MetaProb 2: Metagenomic Reads Binning based on Assembly using Minimizers and K-mers Statistics *

F. Andreace[†], C. Pizzi*, M.Comin*[‡]

Keywords: metagenomic binning, minimizers, k-mers statistics

Abstract: Current technologies allow the sequencing of microbial communities directly from the environment without prior culturing. One of the major problems when analyzing a microbial sample is to taxonomically annotate its reads to identify the species it contains. The major difficulties of taxonomic analysis are the lack of taxonomically related genomes in existing reference databases, the uneven abundance ratio of species, and sequencing errors. Microbial communities can be studied with reads clustering, a process referred to as genome binning.

*This is the accepted version of the following article: [F.Andreace, C.Pizzi, M.Comin MetaProb 2: Metagenomic Reads Binning based on Assembly using Minimizers and K-mers Statistics Journal of Computational Biology, 28(11), 1052-1062, 2021], which has now been formally published in final form at [Journal of Computational Biology] at [https://doi.org/10.1089/cmb.2021.0270]. This original submission version of the article may be used for non-commercial purposes in accordance with the Mary Ann Liebert, Inc., publishers' self-archiving terms and conditions.

[†]Department of Information Engineering, University of Padova, Padova, 35131,Italy

[‡]To whom correspondence should be addressed: `comin@dei.unipd.it`

In this paper we present MetaProb 2 an unsupervised genome binning method based on reads assembly and probabilistic k-mers statistics. The novelties of MetaProb 2 are the use of minimizers to efficiently assemble reads into unitigs and a community detection algorithm based on graph modularity to cluster unitigs and to detect representative unitigs. The effectiveness of MetaProb 2 is demonstrated in both simulated and real datasets in comparison with state-of-art binning tools such as MetaProb, AbundanceBin, Bimeta and MetaCluster. On real datasets, it is the only one capable of producing promising results while being parsimonious with computational resources.

Code: https://github.com/frankandreace/metaprob2 [1]

1 Introduction

Metagenomics is the study of the heterogeneous microbes samples (e.g. soil, water, human microbiome) directly extracted from the natural environment with the primary goal of determining the taxonomical identity of the microorganisms residing in the samples (Staley and Konopka, 1985). Shifting the focus from the individual microbe study to a complex microbial community is a revolutionary milestone. The classical genomic-based approaches require the prior clone and culturing for further investigation (Felczykowska et al., 2012; Mande et al., 2012). However, not all bacteria can be cultured. The advent of metagenomics allowed researchers to overcome this difficulty. Microbial communities can be analyzed and compared through the detection and quantification of the species they contain (Kang et al., 2015; Qian and Comin, 2019; Pellegrina et al., 2020).

[1] A preliminary version of this work appeared in the proceedings of ICCABS 2020 (Andreace et al., 2021)

In this paper, we will focus on the unsupervised detection of species in a sample without the use of reference genomes. Despite extensive studies, accurate binning of reads remains challenging (Sczyrba et al., 2017; Comin et al., 2020). Supervised methods require to index a database of reference genomes, e.g. the NCBI/RefSeq databases of bacterial genomes, that is used to classify (Wood and Salzberg, 2014; Ounit et al., 2015; Qian et al., 2018; Marchiori and Comin, 2017; Segata et al., 2012). Although the reads classification is very efficient, the construction of k-mers DB usually is very demanding, requiring computing capabilities with large amounts of RAM and disk space. Another drawback is the fact that most bacteria found in environmental samples are unknown and cannot be cultured and separated in the laboratory (Eisen, 2007). As a consequence, the genomes of most microbes in an environmental sample lack a taxonomically related sequences in existing reference databases. For these reasons, when using supervised methods the number of unassigned reads can be very high (Lindgreen et al., 2015; Girotto et al., 2017a; Storato and Comin, 2020).

Unsupervised methods do not require to know all the genomes in the sample, instead they try to divide the reads into groups so that reads from the same species are clustered together. Unsupervised classification tools, also known as genome binning, are based on the observation that the k-mer distributions of the DNA fragments from the same genome are more similar than those from different genomes. Thus, without using any reference genome, one can determine if two fragments are from genomes of similar species based on their k-mer distributions. The major problem when processing metagenomic data is the fact that the proportion of species in a sample, a.k.a. abundance rate, can vary greatly. Most of the tools can only handle species with even abundance ratios, and their binning performances degrade significantly in real situations when the abundance ratios of the species are different. For example, AbundanceBin (Wu

and Ye, 2011) works well only with a limited number of species and with very different abundance ratios, but problems arise when some species have similar abundance ratios. Other tools like BiMeta (Vinh et al., 2015) and MetaCluster (Wang et al., 2012) try to group the reads into many small clusters so that reads from minority species (with low abundance ratios) could exist as isolated clusters. Both these methods use as means of comparison the Euclidean distance between the vectors of k-mers counts on the clusters groups. In MetaProb (Girotto et al., 2016) reads are clustered based on a self-standardized statistic, derived from alignment-free statistics, that is not dominated by the noise in the individual sequences, and that can compare groups of reads with different abundance ratios. The sensitivity can be improved by using spaced seeds instead of k-mers (Girotto et al., 2017b), however at the expenses of the computing resources.

