ganon: continuously up-to-date with database growth for precise short read classification in metagenomics

Vitor C. Piro*[1,2], Temesgen H. Dadi[3], Enrico Seiler[3], Knut Reinert[3] and Bernhard Y. Renard†[1]

[1]Bioinformatics Unit (MF1), Robert Koch Institute, Nordufer 20, 13353 Berlin, Germany
[2]CAPES Foundation, Ministry of Education of Brazil, 70040-020, Brasília, DF, Brazil
[3]Department of Mathematics and Computer Science, Freie Universität Berlin, Takustr. 9, 14195 Berlin, Germany

Abstract

The exponential growth of assembled genome sequences greatly benefits metagenomics studies, providing a broader catalog of reference organisms in a variety of environments. However, due to the massive amount of sequences, tools are struggling to manage such data and their frequent updates. Processing and indexing currently available repositories is no longer possible on standard infrastructures and can take days and hundreds of GB of memory even on large servers. Few methods have acknowledged such issues thus far and even though many can theoretically handle large amounts of references, time-memory requirements are prohibitive for a real application. As a result, many tools still rely on static outdated datasets and clearly underperform under the currently available wealth of genome sequences. The content and taxonomic distribution of such databases is also a crucial factor, introducing bias when not properly managed. Motivated by those limitations we created ganon, a k-mer based read classification tool which uses Interleaved Bloom Filters in conjunction with a taxonomic clustering and a k-mer counting-filtering scheme. Ganon provides an efficient method for indexing references and keeping them updated, requiring less than 45 minutes to index RefSeq genome sequences of bacteria, archaea, fungi and virus, covering all diversity currently available in the repository. The tool can keep such indices up-to-date in a fraction of the time necessary to create them allowing researchers to permanently work on the most recent references. Ganon enables querying in those large reference sets, significantly classifying more reads and identifying more species. Using a high complexity set from the CAMI challenge, it shows strongly improved precision (in some cases manifold) while having equal or better sensitivity compared to kraken and centrifuge based in a set of complete (i.e. high quality) reference sequences. When querying the same data against the whole RefSeq, ganon improved the F1-Score by 38% at genus level. Ganon supports taxonomy and assembly level classification as well as multiple indices for a hierarchical classification. The software is open-source and available at: https://gitlab.com/rki_bioinformatics/ganon

1 Introduction

Reference- and taxonomy-based short read classification is a fundamental task in metagenomics. Defining the origin of each read from an environmental sample, which can be done during [1] or after sequencing, is usually the first step prior to abundance estimation, profiling and assembly. Over the last years many tools have been specifically developed for this task with different strategies [2, 3, 4, 5, 6] to achieve good performance classifying a large amount of short reads against a predefined and static set of reference sequences. Many of those approaches use taxonomy based classifications

*PiroV@rki.de
†RenardB@rki.de

[7] providing a backbone arrangement of already obtained sequences to better understand the composition of samples.

Due to advances in genome sequencing, improvements in read quality, length, and coverage and also better algorithms for genome assembly, the amount of complete or draft genomic sequences in public repositories is growing fast (Figure 1). In addition, many partial and some complete genome sequences are coming directly from metagenome-assembled genomes [8, 9, 10], a technique which is boosting the growth of public repositories. This enormous amount of references poses a big challenge for current tools which, in general, were not designed to deal with such amounts of data [11]. The problem that was at first mainly the speed of short read classification is now also shifting towards managing the huge reference sizes and their frequent updates [12].

