A peer-reviewed version of this preprint was published in PeerJ on 26 July 2019.

View the peer-reviewed version (peerj.com/articles/7359), which is the preferred citable publication unless you specifically need to cite this preprint.

Kang DD, Li F, Kirton E, Thomas A, Egan R, An H, Wang Z. 2019. MetaBAT 2: an adaptive binning algorithm for robust and efficient genome reconstruction from metagenome assemblies. PeerJ 7:e7359 https://doi.org/10.7717/peerj.7359

MetaBAT 2: an adaptive binning algorithm for robust and efficient genome reconstruction from metagenome assemblies

Dongwan Kang [1] , **Feng Li** [2] , **Edward S Kirton** [1] , **Ashleigh Thomas** [1] , **Rob S Egan** [1] , **Hong An** [2] , **Zhong Wang**
Corresp. 1, 3, 4

[1] Department of Energy, Joint Genome Institute, Walnut Creek, CA, United States of America
[2] School of Computer Science and Technology, University of Shanghai for Science and Technology, Hefei, Anhui, China
[3] Genomics Division, Lawrence Berkeley National Laboratory, Berkeley, United States of America
[4] School of Natural Sciences, University of California at Merced, Merced, United States of America

Corresponding Author: Zhong Wang
Email address: zhongwang@lbl.gov

We previously reported MetaBAT, an automated metagenome binning software tool to reconstruct single genomes from microbial communities for subsequent analyses of uncultivated microbial species. MetaBAT has become one of the most popular binning tools largely due to its computational efficiency and ease of use, especially in binning experiments with a large number of samples and a large assembly. MetaBAT requires users to choose parameters to fine-tune its sensitivity and specificity. If those parameters are not chosen properly, binning accuracy can suffer, especially on assemblies of poor quality. Here we developed MetaBAT 2 to overcome this problem. MetaBAT 2 uses a new adaptive binning algorithm to eliminate manual parameter tuning. We also performed extensive software engineering optimization to increase both computational and memory efficiency. Comparing MetaBAT 2 to alternative software tools on over 100 real world metagenome assemblies shows superior accuracy and computing speed. Binning a typical metagenome assembly takes only a few minutes on a single commodity workstation. We therefore recommend the community adopts MetaBAT 2 for their metagenome binning experiments. MetaBAT 2 is open source software and available at https://bitbucket.org/berkeleylab/metabat.

MetaBAT 2: an adaptive binning algorithm for robust and efficient genome reconstruction from metagenome assemblies

Dongwan D. Kang[1], Feng Li[1,2], Edward Kirton[1], Ashleigh Thomas[1], Rob Egan[1], Hong An[2], and Zhong Wang[1,3,4]

[1]**Department of Energy, Joint Genome Institute, Walnut Creek, CA 94598, USA**
[2]**School of Computer Science and Technology, University of Science and Technology of China, Hefei, Anhui, China**
[3]**Environmental Genomics and System Biology Division, Lawrence Berkeley National Laboratory, Berkeley, CA 94720, USA**
[4]**School of Natural Sciences, University of California at Merced, Merced, CA, 95343, USA**

Corresponding author:
Zhong Wang[1,3,4]

Email address: zhongwang@lbl.gov

ABSTRACT

We previously reported MetaBAT, an automated metagenome binning software tool to reconstruct single genomes from microbial communities for subsequent analyses of uncultivated microbial species. MetaBAT has become one of the most popular binning tools largely due to its computational efficiency and ease of use, especially in binning experiments with a large number of samples and a large assembly. MetaBAT requires users to choose parameters to fine-tune its sensitivity and specificity. If those parameters are not chosen properly, binning accuracy can suffer, especially on assemblies of poor quality. Here we developed MetaBAT 2 to overcome this problem. MetaBAT 2 uses a new adaptive binning algorithm to eliminate manual parameter tuning. We also performed extensive software engineering optimization to increase both computational and memory efficiency. Comparing MetaBAT 2 to alternative software tools on over 100 real world metagenome assemblies shows superior accuracy and computing speed. Binning a typical metagenome assembly takes only a few minutes on a single commodity workstation. We therefore recommend the community adopts MetaBAT 2 for their metagenome binning experiments. MetaBAT 2 is open source software and available at `https://bitbucket.org/berkeleylab/metabat`.

INTRODUCTION

Studies of microbial communities based on microbial isolation and cultivation have been gradually replaced by high throughput, whole genome shotgun sequencing based metagenomics (reviewed in (Van Dijk et al., 2014; Tringe and Rubin, 2005)). Advances in computational metagenomics have produced tools that assemble billions of short sequence reads derived from deep metagenome sequencing into larger fragments (contigs), and subsequently group them into draft genomes by metagenome binning (reviewed in (Kang et al., 2016)).

Recently we have witnessed exciting progress in metagenome binning as several automatic binning tools become available. Our group developed MetaBAT (Kang et al., 2015) in 2015, among a few others developed around the same time, including MyCC (Lin and Liao, 2016), MaxBin 2.0 (Wu et al., 2015), MetaWatt-3.5 (Strous et al., 2012) and CONCOCT (Alneberg et al., 2014). These binning software tools have achieved various extents of success with simulated data or real world data. However, in practice the quality of binning experiments is largely dependent on characteristics of the underlying dataset and hence the choice of binning parameters. Our users and ourselves both independently observed that MetaBAT's binning performance can vary greatly among different parameter choices. As there are no established

parameter optimization methods, to get a comprehensive binning result one has to run multiple binning experiments with different sets of parameters followed by merging the results. For example, in a recent large scale study of over 1500 metagenome datasets, 8,000 draft genomes were obtained by merging 5 MetaBAT binning results, each derived from a different parameter set (Parks et al., 2017).

