig division boundaries of the original bin predictions are satisfied. This breaks up the contigs into many more bins, reducing their completion, but it also reduces their contamination. While taking opposite approaches consolidate sets of bins from different software and result in a superior bin set, they have limitations – DAS\_Tool focuses on completion at expense of introducing contamination, while Binning\_refiner prioritizes purity, but loses completeness. This problem inspired the creation of the meraWRAP-Bin\_refinement module, which is able to reduce contamination and improve completion of bin sets by having both a splitting and a collapsing step.

A relatively unexplored way to improve bin quality is bin reassembly – extracting reads that belong to a given bin and assembling them separately from the rest of the metagenome. This idea was featured in MaxBin, but it was highly experimental. With proper benchmarking and reassembly evaluation this approach has promise to significantly improve at least some bins in a microbial community, therefore improve the quality of the genome’s functional annotation. This idea lead to the creation of the metaWRAP-Reassemble\_bins module.

Another challenge of metagenomic bin prediction is evaluating the quality of the resulting bins. In order for a metagenomic bin to be considered the genome of a single organism, it must cover a significant length of the true genome (have a high completion), and also not have sequences belonging to other organisms (have a low contamination). The completion and contamination of a bin can be estimated by finding and counting universal single-copy genes that they have. A perfect bin is expected to have all such genes, but each occurring only once. CheckM is a powerful tool that takes this idea to the next level, inspecting not only the universal single-copy genes of all domains of life, but the single-copy genes that a genomes of a particular taxonomy is expected to have.

Because the field is relatively new, there is a lack of software to inspect, analyze, and visualize metagenomic bins. While there are many softwares that can accurately predict the taxonomy of metagenomic scaffolds (such as Taxator-tk), there is no software to classify entire metagenomic bins. Similarly, there are many ways to estimate the coverage of scaffolds based on read alignment depth, but no way to find the coverages of entire bins across many samples. Finally, there is a lack of software to visualize bin composition and distribution in metagenomic communities. These knowledge gaps inspired the creation of metaWRAP’s Quant\_bins, Classify\_bins, and Blobology modules.

RESULTS AND DISCUSSION

**MetaWRAP is a flexible, modular pipeline**

MetaWRAP is a modular, comprehensive platform for analysis, visualization, and interpretation of metagenomic data, with emphasis on extracting and analyzing high-quality draft genomes (bins). With the help of Anaconda, metaWRAP is easy to download and install for biologists without significant computation experience. The metaWRAP installation produces a bioinformatics environment with over 150 commonly used bioinformatics software and libraries, saving the user from installing and configuring them individually (Figure S1). MetaWRAP itself is a collection of modules, each of which uses a variety of pre-existing software, custom scripts, and databases to accomplish a major goal in metagenomic analysis. The user may follow the intuitive workflow starting from raw metagenomic sequencing reads all the way to high-quality draft genomes and their functional annotation. However, each module is a standalone program, giving the user control of the pipeline. (Figure 1).

The metaWRAP workflow starts with modules that conveniently wrap common software to quality-control and taxonomically profile the reads, assemble them, and then bin the assembly. MetaWRAP-Read\_qc module trims the raw sequence reads and removes human contamination from each of the sequenced samples. Quality control reports are also generated to evaluate the sequencing quality. The reads from all given samples are then assembled with the metaWRAP-Assembly module, also producing an assembly report. Both the reads from each sample and the assembly can be taxonomically profiled with the Kraken module, producing interactive kronagrams of community taxonomy. The metaWRAP-Binning module is then used to bin the scaffolds of the joint assembly using three metagenomic binning software – MaxBin2, metaBAT2, and CONCOCT.

The rest of metaWRAP’s modules focus on refining and analyzing metagenomic bins. It is important to note that the user can provide their own sets of bins produced with software besides the ones available in metaWRAP-Binning. Also, none of the modules are dependant on each other, so the user is free to provide their own bin sets for each module. MetaWRAP-Bin\_refinement module hybridizes up to three bin sets with Binning\_refiner, and then finds the best version of each bin based on completion and contamination metrics estimated with CheckM (Figure S2). The scaffolds in the final bin set is then de-replicated, and a report of their completion, contamination, and other metrics is produced. MetaWRAP-Reassemble\_bins can then be used to reassemble the reads belonging to each bin, improving their N50, completion, and contamination (Figure S3).