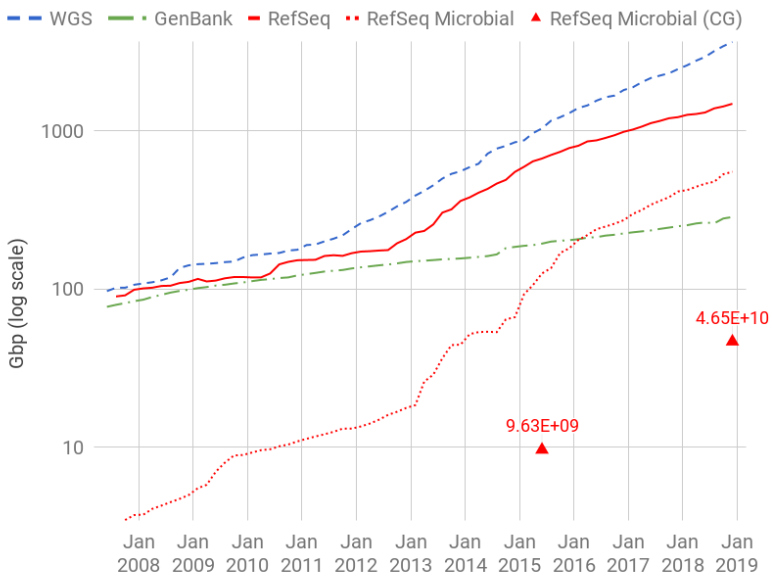

Figure 1: **Number of available sequences in NCBI repositories from June 2007 to December 2018 on a logarithmic scale.** Microbial stands for Archaeal and Bacterial organisms and CG stands for Complete Genomes. RefSeq Microbial has an uninterrupted and linear growth on a logarithmic scale. Data collected from: https://ftp.ncbi.nlm.nih.gov/refseq/release/release-statistics/ and https://www.ncbi.nlm.nih.gov/genbank/statistics/

Figure 1 shows the amount of reference sequences available over the last 11 years in the GenBank [13] and RefSeq [14] repositories from NCBI. The growth is exponential. RefSeq sequences from Archaeal and Bacterial genomes are highlighted for being a commonly used reference set for classification. In an interval of two and a half years (from June 2015 to December 2018) the RefSeq Microbial of Complete Genomes (CG) grew more than four times, with 2.5 times more species represented in the most recent set (1529 to 3850). Looking at the same data point (end of 2018), the complete RefSeq Microbial has >12 times base pairs and >5 times species compared to the CG set. These data show that databases are growing fast and the variation among them is very significant. Nowadays, such repositories are too big to be analyzed by standard hardware and if the observed growth continues, all this wealth of data will be constrained to just a few groups with available resources to process them.

The choice of the database and their subsets to perform reference-based classification is an important step and known issue in metagenomics [15]. As a crude rule of thumb, the more sequences the better. But even the complete set of sequences is not evenly distributed throughout the taxonomic tree, with different taxa being represented in different levels of quantity and quality. In addition, most of the taxa are still unknown and do not have any genomic sequence or entry in the taxonomic tree. This requires the tools to always keep up to date with the latest releases of public repositories, a task which is not trivial when dealing with very large amounts of sequences. Most of the tools lack the ability to update their own indices and databases and currently many analyses are performed with outdated resources.

Based on the RefSeq Microbial repository, a year-old release (from beginning of 2018) is 10% less

taxonomic diverse than it is today. An even older set of references obtained from June 2015 lacks 27% of today's taxonomic diversity. Further, a commonly used subset of RefSeq, the microbial complete genomes, covers only 15% of the available diversity of the full repository (December 2018). As an example, the latest release of kraken's [16] MiniKraken database (as of 18-Oct-2017) based on complete bacterial, archaeal and viral genomes, although helpful to obtain fast insights on community composition, comprises only 11% of the total taxonomic diversity available on the latest RefSeq release from 04-Jan-2019. Metagenomics analyses based on such releases are prone to underperform and miss potential species of interest. However, the use of outdated references is still common practice in some analysis based on "pre-built" indices. Most methods provide custom database building features but without incremental update capabilities. Weekly or daily updates with the most recent data are almost impossible giving the time requirements to re-build such indices with the current volume of data.

The sequence classifiers MetaPhlAn [17] and Kaiju [18] created alternatives to cover most of the diversity contained in such repositories by selecting a subset of marker genes and protein sequences, respectively. On the one hand, those methods are very powerful and provide fast and precise community profiles and read classification given their reduced index sizes. On the other hand, when analyzing whole genome sequences of complex environments, organisms with low abundance are easily missed since their genomic content may not be completely covered. In addition, current methods using complete genome sequences are struggling with the present amount of available data [11].

With those limitations in mind we developed ganon, a new reference and taxonomy-based short read classification tool for metagenomics. Ganon uses Interleaved Bloom Filters (IBF) [19] to represent very large amounts of sequences in a searchable index. This enables the indexing of large sets of sequences (e.g. complete RefSeq) in short time and with low memory consumption, consequently improving read classification for whole metagenomic sequencing experiments. Ganon also provides updatable indices which can incorporate new released sequences in short time. The classification method, which is based on k-mer counting lemma and a progressive filtering step, improves the precision of the classification without harming sensitivity when compared to state-of-the-art tools. Ganon was developed in C++ using the SeqAn library [20] and python. The code is open source and available at: https://gitlab.com/rki_bioinformatics/ganon

2 Methods

Ganon classifies reads against a set of reference sequences to find their exact or closest taxonomic origin. The method can also work in a further specialized level (e.g. assembly). An indexing step is necessary before classification, where the reference sequences will be clustered into groups based on their taxonomic classification. Ganon indices store all k-mers present in the groups of reference sequences into a specialized type of bloom filter. Once the index is created, ganon classifies the reads based on the k-mer counting lemma together with a post-filtering step providing a unique or multiple classifications for each read. Multiple classifications are solved optionally with the lowest common ancestor (LCA) algorithm [21]. In the following sections we will explain each of those steps in detail.