In the recent Critical Assessment of Metagenome Interpretation (CAMI) metagenome binning challenge (Sczyrba et al., 2017), MetaBAT is the fastest and most robust software that can scale up to large metagenomic datasets with millions of contigs. Its accuracy was not the best, however, likely due to its lack of robustness towards various datasets. We therefore replaced the core binning algorithm with a completely new one and report MetaBAT 2 (the original MetaBAT hereafter referred as MetaBAT 1) in this study. The new algorithm consists of several new aspects: 1) normalized TNF scores, 2) a graph structure and an iterative graph partitioning procedure for clustering and 3) additional steps to recruit smaller contigs. In addition, we greatly improve the computational efficiency so that the increase in calculations does not affect the program's scalability.

MetaBAT 2 has been packaged by the research community as a Bioconda package (`https://bioconda.github.io/recipes/MetaBAT2`) and as a standard APP on the DOE Knowledgebase platform (`https://kbase.us/applist/apps/metabat/run_metabat/release`). A docker image is also available (`https://hub.docker.com/r/metabat/metabat`). There are numerous studies that have reported using MetaBAT 2 and its associated tools for successful large scale metagenomic analyses (e.g., Rinke et al. (2018); Bahram et al. (2018); Pasolli et al. (2019)). Here we focus on describing how MetaBAT 2 works, while providing performance benchmarks on a few synthetic datasets and a large number of real world datasets.

METHODS

The adaptive binning algorithm

MetaBAT 2 uses the same raw tetra-nucleotide frequencies (TNF) and abundance (ABD) scores as those in MetaBAT 1. There are three major changes in binning algorithms as listed below.

Score normalization

We use ABD to rank-normalize TNF (where the smallest TNF becomes the smallest ABD and the greatest TNF becomes the greatest ABD, and so on) then the composite score (S) is calculated by the geometric mean of TNF and ABD as the following,

$$S = TNF^{(1-w)} * ABD^{w}$$

where $w = nABD/(nABD+1)$, where nABD represents the number of effective samples which have enough coverage (by default > 1) for at least one of the contigs. Whenever there are 3 or more samples available, an abundance correlation score (COR) is also calculated using the Pearson correlation coefficient and then rank-normalized using ABD. In this case, S is calculated as the geometric mean of TNF, ABD, and COR.

$$S = sqrt(TNF^{(1-w)} * ABD^{w} * COR)$$

In this way all scores fall within the same range. S should be more accurate especially when the communities are extremely complex.

Graph-based clustering

Instead of the modified k-medoid clustering algorithm implemented in MetaBAT 1, MetaBAT 2 uses a graph based structure for contig clustering. A graph with contigs as nodes and their similarity as edges is constructed in two steps. During the first step, an initial graph is constructed by only using TNF. Since TNF usually are not very reliable, we only use strong TNF scores for the first stage graph. Here we also put a limit on the number of edges per node to reduce computation, a parameter that can be adjusted to control sensitivity/specificity.

The second step is an iterative procedure of graph building and graph partitioning. At each iteration, a subset of edges with the highest similarity scores (S) are incorporated into the above graph, followed by graph partitioning using a modified label propagation algorithm (LPA, (Zhu and Ghahramani, 2002)). The LPA was modified so that the partitioning is deterministic since the search order is decided by edge strength, and the previous partitioning results are used as labels for the graph in the next iteration to speed

PeerJ Preprints | https://doi.org/10.7287/peerj.preprints.27522v1 | CC BY 4.0 Open Access | rec: 6 Feb 2019, publ: 6 Feb 2019

up binning. In addition, we use Fisher's method (also used in MetaBAT 1) to combine edge strengths and compare them to decide the best neighborhood.

Small contigs/bins recruiting

MetaBAT 1 by default uses contigs 2.5kb or larger. As many metagenome assemblies contain smaller contigs, MetaBAT 2 includes an additional step to include small contigs (between 1kb and 2.5kb), as well as the contigs from small bins ($< 200kb$) if there are 3 or more samples in the dataset. In this additional step, a "free" contig is assigned to a specific bin where its correlation to member contigs from that bin is larger than the mean correlation among the contigs themselves.

Metagenome assemblies used for benchmarking binning

Three synthetic datasets (Low-, Medium- and High-complexity, respectively) were downloaded from the CAMI website (Sczyrba et al., 2017).

120 real world metagenome assemblies were obtained from The Integrated Microbial Genomes & Microbiomes system (IMG/M: `https://img.jgi.doe.gov/m/`) (Chen et al., 2018). A complete list of the samples and their IMG access IDs are available in the supplemental table S1.

Software tools used for benchmarking binning

The other software tools we used are their latest version, CONCOCT 0.4.0, MaxBin 2.2.4, MyCC(docker image 990210oliver/mycc.docker:v1), BinSanity v.0.2.6.4 and COCACOLA python version (updated on March 5,2017), respectively. All tools were run with their default parameters.