The resulting bins can then be visualized by using the metaWRAP-Blobology module, which plots the contigs of the joint assembly on a blob plot, and annotating them with their taxonomy and bin membership. The metaWRAP-Quant\_bins module can be used to quickly estimate the abundance of each bin in each of the metagenomic samples. MetaWRAP-Classify\_bins can be used to conservatively, but accurately estimate their taxonomy. Finally, the bins can be functionally annotated with the metaWRAP-Annotate\_bins module.

**MetaWRAP-Bin\_refinement improved bin predictions in synthetic data**

In order to objectively compare the performance of different analysis approaches, synthetic metagenomic data sets of varying complexity were created in the Critical Assessment of Metagenomic Interpretation (CAMI) study. In order to test the efficacy of the metaWRAP-Bin\_refinement module in improving bins based on three different bin sets, we tested applied the module to CAMI’s binning challenge, which consisted of synthetic metagenomic assemblies that needed to be binned to reproduce the original known genomes.

The “gold standard” assemblies from the “high”, “medium”, and “low” diversity challenges were binned with metaBAT2, Maxbin2, and CONCOCT using the metaWRAP-Binning module. The resulting three bin sets were then consolidated with DAS\_Tool, Binning\_refiner, and metaWRAP-Bin\_refinement to attempt to improve the bin sets. The completion and contamination of the resulting bins were first evaluated with CheckM (Figure S4), and then the true values were determined by comparing to the true genomes and the recall and precision of each bin was calculated with Amber (Figure S5). Bin recall and precision were converted to completion and contamination percentages to be comparable to the CheckM results (Figure 2).

Between the original binning software, metaBAT2 consistently outperformed MaxBin2 and CONCOCT, producing a total of 385 high quality bins between all the challenges (completion greater than 90% and contamination less than 5%), and 271 near-perfect bins (completion greater than 95% and contamination less than 1%). MaxBin2 came in second with 275 high quality bins and 164 near-perfect bins. Finally, CONCOCT performed rather poorly in all but the smallest CAMI challenge data sets, producing 58 high quality bins and 40 near-perfect bins.

In the consolidated bin sets, DAS\_Tool consistently produced high-completion bins in all CAMI challenges. Between all three challenges, DAS\_Tool was able to produce 426 high quality bins and 263 near-perfect bins. However, these bins had relatively high contamination when compared to other bin refiners - Binning\_refiner and metaWRAP-Bin\_refinement, which stems from the aggregation approach that DAS\_Tool takes. Binning\_refiner on the other hand produced very pure bins with its aggressive splitting approach, however it did so at the expense of significantly reduced completion. In total, Binning\_refiner was able to produce 289 high quality bins and 210 near-perfect bins.

MetaWRAP-Bin\_refinent was able to consistently produce high completion and low contamination bins. In total, it produced 457 high quality bins and 339 near-perfect bins (Figures 2, S4). Because metaWRAP-Bin\_refinement has both a splitting and aggregation step, it was able to come close to DAS\_Tool in completion while at the same time coming close to Binning\_refiner in contamination. These results confirm that metaWRAP not only consistently improves bin sets through its consolidation approach, but it also outperforms other consolidation algorithms in data sets of varying complexity.

The use of CheckM (Figure S4) and Amber (Figure 2) to evaluate the binning sets produced similar results, although overall CheckM slightly overestimated both completion and contamination of the produced bins. More importantly, however, the relative performance of the six binning approaches was the same when evaluating with CheckM or Amber. This validates the use of CheckM for benchmarking binning results in data sets where the true genomes remain unknown.

**Bin\_refinement improves bin predictions in real data**

MetaWRAP was also benchmarked against real metagenomic data. To test its efficacy on a range of microbial communities, it was tested on a water, gut, and soil microbiome WGS metagenomic data sets. The water data sets came from a brackish water survey of the Baltic Sea, and includes 36 samples for a total of 196Gbp in sequencing data. The gut data set came from the Metagenomic of the Human Intestinal Tract (MetaHIT) survey, and consisted of 50 samples and a total of 144Gbp. Finally, the soil data set came from grassland soil microbial communities from Angelo Coastal Reserve, and included 6 samples for a total of 481Gbp of sequencing data.