In terms of precision Metaprob has shown to be one of the best performing methods, however the major bottleneck is the high memory consumption. Another important observation is that all reads binning methods try to cluster reads, based on overlaps and k-mers counts, but without assembling the reads. A possible explanation is because metagenomics reads assembly is very challenging (Sczyrba et al., 2017). However, efficient techniques based on minimizers have been recently devised for long reads mapping and assembly (Li, 2018, 2016). Recently, GraphBin (Mallawaarachchi et al., 2020) has shown that assembly can be of help also for the problem of contig binning.

In this paper, we present MetaProb 2, a new approach to address the problem of unsupervised metagenomics reads binning. To this purpose, MetaProb 2 assembles reads into unitigs using efficient techniques based on minimizers, as well as probabilistic sequence signatures based on k-mers. The use of unitigs will also prevent the overestimation of k-mers frequency, and it does not re-

quire complex counting procedures like finding sets of independent reads as in MetaProb (Girotto et al., 2016). Another novelty of MetaProb 2 is a community detection algorithm based on graph modularity (Blondel et al., 2008) to cluster unitigs and to detect putative species. This novel paradigm exploited by MetaProb 2 will further improve the classification accuracy while reducing the computational resources. This is particularly important because not all the tools are able to handle large real datasets (see Section 3).

2 Methods

The study of DNA based on its k-mers is a well-known technique to identify the species in a metagenomic sample. One drawback of this approach is the large amount of memory required to compute reads overlaps and to store all the k-mers of the sequences. To solve these issues, we propose MetaProb 2, a new metagenomic reads binning algorithm based on minimizers. This algorithm uses short paired-end reads to infer the number of species and the abundance in the sample: short reads provide high accuracy and the paired-end information will be useful to improve the precision and overall performances of the algorithm.

An overview of MetaProb 2 can be found in Figure 1. The method consists of three main steps. In the first phase, reads are grouped together based on their overlap, using minimizers instead of k-mers. Since these reads share a common subsequence, they are assumed to belong to the same species and assembled together to generate an unitig, i.e. a precise contig in which the consensus is unambiguous. These operations are performed using two long reads de-novo assembly algorithms, Minimap2 (Li, 2018) and Miniasm (Li, 2016), with some additional modifications to comply with the short reads input.

In the second phase a unitig graph is built considering the unitigs - and their associated reads - as nodes. From this graph, it is possible to infer communities

of nodes that will likely represent unitigs of the same species. The third and last step is the identification of putative species and the estimation of their abundances. In this phase, the representative unitigs and the unassembled reads are clustered together based on k-mers content using a probabilistic sequence signature derived from MetaProb (Girotto et al., 2016). Next, a more detailed description of each of these steps is given.

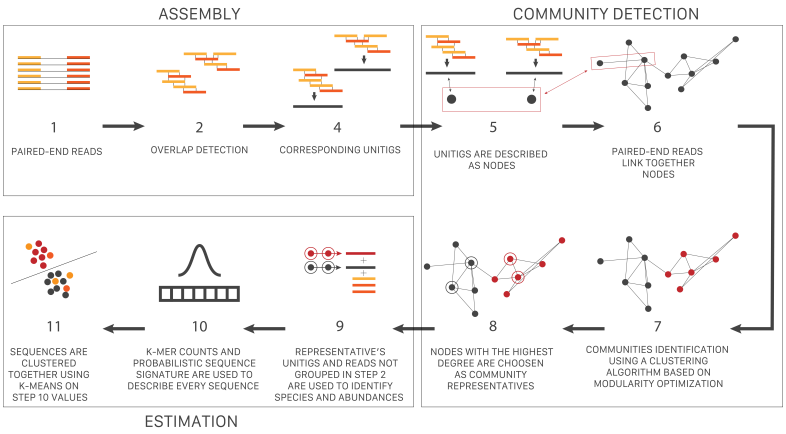

Figure 1: An overview of MetaProb 2 divided into the three main phases: Unitig Construction, Community Detection and Species Identification.

2.1 Phase 1: Unitig Construction

In the first phase, reads are grouped together, based on their overlaps, and then assembled. This operation is performed using Minimap2 (Li, 2018), a long-read de novo mapping tool that uses minimizers instead of k-mers to find shared subsequences between reads. The use of minimizers is crucial because it stores only a fraction of all the k-mers to perform the all vs. all comparison between the sequences, resulting in faster computation and lower memory usage. In fact, Minimap2 has the best performances in long reads mapping and assembling. Unlike MetaProb, the k-mer length is set to 15 and not 32, which is a good trade-off between resource usage, precision and the number of reads grouped:

higher k-mer length means worst performances in computation time, memory usage and grouped reads but it guarantees higher precision. Regarding the window size in which minimizers are chosen we used 10, as the recommended value was 2/3 of the k-mer length (Li, 2018).