Searching best parameters by a genetic algorithm

The genetic algorithm was performed using the following parameters: population size: 10, selection size: 3, mutation rate: 0.05, crossover rate: 0.01, minimum/maximum generations: 3/10, and binary tournament selection. To evaluate the performance of binning, we used the Minimum Information about Metagenome-Assembled Genome (MIMAG) standards described in (Bowers et al., 2017). The number of high-quality putative genomes was used as the fitness score, where high-quality is defined as $>= 90\%$ complete and $<= 5\%$ contamination as determined by CheckM, $>= 18$ tRNAs identified by tRNAscan-SE (Lowe and Eddy, 1997), and all three ribosomal subunits, found by cmsearch. While tRNA and rRNA annotations can be annotated just once per contig, CheckM (Parks et al., 2015) must be run on each parameter set's results and is the time-limiting step.

Computational optimization

The above changes in the binning algorithm in MetaBAT 2 require significantly more computation than MetaBAT 1. To make MetaBAT 2 work well with similar computing resources in a comparable runtime, we implemented several computational optimization techniques to improve its resource efficiency.

Computing efficiency

In addition to the original multi-thread strategy in MetaBAT 1, we also applied a lower level optimization on CPU cache memory access. A typical CPU has only 8-64KB level-1 cache and 256KB-2MB level-2 cache. When the data is bigger than the level-2 cache (e.g., TNF distance matrix), the data is kept in random-access memory that is much slower to access. We adopted a loop tiling threading model that divides the TNF distance matrix into many smaller "tiles" that fit into level-2 cache, and distributes the calculation of each tile among many threads. This optimization alone gains a 35% performance improvement over the original parallel code.

Memory efficiency

With thousands or even millions of contigs, the pair-wise distance matrix gets bigger and can take a large amount of memory. To avoid storing the entire matrix in RAM, we use a priority queue data structure to store only the top k strong links of every contig for iterative clustering. This method scaled down the memory usage from $O(N^2)$ to $O(N)$, at the cost of a few extra calculations. The parameter k will affect the sensitivity and specificity of binning results, as smaller k values lead to high specificity but low sensitivity.

PeerJ Preprints | https://doi.org/10.7287/peerj.preprints.27522v1 | CC BY 4.0 Open Access | rec: 6 Feb 2019, publ: 6 Feb 2019

RESULTS

Accuracy benchmarks on synthetic and real world metagenome assemblies

During the first CAMI challenge, MetaBAT 1 was the fastest and most scalable software, but its accuracy was only average compared with CONCOCT, MyCC and MaxBin 2.0 (Sczyrba et al., 2017). To test whether MetaBAT 2 improves binning accuracy, we benchmarked MetaBAT 2 by comparing it against MaxBin2, CONCOCT, and MyCC. In addition, we added BinSanity (Graham et al., 2017) and COCACOLA (Lu et al., 2017), two new automatic binners developed after CAMI 1.0, for comparison. All the tools were run using their default parameters (methods).

The CAMI metagenome datasets were simulated with different species complexity and genome sizes, consisting of reads from 700 microbial genomes including strain-level diversity and 600 plasmids and viruses (Sczyrba et al., 2017). Three datasets were simulated at different complexity levels (high, medium and low complexity). To date they represent the best benchmark datasets for metagenome binning with known ground truth. We therefore ran the above tools on these three datasets. We used the same accuracy measures as we did in MetaBAT 1, i.e., number of genomes recovered at certain genome completeness (recall, 0.5, 0.6, 0.7, 0.8 and 0.9) and certain precision (0.9 and 0.95) cutoffs. The results are shown in Figure 1.

MetaBAT 2 shows better performance over the other tools in these experiments. In the CAMI Low Complexity dataset, MetaBAT 2 recovers the most genomes at almost every completeness/precision cutoff, except that CONCOCT recovers one more genome at the 90% completeness cutoff. In the other two datasets with higher community complexity, MetaBAT 2 bins more genomes than any other tool tested at every threshold. The difference seems to be more pronounced when complexity increases. For example, at 90% completeness and 95% precision levels MetaBAT 2 recovers 333 out of 753 genomes (44.2%) from the CAMI High Complexity dataset, while the next best software, MaxBin2, only recovers 195 genomes (25.9%). These results suggest the adaptive binning algorithm implemented in MetaBAT 2 can adapt to very complex microbial communities.

We also carried out benchmarking experiments on real world metagenome datasets downloaded from Integrated Microbial Genomes & Microbiomes (IMG/M) (Markowitz et al., 2011, 2013; Chen et al., 2018). We chose 120 metagenomes assembled from a very diverse environmental sample as our test dataset (methods). All these metagenome datasets were assembled with metaSPAdes (Nurk et al., 2017). Hereafter we refer this dataset as IMG100.