The samples from each microbiome type were pre-processed through the metaWRAP-Read\_qc module to trim the reads and remove human contamination, and then the Kraken module of metaWRAP was run on the reads to investigate the taxonomic profile of the community (Figure S7). The water samples were dominated by Alphaproteobacteria and Actinobacteria, the gut samples were dominated by Bacteroidetes, and Clostridia, and the soil samples were comprised of a wide variety of Proteobacteria and Terrabacteria. (Figure 3)

The quality-controlled reads were then co-assembled with metaWRAP-Assembly module, and the assemblies were then binned with metaBAT2 Maxbin2, and CONCOCT using the metaWRAP-Binning module. The resulting three bin sets of each microbiome type were then passed to DAS\_Tool, Binning\_refiner, and metaWRAP-Bin\_refinement, and the completion and contamination of all the resulting bins were evaluated with CheckM. (Figure 3)

Between the original binning software, metaBAT2 consistently produced the best sets of bins when compared to MaxBin2 and CONCOCT, producing 202, 146, and 88 acceptable quality bins (comp ≥ 50%, cont ≤ 10%) in water, gut, and soil samples, respectively. MaxBin2 came out with 151, 98, and 40 bins, and CONCOCT with 65, 121, and 39 bins.

Despite incorporating all the binning methods, DAS\_Tool struggled to outperform metaBAT2 bins in all three data sets, producing 198, 130, and 63 acceptable quality bins in water, gut, and soil samples, respectively. DAS\_Tool still came out ahead at higher bin completion ranges (≥ 80%), although at the expense of increased contamination. Binning\_refiner also produced a similar number of bins in the acceptable quality range when compared to metaBAT2, with 206, 138, and 83 bins in water, gut, and soil data sets, respectively. The resulting bins were less complete in the gut data sets, however Binning\_refiner was able to significantly reduce the contamination of bins in all the data sets when compared to to the original bin sets.

MetaWRAP-Bin\_refinement was able to significantly improve the original bins of all three data sets, producing 235, 175, and 134 acceptable quality bins in water, gut, and soil samples. The Bin\_refinement module uses Binning\_refiner in its pipeline to hybridize the input bin sets, and then choses the best version of each bin from the original and hybridized sets. Because the module leverages the strengths of both DAS\_Tool and Binning\_refiner, it is able to match DAS\_Tool’s high completion rankings, while retaining the low contamination rankings of Binning\_refiner. MetaWRAP consistently produced the highest quality bin sets in all the tested metagenomic data sets, which ranged greatly in diversity, taxonomic composition, and sequencing depths.

It is important to note that the use of metaWRAP’s Bin\_refinement module to improve binning predictions is not limited to the bin sets produced from the metaWRAP-Binning module (metaBAT2, MaxBin2, and CONCOCT). Bin sets from any 2 or 3 binning software may be used as input for the module. Furthermore, because the algorithm leverages the differences between the input bin predictions, it is also possible to use bin sets produced from different parameters of the same software as input, and the metaWRAP-Bin\_refinement will consolidate them into a superior bin set.

**Bin\_refinement adjusts to the desired bin quality**

To consolidate the original and hybridized bin sets, metaWRAP-Bin\_refinement needs to choose the best version of each bin based on their completion and contamination values. However, this selection is rather subjective, and depends on what the user believes to be the “best bin”. One of the core inputs of the metaWRAP-Bin\_refinement module is the minimum completion (-c) and maximum contamination (-x) parameters, which signal to the program the prioritized bin quality range. Based on these inputs, metaWRAP will dynamically adjust its algorithms to produce the maximum number of bins in this range.

To demonstrate the effects of changing the –c and –x parameters of metaWRAP’s Bin\_refinement module, it was re-run on the original bin sets from water, gut, and soil data sets with varying minimum completion (but fixed maximum contamination), and varying maximum contamination (but fixed minimum completion) parameters. The number of extra bins that was gained at that quality threshold when compared to the original run (-c 50 –x 10) was noted. (Figures S6, S7)

When changing the minimum completion parameter of metaWRAP but keeping the maximum contamination at 10%, the number of bins recovered at each threshold was notably higher when compared to the original metaWRAP run (-c 50 –x 10), although the improvements in the soil bins were not as significant (Figure S5). The improvements were especially noticeable at higher completion ranges. MetaWRAP –c 90 –x 10 recovered 19, 18, and 1 (water, gut, and soil, respectively) extra bins with a minimum completion of 90%, when compared to metaWRAP –c 50 –x 10.

When changing the maximum contamination parameter of metaWRAP, but keeping the minimum completion at 50%, the number of bins recovered at each threshold was higher when compared to the original metaWRAP run (-c 50 –x 10) (Figure S6). Just as with varying -c, the improvements in the soil bins were not as significant. These improvements were especially significant at the lower contamination range. MetaWRAP with –c 50 –x 1 parameters extracted 8, 21, 4 (water, gut, and soil, respectively) more bins at a maximum contamination of 1%, when compared to metaWRAP –c 50 –x 10.