Instead of working on the groups of overlapping reads, we assemble the sequences in each group and we consider the resulting unitig. Unitigs are precise assemblies generated from overlapping sequences: we decided to not combine them together into contigs to preserve the high quality of the assembled reads, since our purpose is to have the more precise information as possible. Moreover, the fact of considering unitigs instead of groups of reads will naturally resolve the problem of k-mers overcounting, and also it will avoid the complex phase of finding sets of independent reads of MetaProb (Girotto et al., 2016). The benefit of using minimizers for short reads assembly has been recently shown in (Bayat et al., 2020). Miniasm (Li, 2016) is a tool often used together with Minimap2 that performs assembly on long reads, it provides as output the unitig sequences along with other information. As suggested by (Bayat et al., 2020), we change the default parameters of minimap2 and miniasm to accommodate for short reads assembly.
Note that not all the reads in the input sample will contribute to the assembly of some unitigs, however, they will be considered in the final phase. The portion of these left-out reads is mostly due to the assembly step, that takes out reads not useful to the unitig generation.

We consider the reads used to create an unitig as components of a cluster represented by the unitig sequence. Their precision is assessed around 99,77% for the most complex synthetic dataset. Although being really precise, the assembly step is not enough to condensate together the reads into significant chunks of DNA sequences. To compensate for that, before moving to the species

identification using k-mer frequencies, an intermediate step is needed.

2.2 Phase 2: Community Detection

In this phase, we use the information provided by the overlap detection together with the paired-end structure of the reads to group unitigs that are likely to be from the same species. To do so every unitig is assigned to a node in a graph and if two unitigs share part of a paired-end read, their respective nodes are linked together. Every edge is weighted with the number of shared paired-end reads between the unitigs. The resulting network is really precise: let's consider an edge correct if it links unitigs that are made of (almost entirely) reads from a certain species. For example, on the most complex synthetic dataset 99,74% of edges are correct. Then we use a graph clustering algorithm on this network to detect communities of unitigs. Since the dimension of this graph can be large, i.e. millions of nodes, this operation is performed using a heuristic method based on modularity optimization. Modularity is a measure that describes how well a network is divided into meaningful communities: networks with high modularity can be divided into clusters of densely connected nodes, with the nodes of different clusters being sparsely connected. In practice, it describes how well communities are connected in the network compared to what would be expected if edges were placed at random. The optimization problem of community detection requires the network to be split up into the communities that give the highest value of modularity. This problem is known to be computationally intractable (Brandes et al., 2006), however suboptimal algorithms exist. We used the library scikit-network (Bonald et al., 2020) that is based on the Louvain algorithm (Blondel et al., 2008). This method is extremely efficient both in time and memory and it can handle very large graphs. Moreover, this operation relies on the assumption that unitigs that share many paired-end reads are likely to

be originated from the same species. It is important to notice that the communities we obtain are very precise, as the reads they contain are almost all from the same species: on the most complex synthetic dataset, the communities have precision of 99,3%. However, a given community does not necessarily contain all the reads from a species. It may well be that two or more communities are composed of reads of the same species. This calls for an additional step based on the sequence statistics, that will have the specific purpose to detect the real number of species and their abundance in the sample.

Once the communities of unitigs have been created, we selected from every community the nodes with the highest degrees, and these unitigs will be considered as representatives for that community in the last phase. Because only a small number of unitigs are selected, this operation has the advantage to require fewer calculations in the next phase, speeding it up and lowering its memory usage. In particular, we chose the nodes with the highest degrees because they will somehow better represent the community while avoiding the possibility of choosing an outlier.on the most complex synthetic dataset, more than 99,91% of the representatives chosen were from the same species of the majority of the reads they were representing. In order to limit the number of representative unitigs we set a threshold on the sum of the representative's sequence length. The representative unitigs of each community are used in the last phase in place of all the reads belonging to that community, making the species identification step faster while keeping the sequence information useful to estimate the number of species.

2.3 Phase 3: Species Identification

In the last phase, we infer the number of species and their abundance in the sample from the sequence information, using sequence signatures (Girotto et al.,

2016) based on k-mer statistics. Several alignment-free statistics have been proposed over the years (Zielezinski et al., 2017, 2019; Apostolico et al., 2016). In the context of metagenomic binning, the probabilistic sequence signatures proposed by MetaProb (Girotto et al., 2016) have shown very good performance and we decided to use the sequence signature for the final phase. To keep the paper self-contained here below we summarize the probabilistic sequence signature procedure.