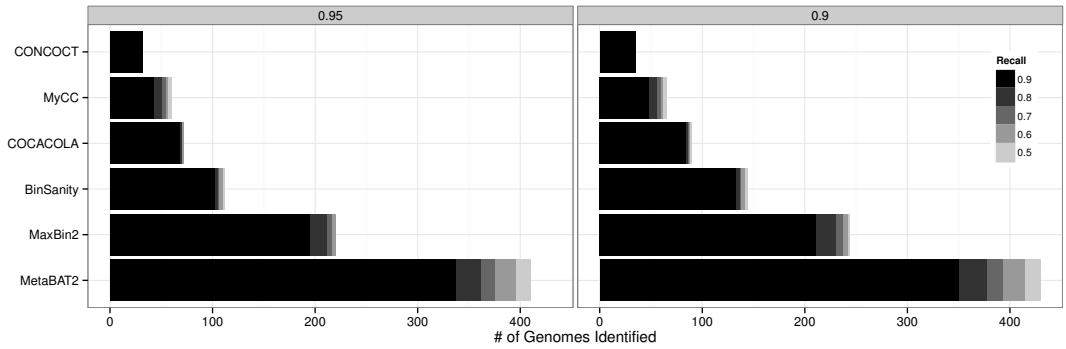
Most of these metagenome datasets produced very few bins from all software tools, with about half of them only producing fewer than 5 genome bins. Figure 2 shows the performance of each method in the top 13 metagenomes ordered by the number of genome bins identified (COCACOLA was not shown since it didn't produce any bins fulfilling the cutoff). Only genomic bins having contamination less than 5% are considered and each bar has 3 stacks representing the number of genomes fulfilling different completeness criteria of 90%, 70%, and 50%. (10% contamination cutoff shows similar pattern; data not shown). MetaBAT 2 outperforms other tools by a large margin for all examples.

The default set of parameters is good for most datasets

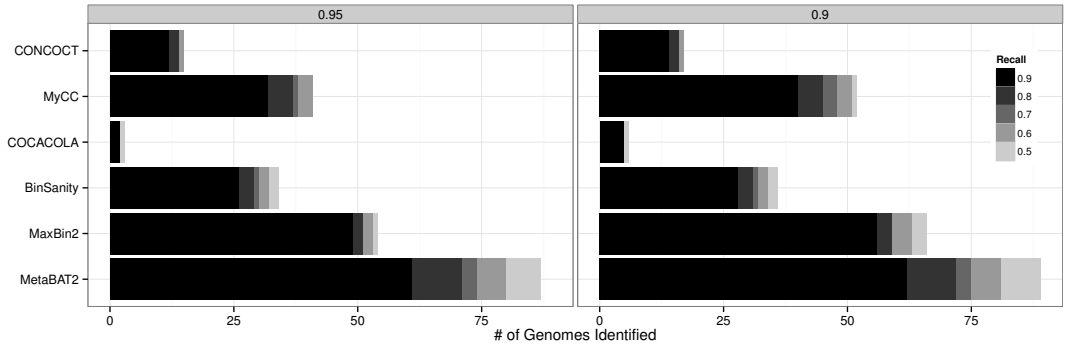
We next ask the question whether or not MetaBAT 2 requires different parameter sets for optimized accuracy for different datasets. There are only three parameters that may affect binning accuracy: 1) maxEdges (the maximum number of edges a node can have when constructing the graph, a lower number should reduce computing time but may also reduce sensitivity); 2) maxP (percentage of high quality contigs included for binning, a higher number gives more sensitivity); and 3) minS (the minimum score of an edge kept for binning, a higher number gives more specificity).

We designed a genetic algorithm to attempt to optimize the above parameters using the IMG 100 dataset (methods). For each of the samples in this dataset, the genetic algorithm systematically explores the parameter space and attempts to find the best parameter set for this sample. In this experiment we only considered the high-quality genome bins for performance scoring (methods). Among all the best parameter sets found from all samples, the default parameter set is selected to be the best for the majority (58%) of the samples (Supplemental Table S2). Comparing the next nine most frequently selected best parameter sets to the default one on all the samples showed that the default parameter set has a consistent performance (Figure 3). The genetic algorithm did find some parameter sets that are slightly better than the default on some samples under this scoring metric, although it significantly increased the total running time. Using a different scoring scheme or using a different set of testing data may select a different set of

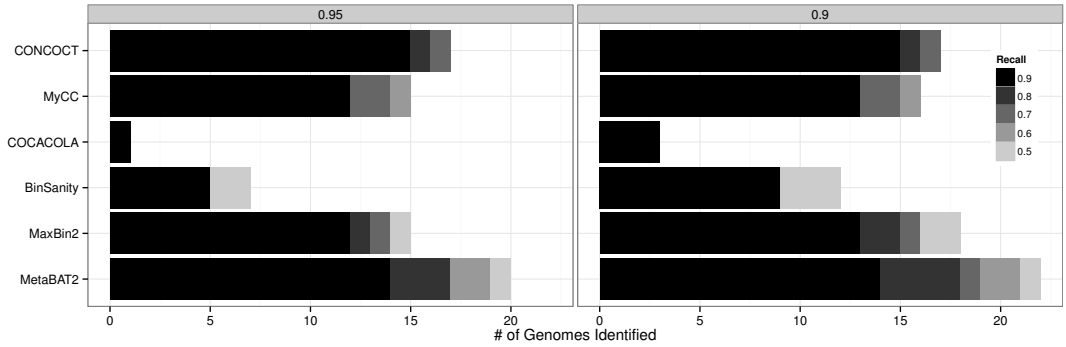
PeerJ Preprints | https://doi.org/10.7287/peerj.preprints.27522v1 | CC BY 4.0 Open Access | rec: 6 Feb 2019, publ: 6 Feb 2019



(a).CAMI High Complexity



(b).CAMI Medium Complexity



(c).CAMI Low Complexity

Figure 1. Benchmark of several popular binning tools on CAMI challenge datasets. The number of identified genomes are shown at two different precision levels, $\geq$ 95% (left column) or $\geq$ 90% (right column). The number of identified genomes recovered with a completeness (recall) level 90%, 80%, 70%, 60%, or 50% are represented by different shades of gray, with 90% being the darkest. Benchmarking results using the high complex dataset (a), medium complex dataset (b), and low complex dataset (c) are shown. All the tools (MyCC, CONCOCT, COCACOLA, BinSanity, MaxBin 2 and MetaBAT 2) were run using their default parameters. Completeness and precision were calculated with the ground truth of each dataset.