The minimum completion (-c) and maximum contamination (-x) options are key parameters that greatly alter the quality of the bins produced by metaWRAP-Bin\_refinement module. Unlike arbitrary and confusing thresholding parameters in many other software, the –c and –x options offer the user an easy and intuitive way to parameterize the program to their needs, and significantly increase the number of appropriate quality bins that they extract from their data. If the parameters are not entered, the default minimum completion is 70%, and default maximum contamination is 5%.

**Reassemble\_bins significantly improves bin quality**

MetaWRAP’s Reassemble\_bins module attepts to reassemble the provided bins, and only replaces the original bin if the reassembled one is better than in terms of completion and contamination. Like the Bin\_refinement module, the Reassemble\_bins module takes in minimum completion (-c) and maximum contamination (-x) parameters to allow the user to define what they consider a “good” bin. The bins produced from the water, gut, and soil data with metaWRAP-Bin\_refinement module runs (–c 50 –x 10) were run through the metaWRAP-Reassemble\_bins module (-c 50 –x 10), and the resulting bins were re-evaluated with CheckM.

Of the 235 bins from the water data set, 184 were improved through reassembly and 51 remained unchanged. Of the 175 bins from the gut data set, 172 were improved through reassembly and 3 remained unchanged. Finally, of the 134 bins from the soil dataset, only 3 were improved through reassembly, and the rest remained unchanged. (Figure 5)

With the exception of the soil bins, the bin sets were significantly improved by the Reassemble\_bins module. In water bins, the N50 increased by an average of 121% +/- 151.5%, the completion increased by an average of 1.49% +/- 2.41%, and the contamination reduced by an average of 0.85% +/- 1.13%. In the gut bins, the N50 increased by an average of 128.8% +/- 85.3%, the completion increased by an average of 2.48% +/- 2.89%, and the contamination reduced by an average of 1.21% +/- 1.13%.

The success of the bin reassembly algorithm relies heavily on accurate and specific recruitment of the correct reads to each bin. In cases where there are many closely-related organisms, the read recruitment may not be specific enough. This confuses the assembler during the re-assembly stage, which results in an inferior bin to the original. In very diverse and heterogeneous communities such as soil the metaWRAP-Reassemble\_bins will have only a small improvement on the bins. However, draft genomes from complex gut and water samples were significantly improved with the module. It is important to note that because the scaffolds are re-named during the reassembly, metaWRAP modules that rely on links between the entire metagenomic assembly and the bins (Quant-bins and Blobology) will not work on the reassembled bins. However, applications that rely on having complete, pure, and contiguous bins such as metaWRAP’s Classify\_bins and Annotate\_bins modules will benefit from the reassembly.

**MetaWRAP produces high-quality bins**

We investigated the performance of different binning approaches when extracting high quality draft genomes, with a contamination less than 5% and completion greater than 70%, 80%, 90%, and 95%. The number of bins meeting these criteria were counted in the original bin sets (metaBAT2, MaxBin2, CONCOCT) and binning refining software (metaWRAP, Binning\_refiner, DAS\_Tool). MetaWRAP was run with a maximum contamination of 5% and a minimum completion of 70% (metaWRAP’s default), 80%, 90%, or 95%. (Figure 6)

The default run of metaWRAP consistently produced the highest number of quality draft genomes at every completion cut off and in every sample type, with the exception of gut bins at 95% completion. The default metaWRAP produced 185, 158, 117, and 64 bins (at 70%, 80%, 90%, and 95% completion, respectively) from the water data set, 132, 103, 66, and 33 bins from the gut data set, and 47, 29, 15, and 6 bins from the soil data set. These numbers further improved when re-running metaWRAP with minimum completion setting (-c) corresponding to the bin quality range being considered. Custom metaWRAP runs returned 185, 164, 128, and 72 water bins, 132, 107, 76, and 46 gut bins, and 47, 30, 15, and 6 soil bins.

While the default performance of metaWRAP’s Bin\_refinement module is impressive, its results were even better with the use of appropriate settings. When informing metaWRAP-Bin\_refinement module of the target bin quality, it significantly outperformed every other tested binning and bin refinement method at every quality threshold.