Let I be the set of input sequences for the species identification step, where I is composed of the representative unitigs detected in Phase 2, which account for all the reads in the communities, and by the remaining unassembled reads. Let S be a sequence (either an unitig or an unassembled read) from the set I, we call S_w the the number of occurrences of the k-mer w in S. To account for the different probability of appearance of k-mers, the k-mers counts are standardized based on the probability of k-mers in each sequence. If $k << |S|$, we can consider the variables S_w, with $S \in I$, as Bernoulli. We used $k = 4$, so that such assumption holds, as in other binning methods (Vinh et al., 2015; Wang et al., 2012). The probability P_w of the k-mer w to occur in the sequence S was computed as in Metaprob (Girotto et al., 2016) for short reads. We can now define the mean and variance of S_w:

$$E[S_w] = \mu_i^w = P_w \times (|S| - k + 1)$$

$$Var[S_w] = (\sigma_w)^2 = P_w \times (1 - P_w) \times (|S| - k + 1)$$

that are used to standardize the variable S_w:

$$\tilde{S}_w = \frac{S_w - \mu_w}{\sigma_w}$$

Finally, in order to compare sequences of different length, the probabilistic

sequence signatures are computed for each input sequence S. The sequence signature is a vector in Σ^k, with $k = 4$, containing the normalized frequency count of each word w in S:

$$f_w^S = \frac{\tilde{S}_w}{\sqrt{\sum_{v \in \Sigma^k} (\tilde{S}_v)^2}}$$

In order to detect sequences that are likely to belong to the same species we evaluate the distance between the sequence signatures, and we apply k-means to group together sequences with a similar distribution as in MetaProb (Girotto et al., 2016).

3 Discussion

In this section, we describe several experiments we performed to assess the performances of MetaProb 2. In particular, we measured both the quality of the results and the computational resource usage in terms of time and space required for the processing. All the experiments were performed on a machine with Intel(R) Xeon(R) Gold 5220 CPUs @ 2.20/3.90GHz and 2TB of RAM.

3.1 Datasets description

We used four different kinds of datasets: ten simulated bacterial metagenomes generated using MetaSim (Richter et al., 2008), called S1-10, two containing synthetic metagenomes based on real reads, called MIX1-2, two datasets that closely mimic the complexity, size and characteristics of real data, called SetA2 and SetB2, and a real dataset, SRR1804065.

3.1.1 Simulated Datasets

The S1-10 datasets were used in previous studies to assess the performances of BiMeta (Vinh et al., 2015) and MetaProb (Girotto et al., 2016). Mix1-2 were also used to validate MetaProb and Kraken.

The S datasets contain short paired-end reads, which length is approximately 80 bp, generated according to the Illumina error profile with an error rate of 1% using MetaSim. These have been used to verify the consistency of this method in different scenarios: from datasets like S1-4 that have only 2 different species and hundreds of thousands of reads with similar abundances to S9-10 that have 15 and 30 species, between 2.3 and 5 million reads and different abundance ratios and different phylogenetic distance. The synthetic datasets, constructed from real metagenomic data are composed of short reads Illumina MiSeq from Kraken (Wood and Salzberg, 2014) with 10 different species and two abundance profiles: spanning between 3.5 to 5 million reads. Table 1 shows the number of reads, species and phylogenetic distance for each dataset.

Dataset	No. of reads	No. of species	Phylogenetic distance
S1	96367	2	Species
S2	195339	2	Species
S3	338725	2	Order
S4	375302	2	Phylum
S5	325400	3	Species and Family
S6	713388	3	Phylum and Kingdom
S7	1653550	5	Genus and Order
S8	456224	5	Genus and Order
S9	2234168	15	various distances
S10	4990632	30	various distances
MIX1	4814943	10	various distances
MIX2	3574950	10	various distances

Table 1: Number of reads, species and phylogenetic distance of each simulated dataset.

3.1.2 Real Datasets

The real datasets are more complex than the synthetic and simulated ones. Datasets SetA2 and SetB2 are based on real sequencing results of pooling 6 soil samples on a single HiSeq2000 lane. The original datasets were generated by (Lindgreen et al., 2016), and used by (Guerrini et al., 2020) for metagenomic classification. Since in (Lindgreen et al., 2016) these datasets were used to evaluate tools at the genus and phylum level, we decided to validate them at the genus level. These two datasets have almost 20 million reads each and 444 genera with different abundances at various phylogenetic distances. The genera in these samples are uneven: for example, in SetA2 there is one with more than 1.3 million reads, there are 34 with between 100000 and 1 million reads each and there are some that have few thousands or fewer reads. Although containing the same genera, SetA2 and SetB2 have different abundance ratios. SRR1804065 is a real stool sample from the Human Microbiome Project, generated using Illumina that originally contains 21873781 reads. Since this is a real dataset, the ground truth is not available: all the 22 million reads were mapped using BLAST and the ones not mapping to a bacterial genome were filtered out. The resulting dataset has more than 5.5 million reads. The abundance ratios of this dataset are very skewed: 2.7 million reads, i.e. more than 49% of the sample, are from the Bacteroides genus, known for being the most substantial portion of the mammalian gastrointestinal microbiota; the second one is Phocaeicola, another relevant bacterial genus present in the gut and colon. Almost 75% of the dataset is composed of these two genera. All the other genera in the sample have at least an order of magnitude less reads and only 69 genera have more than 1000 reads. In Table 2 is presented a summary of the real datasets.