PeerJ Preprints | https://doi.org/10.7287/peerj.preprints.27522v1 | CC BY 4.0 Open Access | rec: 6 Feb 2019, publ: 6 Feb 2019

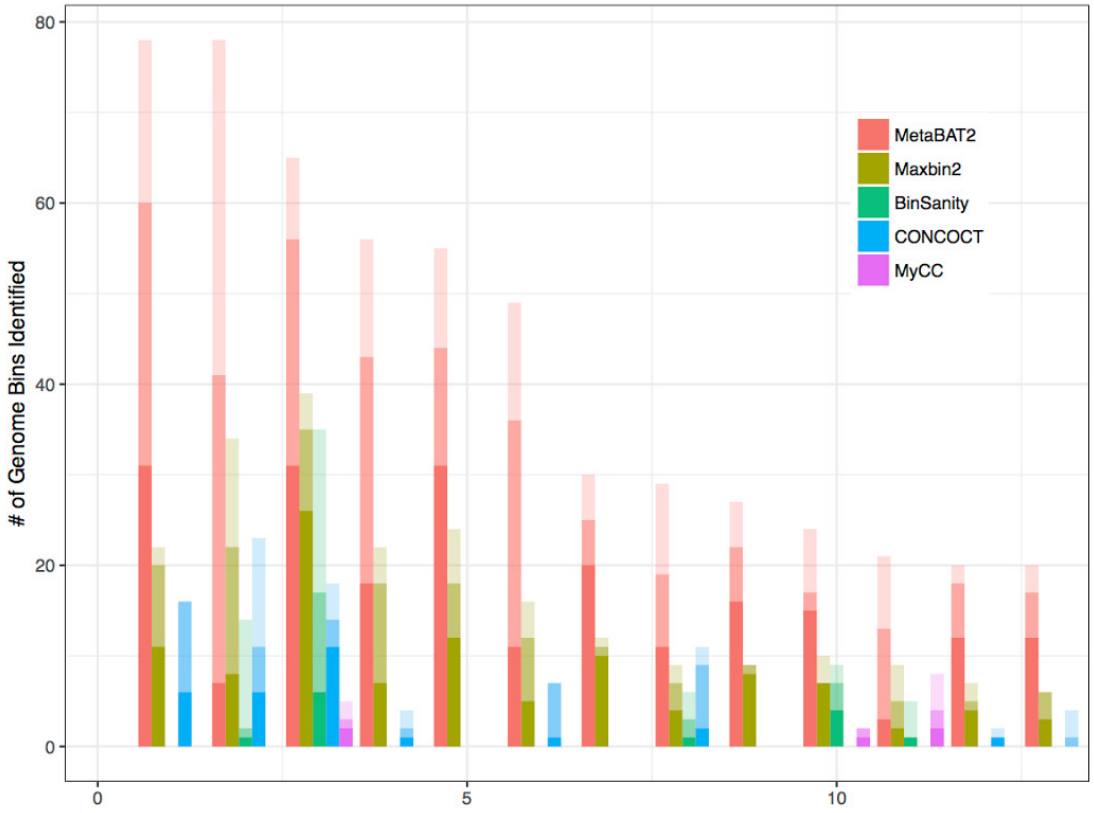


Figure 2. Comparing binning performance of MetaBAT 2 with alternative binning tools on real world metagenomes. 120 metagenome assemblies were obtained from IMG (IMG100, see methods). The top 13 assemblies with the most genome bins were shown. Using a 5% contamination cutoff, each bar has 3 stacks of results representing 90%, 70%, and 50% completeness from the same binning tool, respectively, with 90% having the darkest shade.

parameters, but the benefit over the default parameters appears to be very small (data not shown).

We also experimented with MetaBAT 1 on the IMG100 set, using the two most commonly used preset parameters (sensitive and superspecific). In this comparison setting we can see whether or not MetaBAT 2 with default parameters is more sensitive than MetaBAT 1 (sensitive) and more specific than MetaBAT 1 (very specific). Figure 4 shows the performance of each method in the top 20 metagenomes ordered by the number of genome bins identified. In the majority of the binning experiments, MetaBAT 2 outperforms both modes of MetaBAT 1, demonstrating a robust performance without parameter tuning.

Benchmarking MetaBAT 2's computing efficiency

In contrast to most of the other software tools that are implemented with Python, MetaBAT 2 is written in C++ with extensive low-level computational optimization (Methods). This gives MetaBAT 2 a unique advantage in computational efficiency. The runtime and memory consumption of MetaBAT 2 and alternative tools on CAMI High, Medium, and Low complexity datasets are shown in Table 1. The tests were run on a workstation with 2 Intel Xeon CPUs @ 2.30GHz, each with 16 cores and 40MB smart cache, and 128 GB RAM.