The reassembly of the refined bins with metaWRAP-Reassemble\_bins made a further improvement on the number of high-quality draft genomes extracted from the gut and water data sets. Default reassembly produced 199, 174, 132, and 75 water bins and 146, 120, 78, and 42 gut bins with 70%, 80%, 90%, and 95% completion, respectively. This was a significant improvement compared to 185, 158, 117, and 64 water bins and 132, 103, 66, and 33 gut bins from the non-reassembled sets. When informing the metaWRAP-Reassemble\_bins module with the completion threshold being considered by using the –c option, the module produced 199, 173, 134, and 76 water bins and 146, 119, 80, and 53 gut bins - an even further improvement. Due to the high diversity and heterogeneity of the communities, the number of bins produced from the soil dataset remained unchanged with reassembly.

While it is unfair to declare that metaWRAP is the best current bin extraction method based on these benchmarking alone, it should be considered that metaWRAP is capable of improving bin sets from any binning software. Therefore, even when better metagenomic binning software are developed, their outputs can still be further improved with metaWRAP refinement and reassembly algorithms.

**MetaWRAP offers analysis and visualization of metagenomic bins**

The bins produced by the metaWRAP-Bin\_refinement module (comp ≥ 50%, cont ≤

10%) were then run through the metaWRAP-Classify\_bins module to determine their taxonomy. The bins from the data sets belonged to a wide range of phyla, but most of the water bins were Proteobacteria and Bacteroidetes, most of the gut bins were Firmicutes, and most of the soil bins were Proteobacteria (Figure S11). The soil bins were the most difficult to assign taxonomy to, with only 37% of the bins classified to the Phylum level (Figure S10).

The metaWRAP-Quant\_bins module was used to estimate the abundance of each bin across samples, and the results are shown in a heatmap (Figure S9). The metaWRAP-Blobology module was used to visualize the taxonomic composition of the water, gut, and soil communities with the use of Taxon-Annotated-GC-Coverage plots (TAGC plots). The module was also used to annotate the bins (comp ≥ 70%, cont ≤ 10%) that the contigs belonged to, allowing for visual inspection of the binning process success (Figure 7), and the BLAST-determined taxonomy of the binned contigs (Figure S12). Finally, the draft genomes were run though the Annotate\_bins module for functional annotation.

While these modules do not offer significant algorithmic breakthroughs metagenomic bin analysis, they offer a convenient way to quickly examine and analyze a set of bins. The Blobology module allow the user to inspect bins in context of the entire community, and the Quant\_bins module allows for visual and numerical inspection of community composition shifts and differences at the single genome level. Finally, the user may use the Classify\_bins module to get a conservative but reliable taxonomy estimate of each draft genome, and use the Annotate\_bins module to get functional gene annotations that they may use in downstream analysis.

METHODS

**CAMI binning benchmarking**

The “gold standard” assemblies from the “high”, “medium”, and “low” diversity CAMI challenges were binned with metaBAT2 v2.12.1, Maxbin2 v2.2.4, and CONCOCT v0.4.0 using the metaWRAP-Binning module with default parameters. The resulting three bin sets were then consolidated with DAS\_Tool v1.1.0 (default settings, blast used for search engine), Binning\_refiner v1.2 (default settings), and metaWRAP-Bin\_refinement v0.7 to attempt to improve the bin sets. To simulate a realistic metagenomic pipeline, the completion and contamination of the bins in all six bin sets was first evaluated with CheckM v1.0.7 with default parameters, and bins with a completion less than 50% or a contamination greater than 10% were discarded. The true recall and precision of the bins within the six resulting bin sets was then determined with Amber v0.6.2, which compared the bins against the known original genomes. Bin recall and precision were converted to completion and contamination percentages.

**Real data binning benchmarking**

The raw sequences from water, gut, and soil microbiomes were run through the metaWRAP-Read\_qc module to trim the reads and remove human contamination, and then the Kraken module of metaWRAP was run on the quality-controlled reads to investigate the taxonomic profile of the community. The reads were then co-assembled within each community type with MegaHit v1.1.2 by using the metaWRAP-Assembly module. Contigs shorter than 1000bp were discarded, with the exception of the soil assembly, for which the cutoff of 3000bp was chosen to reduce the binning time. The contigs were then binned with metaBAT2 v2.12.1, Maxbin2 v2.2.4, and CONCOCT v0.4.0 using the metaWRAP-Binning module at default settings. The resulting three bin sets of each microbiome type were then passed to DAS\_Tool v1.1.0 (default settings, blast used for search engine), Binning\_refiner v1.2 (default settings), and metaWRAP-Bin\_refinement v0.7 to attempt to improve the bin sets. For the main benchmark, metaWRAP was run with –c 50 –x 10 settings. To benchmark the bins produced by all the binning methods, the completion and contamination of the bins was estimated with CheckM v1.0.7.