Dataset	No. of reads	No. of genera	Phylogenetic distance
SetA2	19724719	444	various distances
SetB2	18523723	444	various distances
SRR1804065	5500983	69	various distances

Table 2: Number of reads, genera and phylogenetic distance of the real datasets.

3.2 Performance evaluation metrics

In order to evaluate the results, we used three performance evaluation metrics: precision, recall and f-measure. As the provenance of the synthetic and simulated reads is known, given n as the number of species in a dataset and C the number of clusters returned by the algorithm, A_{ij} is the number of reads from species j assigned to cluster i. We used the same definitions of precision, recall and f-measure as in MetaProb (Girotto et al., 2016) and BiMeta (Vinh et al., 2015):

$$\text{Precision} = \frac{\sum_{i=1}^{C} \max_j A_{ij}}{\sum_{i=1}^{C} \sum_{j=1}^{n} A_{ij}} \tag{1}$$

$$\text{Recall} = \frac{\sum_{j=1}^{n} \max_i A_{ij}}{\sum_{i=1}^{C} \sum_{j=1}^{n} A_{ij} + \#\text{unassigned reads}} \tag{2}$$

$$\text{F-measure} = \frac{2 * \text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}} \tag{3}$$

The F-measure is the harmonic mean of precision and recall. We evaluated also the total computation time and the peak memory used by the algorithms as well.

3.3 Results

In this section we discuss the results of the comparison between MetaProb 2 and MetaProb, alongside with other algorithms like MetaCluster 5.0.1 (Wang et al., 2012), AbundanceBin (Wu and Ye, 2011) and BiMeta (Vinh et al., 2015) on simulated and real datasets.

3.3.1 Quality of Binning: Simulated Datasets

The experiments on synthetic and simulated datasets had the purpose of measuring the ability of MetaProb 2 to perform metagenomic binning compared against the performances of its predecessor, MetaProb, along with MetaCluster, AbundanceBin and BiMeta. Table 3 shows the overall F-measure values of all the algorithms for each dataset (S1-10, MIX1-2).

Dataset	Abundance Bin	MetaCluster	BiMeta	MetaProb	MetaProb 2
S1	0.683	0.672	0.978	0.992	**0.994**
S2	0.713	0.631	0.588	**0.879**	0.83
S3	0.824	0.415	0.847	0.920	**0.957**
S4	0.883	0.460	0.992	0.916	**0.997**
S5	0.552	0.643	0.781	0.828	**0.880**
S6	0.692	0.492	0.993	0.953	**0.997**
S7	0.606	0.652	0.705	0.774	**0.85**
S8	0.528	0.529	0.732	0.769	**0.874**
S9	Error	0.639	0.761	0.718	**0.842**
S10	Error	0.052	0.636	0.713	**0.736**
MIX1	Error	0.555	0.713	**0.868**	0.835
MIX2	0.645	0.630	0.707	0.775	**0.824**
AVERAGE	0.68	0.531	0.786	0.842	**0.879**

Table 3: The comparison of F-measure for all algorithms on all simulated and synthetic datasets.

We observed that increasing the dataset complexity (number of species, different abundances) results in a decrease in performances for every algorithm. While AbundanceBin and MetaCluster have significantly lower performances

than the others, BiMeta and MetaProb have overall good performances and perform really well on specific datasets. AbundanceBin reported results with no read clustered or failed its execution on datasets S9, S10 and MIX1.

MetaProb 2 resulted as the best tool on 10 out of 12 datasets, in particular, it outperforms all the other algorithms on the most difficult datasets, except for MIX1, where the result is similar to MetaProb, the best on that sample. It is important to notice that the best improvements in terms of overall binning quality (F-measure) have been made for the most complex datasets. MetaProb 2 F-measure values are between 5% and 15% better than the next best BiMeta and MetaProb.

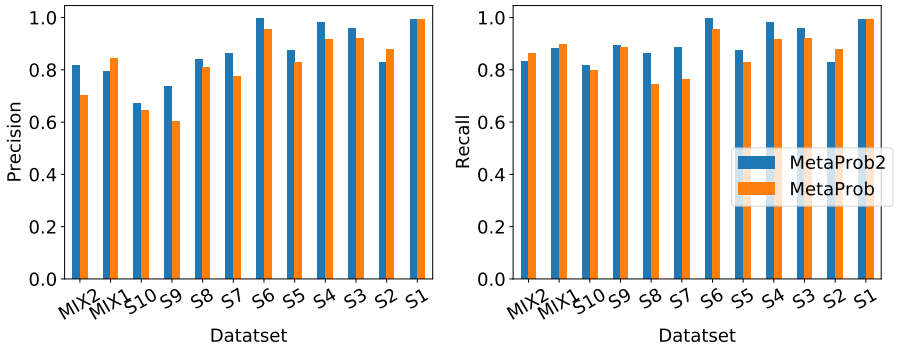

Figure 2: Precision and recall comparison between MetaProb and MetaProb 2.