MetaBAT 2 finished the CAMI Low dataset in just 7 seconds, while most other tools took 11 minutes or more, which is 90 times or more slower than MetaBAT 2. It finished the Medium and the High datasets in 25 seconds and 1 min 54 seconds, respectively. The others need from one hour to several hours for binning the two datasets. Memory requirement by all tools varies. In general, MetaBAT 2 requires the least while BinSanity requires the most (with a factor of 20x, in constrast to others with about 1-4x).

Running on the entire IMG100 datasets we observed a much more pronounced difference in computational resource requirement. MetaBAT 2 finished binning in a few hours, but other tools took days even

PeerJ Preprints | https://doi.org/10.7287/peerj.preprints.27522v1 | CC BY 4.0 Open Access | rec: 6 Feb 2019, publ: 6 Feb 2019

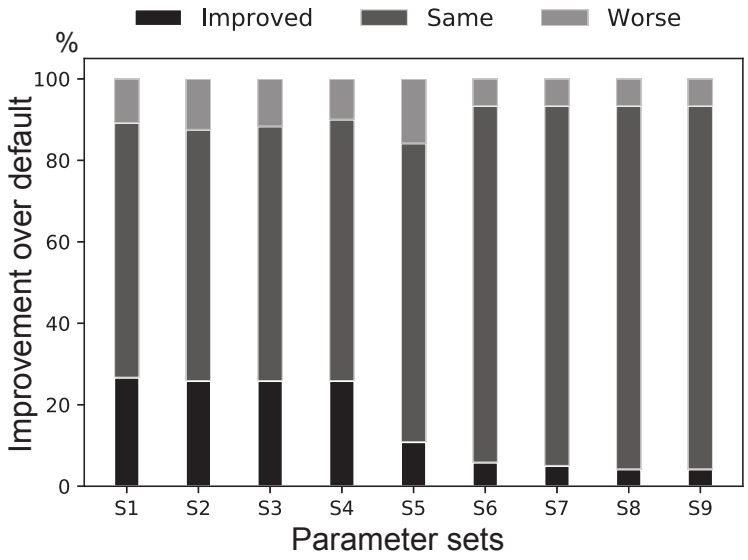


Figure 3. A binning performance comparison between the default parameter set of MetaBAT 2 against several common best parameter sets found by the genetic algorithm. The IMG100 dataset was used for searching for the best parameter set for each sample. For each parameter set (S1 to S9, see supplemental Table S2), a stacked bar shows the percentages of datasets where its performance is better than (darkest gray at the bottom), the same as (medium gray in the middle), or worse than (light gray at the top) the default parameter set. Overall the default parameter set is consistently selected as the best parameter set for most samples.

Runtime Memory	MetaBAT 2	CONCOCT	MaxBin	MyCC	BinSanity	COCACOLA
CAMI High	1min54sec 2.63 GB	2hr40min 1.28 GB	7hr1min 2.99 GB	6hr5min 3.04 GB	5hr42min 50.82 GB	14hr2min 4.95 GB
CAMI Medium	25sec 0.56 GB	1hr21min 0.96 GB	1hr10min 1.04 GB	1hr26min 2.10 GB	2hr3min 25.52 GB	3hr1min 1.21 GB
CAMI Low	7sec 0.16 GB	13min52sec 0.38 GB	11min5sec 0.69 GB	20min25sec 0.82 GB	18min47sec 2.73 GB	30min8sec 0.85 GB

Table 1. Runtime and memory comparison on CAMI high, medium, low dataset. All tests were run on a workstation with 2 CPUs of Intel(R) Xeon(R) CPU E5-2698 v3 @ 2.30GHz, 128 GB Memory.

weeks, if they can finish. Some tools failed on some of the large assemblies after a long time probably due to their high memory requirement.

DISCUSSION

In conclusion, we show that the adoption of a new adaptive binning algorithm makes MetaBAT 2 automatically adapt to datasets with various characteristics and provides robust metagenome binning. This should greatly reduce users' time needed to manually explore the underlying datasets and experiment with different parameter sets. This capability should be particularly useful for datasets derived from unknown complex microbial communities, as empirically setting parameters might be challenging. Extensive low-level computational optimization taking advantage of the underlying hardware capabilities also makes MetaBAT 2 run very efficiently, and makes it scalable for very large datasets.

There are a couple of considerations when using MetaBAT 2. First of all, MetaBAT 2 uses an adaptive binning algorithm which puts more weight on abundance but less weight on TNF, which makes it work well on datasets with multiple samples (CAMI synthetic sets, e.g.). In general we expect more samples to produce better accuracy. For single-sample datasets, some users reported a genome can sometimes be

PeerJ Preprints | https://doi.org/10.7287/peerj.preprints.27522v1 | CC BY 4.0 Open Access | rec: 6 Feb 2019, publ: 6 Feb 2019