**Bin\_refinement optimization demonstration**

The metaWRAP-Bin\_Refinement module was run with a variety of settings to demonstrate performance changes at different –c (minimum completion) and –x (maximum contamination) settings. First, the bin sets produced with metaBAT2 v2.12.1, Maxbin2 v2.2.4, and CONCOCT v0.4.0 were refined with the module with a constant maximum contamination setting –x 10, but varying minimum completion settings –c 50, 60, 70, 80, 90, and 95. Then the same bin sets were refined with a constand minimum contamination setting –c 50, but varying maximum contamination setting of –x 10, 8, 6, 4, 2, and 1.

**Reassembly benchmarking**

To benchmark overall reassembly performance, bin sets produced by the metaWRAP-Bin\_refinement module with -c 50 –x 10 settings were run through the metaWRAP-Reassemble-bins module with -c 50 –x 10 settings. The resulting bins were evaluated with CheckM v1.0.7, and the completion and contamination values were sorted and plotted.

**Extracting high-quality draft genomes**

MetaWRAP’s Bin\_refinement and Reassemble\_bins modules were run with different settings to extract high quality draft genomes (contamination less than 5%, completion greater than 70%, 80%, 90%, or 95%) to showcase the overall binning potential of metaWRAP. To benchmark the Bin\_refinement module, it was run on bin sets produced with metaBAT2 v2.12.1, Maxbin2 v2.2.4, and CONCOCT v0.4.0 with four different settings: -c 70 –x 5, -c 80 –x 5, -c 90 –x 5, -c 95 –x 5. To benchmark the Reassemble\_bins module, it was run with the same settings on the output of of Bin\_refinement with -c 60 –x 10, -c 70 –x 10, -c 80 –x 10, and –c 90 –x 10 settings, respectively. Finally, all the resulting metaWRAP bin sets, the original bin sets, as well as the refinements from DAS\_Tool and Binning\_refiner were evaluated with CheckM v1.0.7 and the number of bins with different completion and contamination values were counted and plotted.

**MetaWRAP’s pre-processing modules**

The Read\_qc module is meant to pre-process raw Illumina sequencing reads in preparation for assembly and alignment. The raw reads are trimmed with Trim-galore v0.4.3 at default settings, and then the human-derived reads (contamination) are removed with bmtagger v3.101. Read pairs with a single suspected human read are also removed. Finally, FASTQC is used to generate quality reports of the raw and final read sets in order to assess read quality improvement. The user has control over which of the above features he wishes to use.

The Assembly module allows the user to assemble a set of metagenomic reads with either metaSPAdes or MegaHit, both set to default assembly settings. While metaSPAdes results in a superior assembly in most samples, MegaHit scales well with large datasets, and is therefore set as the defualt. The assemblies are then formatted to include the scaffold length and kmer depth, sorted by length, and contigs shorter than 1000bp are removed. Finally, an assembly report is generated with QUAST v4.5 with default settings.

The Kraken module takes in any number of FastQ or FastA files, classifies the contained sequences with KRAKEN v0.10.6, and reports the taxonomy distribution in an interactive html kronagram using KronaTools v2.7. If a passed FastA file is an assembly file from the Assembly module, the taxonomy of each contig is weighted based on its length and coverage [weight=coverage\*length], which are encoded in the scaffold naming.

The Binning module is meant to be a convenient wrapper around three metagenomic binning software: metaBAT v2.12.1, Maxbin v2.2.4, and CONCOCT v0.4.0. First the metagenomic assembly is indexed and paired end reads from any number of samples are aligned to it with bwa v0.7.15. The alignments are sorted and compressed with Samtools v1.6, and library insert size statistics are also gathered at the same time (insert size average and standard deviation). MetaBAT2’s jgi\_summarize\_bam\_contig\_depths function is used to generate contig abundance table, and it is then converted into the correct format for each of the three binning softwares. The assembly is binned with software(s) of the user’s choice, and the resulting bins are optionally evaluated with CheckM v1.0.7.