Precision and Recall values for MetaProb and MetaProb 2 are shown in details in Figure 2, as they were the two best performing methods. Both algorithms have balanced levels of precision and recall in all datasets. MetaProb 2 obtains in most cases better performance than MetaProb in terms of both precision and recall. Results show very high values even for the synthetic datasets (MIX1 and MIX2), and consistent with the most complex among the simulated datasets. These results show that the probabilistic sequence signature introduced in MetaProb is a powerful tool and that the two new phases of assembly

and unitig clustering that have been introduced in MetaProb 2 strengthen it even further.

3.3.2 Quality of Binning: Real Datasets

Real datasets are more difficult to analyze and evaluate, mainly because of the dimension, the complexity of the sample, and the number of genera present. All other tools failed to run or they could not run in a reasonable time, even after 4 days of computation, compared to few hours of MetaProb and MetaProb 2, therefore they are not considered suitable to analyze these large real datasets. Moreover, since in real experiments it is more important to have a broad picture of the sample composition rather than to classify every read, during the MetaProb 2 tests, we decided to not consider the reads not assembled in the first phase. To evaluate the output of MetaProb and MetaProb 2, we assessed the number of different genera present in the clusters, the number of different genera present in the top 30 clusters by dimension, i.e. number of reads, and the number of clusters in the top 30 with precision higher or equal than 70%.

Dataset	Tool	Different genera detected	Different genera detected in top 30 clusters	Clusters with precision $> 70\%$ in top 30
SetA2	MetaProb 2	85	28	22
	MetaProb	70	19	7
SetB2	MetaProb 2	88	29	16
	MetaProb	74	18	4
SRR1804065	MetaProb 2	15	9	6
	MetaProb	1	1	1

Table 4: Real datasets results: different genera detected in the sample and in the top 30 cluster, and the number of clusters with precision higher than 70%.

The results, shown in Table 4, clearly display the ability of MetaProb 2 to better understand the complexity of the samples. In the SetA2 and SetB2 datasets, MetaProb 2 detects more genera than MetaProb. Moreover, on the

top 30 clusters, MetaProb 2 is able to report more clusters of reads belonging to different genera. Also, the number of clusters with high precision in the top 30 is higher compared to MetaProb. In the real SRR1804065 dataset, MetaProb is able to detect only the dominant genera in all the clusters it outputs, whereas MetaProb 2 displays a clearer picture, detecting up to 15 different genera. In the top 30 clusters, MetaProb 2 detects with high precision *Bacteroides*, i.e. a bacteria genera that makes the most substantial part of mammals gastrointestinal microbiota, *Alistipes* and *Phocaeicola*. Moreover, it detects in the top 30 clusters *Parabacteroides* with high precision 93%. All these genera are known to be among the most abundant genera in stool samples (Qin et al., 2010).

3.3.3 Computational Resources

In this section we compared the running time and memory usage of MetaProb 2 with other binning tools.

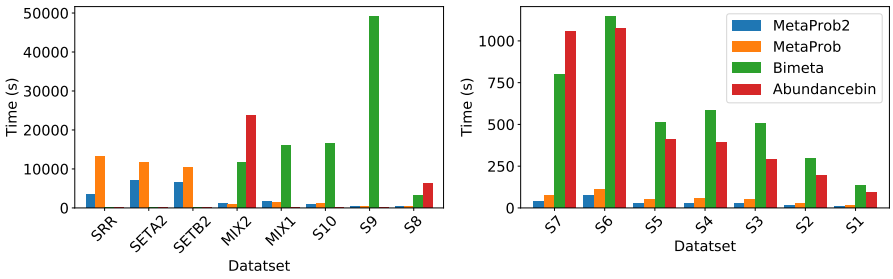

Figure 3: Runtime comparison for all tools on all datasets.

Processing time was not considered an issue: as shown in (Girotto et al., 2016), MetaProb was already an order of magnitude faster than AbundanceBin and BiMeta. Nevertheless, MetaProb 2 is faster than MetaProb on almost every dataset (see Figure 3). This has been possible since reads assembly using minimizer is a fast operation, and the graph clustering algorithm scales well with the dimension of the dataset. Moreover, the other tools fail to run on the

large real datasets.

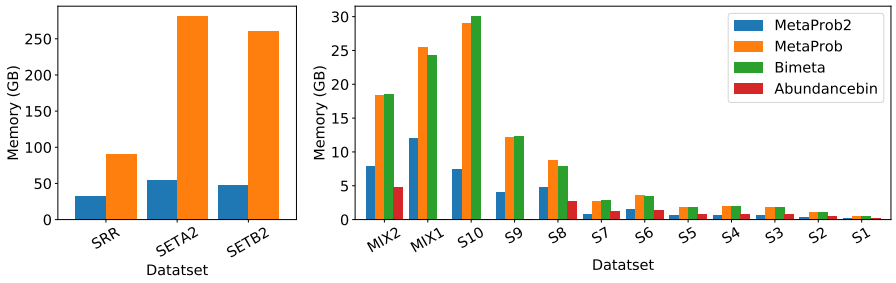

Figure 4: Maximum memory usage comparison for all tools on all datasets.