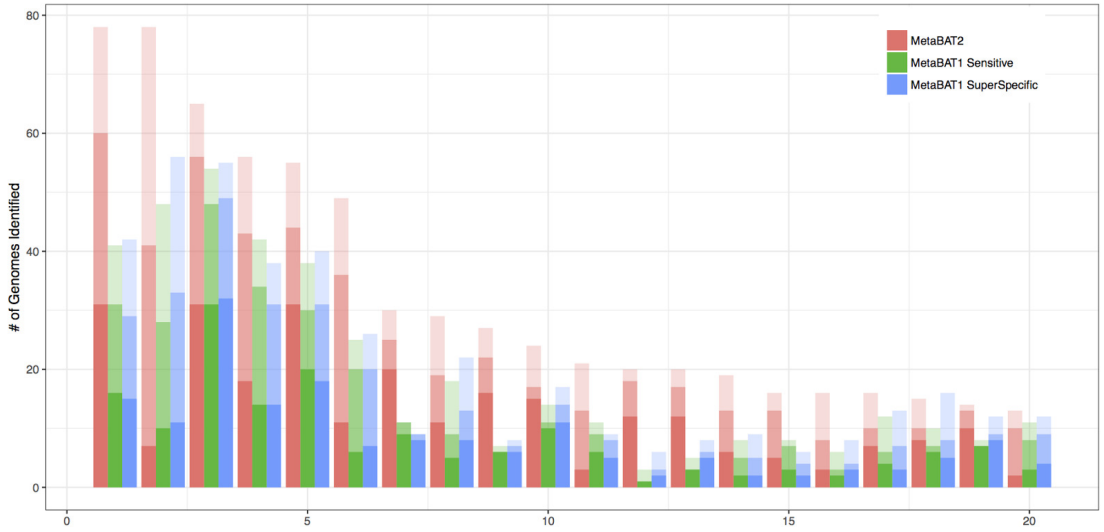


Figure 4. Comparing MetaBAT 2 with two sets of MetaBAT 1 binning experiments using real world metagenomes. IMG 100 dataset was used for benchmarking experiment. Top 20 metagenomes ordered by the number of genome bins identified are shown. X-axis represents each metagenome, and Y-axis shows the number of genome bins identified using 5% contamination cutoff. Each bar represents 3 completeness results of 90%, 70%, and 50% by the order of color density (i.e. darkest color represents 90%). The completeness and contamination were estimated by CheckM. MetaBAT 2 outperforms both modes of MetaBAT 1 in most cases.

split among different bins in spite of consistent TNF composition. This illustrates that MetaBAT 2 weighs heavily on purity with some sacrifice in completeness. A manual, post-binning polishing step may be required to further improve completeness. A second consideration is that MetaBAT does not eliminate chimeric contigs or other artifacts from assembly, so binning results from very poor assemblies will not be reliable. Some post-binning polishing process, such as d_2^S Bin (Wang et al., 2017), may help reduce the contamination problem.

The method of normalizing TNF and correlation scores using ABD makes it possible for future MetaBAT versions to incorporate additional scoring matrices in a similar fashion. Interestingly, in a recent preprint, Nissen et al. developed an alternative strategy based on deep variational autoencoders (VAE) to accomplish a similar task and got very good results (Nissen et al., 2018). Additional matrices could be taxonomic similarity, physical linkage (provided by Hi-C experiments or paired end reads), and/or other similarity matrices of the contigs. Incorporating taxonomic information would provide a framework to unite taxonomy dependent and independent binning strategies.

FUNDING

The work was conducted by the US Department of Energy Joint Genome Institute. Dongwan Kang, Edward Kirton, Ashleigh Thomas, Rob Egan, and Zhong Wang's work was supported by the U.S. Department of Energy, Office of Science, Office of Biological and Environmental Research under Contract No. DE-AC02-05CH11231. Feng Li was supported by a exchange student fellowship from China Scholarship Council (CSC).

REFERENCES

Alneberg, J., Bjarnason, B. S., De Bruijn, I., Schirmer, M., Quick, J., Ijaz, U. Z., Lahti, L., Loman, N. J., Andersson, A. F., and Quince, C. (2014). Binning metagenomic contigs by coverage and composition. *Nature methods*, 11(11):1144.

Bahram, M., Hildebrand, F., Forslund, S. K., Anderson, J. L., Soudzilovskaia, N. A., Bodegom, P. M., Bengtsson-Palme, J., Anslan, S., Coelho, L. P., Harend, H., et al. (2018). Structure and function of the global topsoil microbiome. *Nature*, 560(7717):233.

PeerJ Preprints | https://doi.org/10.7287/peerj.preprints.27522v1 | CC BY 4.0 Open Access | rec: 6 Feb 2019, publ: 6 Feb 2019

Bowers, R. M., Kyrpides, N. C., Stepanauskas, R., Harmon-Smith, M., Doud, D., Reddy, T., Schulz, F., Jarett, J., Rivers, A. R., Eloe-Fadrosh, E. A., et al. (2017). Minimum information about a single amplified genome (misag) and a metagenome-assembled genome (mimag) of bacteria and archaea. *Nature biotechnology*, 35(8):725.

Chen, I.-M. A., Chu, K., Palaniappan, K., Pillay, M., Ratner, A., Huang, J., Huntemann, M., Varghese, N., White, J. R., Seshadri, R., et al. (2018). Img/m v. 5.0: an integrated data management and comparative analysis system for microbial genomes and microbiomes. *Nucleic acids research*, 47(D1):D666–D677.

Graham, E. D., Heidelberg, J. F., and Tully, B. J. (2017). Binsanity: unsupervised clustering of environmental microbial assemblies using coverage and affinity propagation. *PeerJ*, 5:e3035.