**MetaWRAP’s bin analysis modules**

The Bin\_refinement module utilizes a hybrid approach to take in two or three bin sets that were obtained with different binning approaches and produces a consolidated, improved bin set. First, binning\_refiner v1.2 is used to hybridize the bin sets in every possible combination. If there are three original bin sets A, B, and C, they will be hybridized to produce bin sets AB, BC, AC, and ABC. CheckM is then run to evaluate the completion and contamination of the bins in each of the 7 bin sets (3 originals, 4 hybridized). The bins sets are then iteratively compared to each other, and each pair is consolidated into an improved bin set. To do this, the same bin is identified within the two bin sets based on a minimum of 80% overlap in genome length, and the better bin is selected based on the scoring function S=Completion-5\*Contamination. Only bins that meet the minimum completion (-c) and maximum contamination (-x) criteria are considered. After all bin sets are incorporated into a consolidated set, duplicate contigs are removed. By default, duplicate kept only in the superior bin (based on scoring function). CheckM is then re-run on the final bin set and a final report file is generated with a custom script (.stats files). Completion and contamination rank plots are also made to compare the quality of the original bins and the Bin\_refinement module output.

The Reassemble\_bins module aims to improve a set of bins by extracting reads that belong to each bin and re-assembling them. First, entire original metagenomic assembly is indexed and FastQ reads are aligned back to it with bwa v0.7.15. Reads pairs mapping back to contigs belonging to the provided bins are stored in separate FastQ files, even if only one read mate aligned. Two sets of reads are stored for each bin – reads mapping perfectly (strict), and reads mapping with <3 mismatches (permissive). Each set of reads is then reassembled with SPAdes v3.11 with the --carefull setting, and short contigs (<1000bp) are removed. CheckM 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” reassembled bin. The best version is chosen based on a scoring funciton S=Completion-5\*Contamination. The final bins set it then re-evaluated with CheckM, and summary statistics are generated. Additionally, a N50, completion and contamination rank plots is generated to evaluate the improvements in the bin sets following reassembly.

The Quant\_bins module rapidly estimates the abundance of bins across a number of samples. Salmon v0.9.1 is used to index the entire metagenomic assembly and align reads from each sample back to the assembly. Coverage tables are generated estimating the abundance of each contig in each sample. The average abundance of each bin in each sample is calculated by taking the length-weighted average of the bins’s contig abundances. A final bin abundance table is made, and a clustered heatmap is generated with Seaborn v0.8.1 to visualize bin abundance variation across samples.

The Blobology module uses a modified version of the original Blobology software to create blobplots (a GC vs abundance plot of all the contigs) of a metagenomic assembly, and annotates it with phylogenetic information or bin information. The taxonomy of each contig is estimated with Mega-BLAST v2.7.1 with NCBI\_nt as the database. The assembly is then indexed and the reads from any number of samples are aligned against it with bowtie2. Finally Blobology’s gc\_cov\_annotate.pl function is used to generate a blobplot file with the GC, coverage (in all samples), and taxonomy of each contig. If the user provided a set of bins to annotate, the contigs are also annotated with the bins they belong to. Finally, Blobology’s makeblobplot.R function is used to make the blobplots of the contigs across all the provided samples, with taxonomic and bin membership annotations.

The Classify\_bins module is a conservative way to assign taxonomy to a set of metagenomic bins. First, the contigs in all bins are combined into one file, and MegaBLAST v2.7.1 is used to align the contigs to the NCBI\_nt database. The alignment results are then used by taxator-kt to estimate the taxonomy of each contig. Finally, the most likely taxonomy of each bin is estimated from individual contig predictions. Taxonomy of each contig are added to a phylogenetic tree, adding weight to each branch based on the length of that contig. The tree is then traversed from the root down the heaviest branches until the next likely branch is <50% of the current branch weight. Once no further taxonomic rank can be estimated, the final taxonomy of that bin is reported.

The Annotate\_bins module takes in a set of bins and quickly functionally annotates them with PROKKA v1.12. The annotation process is parallelized for any number of bins and threads. For each bin, the module returns the annotation file in GFF format, and two FastA files with untranslated and translated genes.

OTHER SECTIONS

**Additional files**

Additional file 1 --- Figure S1

Detailed walkthrough of the data files, software, databases, and custom scripts that metaWRAP uses. The components of each metaWRAP module grouped and denoted with dotted lines.

Additional file 2 --- Figure S2

Logical workflow of the Bin\_refinement modules of metaWRAP. The module takes in three bin sets produced from the same assembly by different software or different parameters of the same software. Binning\_refiner is used to create hybridized intermediates (4 possible combinations), and the completion and contamination of the original and hybridized bins is estimated with CheckM. The best version of each bin is then found in the resulting 7 bin sets.