On the other hand, the heavy memory usage of MetaProb was the driving factor for the development of a new approach. Even if the performances of MetaProb are good, the amount of RAM used can be reduced.

As shown in Figure 4, MetaProb 2 consistently uses less memory than its predecessor, requiring significantly less space as the size of the dataset grows. Its results are comparable with Abundancebin, which has in the low memory usage the best of its strengths. On real datasets, the advantage w.r.t. MetaProb makes a huge difference in the usability of MetaProb 2 on these samples. Even without a machine with hundreds of Gb of RAM it is possible to analyze complex datasets and extract useful insights.

These results have been possible thanks to the use of minimizers that considerably reduce the number of k-mers stored for the overlap detection. Finally, the efficient unitig graph algorithms, and the resulting smaller number of sequences to compare, make then possible to keep the memory usage low during the last phase: in fact, the highest amount of memory usage is always registered in the first phase.

4 Conclusions

Binning metagenomic reads remains a crucial step in the metagenomic analysis. In this work, we presented MetaProb 2, an unsupervised approach for metagenomic reads binning based on reads assembly using minimizers and on probabilistic k-mers statistics. We compared the binning performance over simulated and real metagenomic datasets against other state-of-art binning algorithms. MetaProb 2 achieves good performances in terms of precision and recall, outperforming MetaProb and the other tools. Another advantage of MetaProb 2 are the small requirements of computational resources, especially on large datasets. On real datasets the memory reduction is up to 80% w.r.t. to MetaProb, the only other tool capable of analyzing them. Moreover, on these big and complex datasets, MetaProb 2 is able to detect more genera and cluster reads with more precision and heterogeneity than its predecessor.

References

Andreace, F., Pizzi, C., and Comin, M. (2021). Metaprob 2: Improving unsupervised metagenomic binning with efficient reads assembly using minimizers. In Jha, S. K., Măndoiu, I., Rajasekaran, S., Skums, P., and Zelikovsky, A., editors, *Computational Advances in Bio and Medical Sciences*, pages 15–25, Cham. Springer International Publishing.

Apostolico, A., Guerra, C., Landau, G., and Pizzi, C. (2016). Sequence similarity measures based on bounded hamming distance. *Theoretical Computer Science*, 638:76–90. cited By 17.

Bayat, A., Deshpande, N. P., Wilkins, M. R., and Parameswaran, S. (2020). Fast short read de-novo assembly using overlap-layout-consensus approach.

IEEE/ACM Transactions on Computational Biology and Bioinformatics, 17(01):334–338.

Blondel, V. D., Guillaume, J.-L., Lambiotte, R., and Lefebvre, E. (2008). Fast unfolding of communities in large networks. *Journal of Statistical Mechanics: Theory and Experiment*, 2008(10):P10008.

Bonald, T., de Lara, N., Lutz, Q., and Charpentier, B. (2020). Scikit-network: Graph analysis in python. *Journal of Machine Learning Research*, 21(185):1–6.

Brandes, U., Delling, D., Gaertler, M., Goerke, R., Hoefer, M., Nikoloski, Z., and Wagner, D. (2006). Maximizing modularity is hard.

Comin, M., Di Camillo, B., Pizzi, C., and Vandin, F. (2020). Comparison of microbiome samples: methods and computational challenges. *Briefings in Bioinformatics*. bbaa121.

Eisen, J. A. (2007). Environmental shotgun sequencing: its potential and challenges for studying the hidden world of microbes. *PLoS Biol.*, 5.

Felczykowska, A., Bloch, S. K., Nejman-Faleńczyk, B., and Barańska, S. (2012). Metagenomic approach in the investigation of new bioactive compounds in the marine environment. *Acta Biochimica Polonica*, 59(4):501–505.

Girotto, S., Comin, M., and Pizzi, C. (2017a). Higher recall in metagenomic sequence classification exploiting overlapping reads. *BMC Genomics*, 18(10):917.

Girotto, S., Comin, M., and Pizzi, C. (2017b). Metagenomic reads binning with spaced seeds. *Theoretical Computer Science*, 698:88 – 99. Algorithms, Strings and Theoretical Approaches in the Big Data Era (In Honor of the 60th Birthday of Professor Raffaele Giancarlo).

Girotto, S., Pizzi, C., and Comin, M. (2016). Metaprob: accurate metagenomic reads binning based on probabilistic sequence signatures. *Bioinformatics*, 32(17):i567–i575.

Guerrini, V., Louza, F. A., and Rosone, G. (2020). Metagenomic analysis through the extended burrows-wheeler transform. *BMC Bioinformatics*, 21(8):299.