Kang, D. D., Froula, J., Egan, R., and Wang, Z. (2015). Metabat, an efficient tool for accurately reconstructing single genomes from complex microbial communities. *PeerJ*, 3:e1165.

Kang, D. D., Rubin, E. M., and Wang, Z. (2016). Reconstructing single genomes from complex microbial communities. *it-Information Technology*, 58(3):133–139.

Lin, H.-H. and Liao, Y.-C. (2016). Accurate binning of metagenomic contigs via automated clustering sequences using information of genomic signatures and marker genes. *Scientific reports*, 6:24175.

Lowe, T. M. and Eddy, S. R. (1997). trnascan-se: a program for improved detection of transfer rna genes in genomic sequence. *Nucleic acids research*, 25(5):955.

Lu, Y. Y., Chen, T., Fuhrman, J. A., and Sun, F. (2017). Cocacola: binning metagenomic contigs using sequence composition, read coverage, co-alignment and paired-end read linkage. *Bioinformatics*, 33(6):791–798.

Markowitz, V. M., Chen, I.-M. A., Palaniappan, K., Chu, K., Szeto, E., Grechkin, Y., Ratner, A., Jacob, B., Huang, J., Williams, P., et al. (2011). Img: the integrated microbial genomes database and comparative analysis system. *Nucleic acids research*, 40(D1):D115–D122.

Markowitz, V. M., Chen, I.-M. A., Palaniappan, K., Chu, K., Szeto, E., Pillay, M., Ratner, A., Huang, J., Woyke, T., Huntemann, M., et al. (2013). Img 4 version of the integrated microbial genomes comparative analysis system. *Nucleic acids research*, 42(D1):D560–D567.

Nissen, J. N., Sonderby, C. K., Armenteros, J. J. A., Groenbech, C. H., Nielsen, H. B., Petersen, T. N., Winther, O., and Rasmussen, S. (2018). Binning microbial genomes using deep learning. *bioRxiv*, page 490078.

Nurk, S., Meleshko, D., Korobeynikov, A., and Pevzner, P. A. (2017). metaspades: a new versatile metagenomic assembler. *Genome research*, 27(5):824–834.

Parks, D. H., Imelfort, M., Skennerton, C. T., Hugenholtz, P., and Tyson, G. W. (2015). Checkm: assessing the quality of microbial genomes recovered from isolates, single cells, and metagenomes. *Genome research*, 25(7):1043–1055.

Parks, D. H., Rinke, C., Chuvochina, M., Chaumeil, P.-A., Woodcroft, B. J., Evans, P. N., Hugenholtz, P., and Tyson, G. W. (2017). Recovery of nearly 8,000 metagenome-assembled genomes substantially expands the tree of life. *Nature microbiology*, 2(11):1533.

Pasolli, E., Asnicar, F., Manara, S., Zolfo, M., Karcher, N., Armanini, F., Beghini, F., Manghi, P., Tett, A., Ghensi, P., et al. (2019). Extensive unexplored human microbiome diversity revealed by over 150,000 genomes from metagenomes spanning age, geography, and lifestyle. *Cell*.

Rinke, C., Rubino, F., Messer, L. F., Youssef, N., Parks, D. H., Chuvochina, M., Brown, M., Jeffries, T., Tyson, G. W., Seymour, J. R., et al. (2018). A phylogenomic and ecological analysis of the globally abundant marine group ii archaea (ca. poseidoniales ord. nov.). *The ISME journal*, page 1.

Sczyrba, A., Hofmann, P., Belmann, P., Koslicki, D., Janssen, S., Dröge, J., Gregor, I., Majda, S., Fiedler, J., Dahms, E., et al. (2017). Critical assessment of metagenome interpretation—a benchmark of metagenomics software. *Nature methods*, 14(11):1063.

Strous, M., Kraft, B., Bisdorf, R., and Tegetmeyer, H. (2012). The binning of metagenomic contigs for microbial physiology of mixed cultures. *Frontiers in microbiology*, 3:410.

Tringe, S. G. and Rubin, E. M. (2005). Metagenomics: Dna sequencing of environmental samples. *Nature reviews genetics*, 6(11):805.

Van Dijk, E. L., Auger, H., Jaszczyszyn, Y., and Thermes, C. (2014). Ten years of next-generation sequencing technology. *Trends in genetics*, 30(9):418–426.

Wang, Y., Wang, K., Lu, Y. Y., and Sun, F. (2017). Improving contig binning of metagenomic data using d 2 s {d} _2^ s oligonucleotide frequency dissimilarity. *BMC bioinformatics*, 18(1):425.

Wu, Y.-W., Simmons, B. A., and Singer, S. W. (2015). Maxbin 2.0: an automated binning algorithm to

PeerJ Preprints | https://doi.org/10.7287/peerj.preprints.27522v1 | CC BY 4.0 Open Access | rec: 6 Feb 2019, publ: 6 Feb 2019

recover genomes from multiple metagenomic datasets. *Bioinformatics*, 32(4):605–607.

Zhu, X. and Ghahramani, Z. (2002). Learning from labeled and unlabeled data with label propagation.

PeerJ Preprints | https://doi.org/10.7287/peerj.preprints.27522v1 | CC BY 4.0 Open Access | rec: 6 Feb 2019, publ: 6 Feb 2019