Additional file 3 --- Figure S3

Logical workflow of the Reassemble\_bins module, which extracts reads belonging to bins in a given bin set, and individually reassembles them. BWA is used to map reads to the assembly, and the reads that fall onto contigs belonging to a bin are that map to particular bins are split into separate files. This process is don’t for perfectly mapping reads (strict) and reads mapping with less than 3 mismatches (permissive). After individual reassembly with SPAdes, CheckM is used to assess the completion and contamination of the original and the two reassembled variants of each bin, and a scoring function is used to choose the best of the three versions.

Additional file 4 --- Figure S4

Recall and precision of bins recovered from the CAMI binning challenge synthetic data sets using different binning strategies. The bin sets in dashed lines (metaBAT2, MaxBin2, CONCOCT) are original sets, while the bin sets in solid lines (DAS\_Tool, Binning\_refiner, metaWRAP) are bins produced by combining the original three sets. Only bins with a recall of greater than 0.5 and precision greater than 0.9 are shown.

Additional file 5 --- Figure S5

Completion of bins recovered from different metagenomic data sets by using metaWRAP with a varying minimum completion parameter (-c), but constant maximum contamination parameter (-x 10). The numbers in the brackets indicate the number of extra bins gained at that threshold compared to the baseline (running metaWRAP with minimum completion of 50% and maximum contamination of 10%).

Additional file 6 --- Figure S6

Contamination of bins recovered from different metagenomic data sets by using metaWRAP with a varying maximum contamination parameter (-x), but constant minimum completion parameter (-c 50). The numbers in the brackets indicate the number of extra bins gained at that threshold compared to the baseline (running metaWRAP with minimum completion of 50% and maximum contamination of 10%).

Additional file 7 --- Figure S7

Taxonomic distribution estimated with the Kraken module of metaWRAP of three different metagenomic sample types. Based on KRAKEN taxonomy of reads subsampled to 10 million reads.

Additional file 8 --- Figure S8

Heatmaps showing the log of bin abundance of extracted bins (comp ≥ 50%, cont ≤ 10%) across samples in three microbiomes, as determined by metaWRAP’s Quant\_bins module. Abundance is as the average read coverage of each bin, standardized to 100M reads in each sample library.

Additional file 9 --- Figure S9

Percent of bins from three metagenomic sample sources classified at different taxonomic ranks using the Classify\_bins module of metaWRAP.

Additional file 10 --- Figure S10

Distribution of the taxonomy among Bacterial bins extracted from three different microbial communities using metaWRAP’s Bin\_refinement module. Taxonomy estimated with metaWRAP’s Classify\_bins module.

Additional file 11 --- Figure S11

Blobplot visualization of three metagenomic data sets, showing the GC and average coverage of each successfully binned contig in the assemblies, and annotated with the taxonomy at the phylum level as determined by BLAST, and the bins that they belong to (bin colors are chosen at random). Only contigs belonging to bins with a completion greater than 70% and contamination less than 10% are shown.

**Abbreviations**

WGS, whole genome sequencing; WMG, whole metagenome;

**Acknowledgements**

We thank early users of metaWRAP for their patience and help with debugging.

**Funding**

This work was supported by grant grant NNX15AP18G from NASA and grant DEB1556574 from the NSF. The funders had no role in study design, data collection, analysis and interpretation, decision to publish, or preparation of the manuscript.

**Availability of data and materials**

MetaWRAP is an open source software available at GitHub https://github.com/ursky/metaWRAP and Anaconda https://anaconda.org/ursky/metawrap-binning. Synthetic data used in benchmarking is available from the original CAMI challenge https://data.cami-challenge.org/participate. The Water metagenomes came from the Central Baltic Surface Water Metagenome, and are available from SRA under accession IDs SRR2053273–SRR2053308. The gut data set came from the Metagenomic of the Human Intestinal Tract (MetaHIT) survey, and is available from SRA under accession IDs ERR011087-ERR011136. The soil data set came from grassland soil microbial communities from Angelo Coastal Reserve, and are available under Gold Analysis Project IDs: Ga0007435, Ga0007436, Ga0007437, Ga0007438, Ga0007439, and Ga0007440. All analysis results and scripts used to generate figures are available at https://github.com/ursky/metawrap\_paper.

**Author's contributions**

GU created, released, and maintained the metaWRAP software, ran the benchmarks, and wrote the manuscript. JDR and JT provided ideas for improving metaWRAP and edited the manuscript. All authors read and approved the final manuscript.

**Ethics approval and consent to participate**

Not applicable

**Consent for publication**

Not applicable

**Competing interests**

The authors declare that they have no competing interests.