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.

Li, H. (2016). Minimap and miniasm: fast mapping and de novo assembly for noisy long sequences. *Bioinformatics*, 32(14):2103–2110.

Li, H. (2018). Minimap2: pairwise alignment for nucleotide sequences. *Bioinformatics*, 34(18):3094–3100.

Lindgreen, S., Adair, K., and Gardner, P. (2015). An evaluation of the accuracy and speed of metagenome analysis tools. *Cold Spring Harbor Laboratory Press*.

Lindgreen, S., Adair, K. L., and Gardner, P. P. (2016). An evaluation of the accuracy and speed of metagenome analysis tools. *Scientific Reports*, 6(1):19233.

Mallawaarachchi, V., Wickramarachchi, A., and Lin, Y. (2020). GraphBin: refined binning of metagenomic contigs using assembly graphs. *Bioinformatics*, 36(11):3307–3313.

Mande, S. S., Mohammed, M. H., and Ghosh, T. S. (2012). Classification of metagenomic sequences: methods and challenges. *Briefings in Bioinformatics*, 13(6):669–681.

Marchiori, D. and Comin, M. (2017). Skraken: Fast and sensitive classification of short metagenomic reads based on filtering uninformative k-mers. In *BIOINFORMATICS 2017 - 8th International Conference on Bioinformatics Models, Methods and Algorithms, Proceedings; Part of 10th International Joint Conference on Biomedical Engineering Systems and Technologies, BIOSTEC 2017*, volume 3, pages 59–67.

Ounit, R., Wanamaker, S., Close, T. J., and Lonardi, S. (2015). Clark: fast and accurate classification of metagenomic and genomic sequences using discriminative k-mers. *BMC Genomics*, 16(1):1–13.

Pellegrina, L., Pizzi, C., and Vandin, F. (2020). Fast approximation of frequent k-mers and applications to metagenomics. *Journal of Computational Biology*, 27(4):534–549. PMID: 31891535.

Qian, J. and Comin, M. (2019). Metacon: Unsupervised clustering of metagenomic contigs with probabilistic k-mers statistics and coverage. *BMC Bioinformatics*, 20(367).

Qian, J., Marchiori, D., and Comin, M. (2018). Fast and sensitive classification of short metagenomic reads with skraken. In Peixoto, N., Silveira, M., Ali, H. H., Maciel, C., and van den Broek, E. L., editors, *Biomedical Engineering Systems and Technologies*, pages 212–226, Cham. Springer International Publishing.

Qin, J., Li, R., Raes, J., and et al. (2010). A human gut microbial gene catalogue established by metagenomic sequencing. *Nature*, 464:59–65.

Richter, D., Ott, F., Auch, A., Schmid, R., and Huson, D. (2008). Metasim—a sequencing simulator for genomics and metagenomics. *PloS one*, 3:e3373.

Sczyrba, A., Hofmann, P., and McHardy, A. C. (2017). Critical assessment of

metagenome interpretation—a benchmark of metagenomics software. *Nature Methods*, 14:1063–1071.

Segata, N., Waldron, L., Ballarini, A., Narasimhan, V., Jousson, O., and Huttenhower, C. (2012). Metagenomic microbial community profiling using unique clade-specific marker genes. *Nat Methods*, 9.

Staley, J. T. and Konopka, A. (1985). Measurement of in situ activities of nonphotosynthetic microorganisms in aquatic and terrestrial habitats. *Annual Review of Microbiology*, 39(1):321–346. PMID: 3904603.

Storato, D. and Comin, M. (2020). Improving metagenomic classification using discriminative k-mers from sequencing data. In Cai, Z., Mandoiu, I., Narasimhan, G., Skums, P., and Guo, X., editors, *Bioinformatics Research and Applications*, pages 68–81, Cham. Springer International Publishing.

Vinh, L. V., Lang, T. V., Binh, L. T., and Hoai, T. V. (2015). A two-phase binning algorithm using l-mer frequency on groups of non-overlapping reads. *Algorithms for Molecular Biology*, 10(1):1–12.

Wang, Y., Leung, H. C., Yiu, S. M., and Chin, F. Y. (2012). Metacluster 5.0: a two-round binning approach for metagenomic data for low-abundance species in a noisy sample. *Bioinformatics.*, 28.

Wood, D. and Salzberg, S. (2014). Kraken: ultrafast metagenomic sequence classification using exact alignments. *Genome Biol.*, 15.

Wu, Y. W. and Ye, Y. (2011). A novel abundance-based algorithm for binning metagenomic sequences using l-tuples. *J Comput Biol.*, 18.

Zielezinski, A., Girgis, H., Bernard, G., and et al. (2019). Benchmarking of alignment-free sequence comparison methods. *Genome Biol.*, 20(1):144.

Zielezinski, A., Vinga, S., Almeida, J., and Karlowski, W. M. (2017). Alignment-free sequence comparison: benefits, applications, and tools. *Genome biology*, 18(1):186.