2.1 Indexing

Ganon indices are based on the k-mer content of the reference sequences, meaning it uses all possible substrings of length k of the given sequences. Instead of using standard methods for k-mer storage which can have high memory and space consumption when k is high (>15) we opted for bloom filters [22], a space-efficient probabilistic data structure. Since the goal of the tool is to classify sequences based on their taxonomic origin, multiple bloom filters would be necessary to represent each distinct group of sequences belonging to a certain taxonomic level (e.g. species). Such approach provides a straightforward solution, but with a high cost on the classification step by needing to compare reads multiple times against different filters. This is solved by interleaving the bloom filters, a technique previously described for the DREAM-Yara tool [19] and also part of the SeqAn library [20]. TaxSBP is used to separate the sequences into taxonomic groups and distribute them better into equal-sized clusters.

2.1.1 TaxSBP

TaxSBP [https://github.com/pirovc/taxsbp] uses the NCBI Taxonomy database [23] to generate clusters of sequences which are close together in the taxonomic tree. It does that based on an implementation of the approximation algorithm for the hierarchically structured bin packing problem [24]. As defined by Codenotti et al. this clustering method "[...] can be defined as the problem of distributing hierarchically structured objects into different repositories in a way that the access to subsets of related objects involves as few repositories as possible", where the objects are sequences assigned to taxonomic nodes of the taxonomic tree. Sequences are clustered together into groups limited by a maximum sequence length size of its components. Splitting sequences into smaller chunks with overlapping ends is supported. TaxSBP also supports one level of specialization after the leaf nodes of the tree, making it possible to further cluster sequences by strain or assembly information which is not directly contained in the NCBI Taxonomy database (Figure 2). TaxSBP also offers pre-clustering, meaning that members of a certain taxonomic level can be prevented to be split among clusters. TaxSBP can further generate bins with exclusive ranks, which are guaranteed to be unique in their cluster. The tool was developed alongside the distributed indices concept [19] and supports the update of pre-generated clusters. Since TaxSBP is based on the "pre-clustered" taxonomic tree information, the algorithm is very efficient and requires very few computational resources, being potentially useful in many other bioinformatics applications.

2.1.2 IBF

A bloom filter is a probabilistic data structure which comprises a bitvector and a set of hash functions. Each of those functions maps a key value (k-mer in our application) to one of the bit positions in the vector. Collisions in the vector are possible, meaning that distinct k-mers can be set to the same bit positions in the vector. Such overlaps can be avoided with a larger bitvector, reducing the probability of false positives.

An Interleaved Bloom Filter (IBF) is a combination of several (b) bloom filters of the same size (n) with the same hash functions into one bitvector. Each i-th bit of every bloom filter is interleaved, resulting in a final IBF of size $b * n$. Querying in such data structure is possible by retrieving the sub-bitvecors for every hash function and merging them with a logical AND operation, which will result in a final bitvector indicating the membership for the query, as depicted in Figure 2 in the DREAM-Yara manuscript by [19].

Aiming at the classification based on taxonomic levels (e.g. species, genus, ...) or assembly level (Figure 2), TaxSBP is set to cluster the input sequences into exclusive groups. Every group will contain only sequences belonging to the same taxon or assembly unit, but the same unit can be split into several groups. Groups are limited by a pre-defined threshold of the sum of the base pair length of its elements and sequences can be sliced into smaller pieces to better generate equal sized clusters.

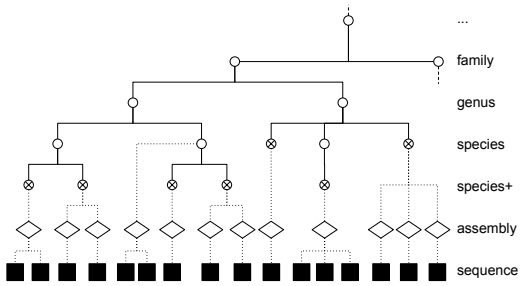

Figure 2: **Classification levels and taxonomic distribution.** Empty circles are inner nodes of the tree; marked circles are leaf nodes (also referenced in this manuscript as "taxid" nodes); full lines represent taxonomic relations, dotted lines represent the extension of the taxonomic classification to the assembly and sequence level. Species+ represents all taxonomic groups more specific than species (e.g. subspecies, species group, no rank) with species in the lineage.

Each of those clusters will correspond to a single bloom filter which is interleaved in a final IBF. Here a trade-off between the number of groups, their maximum size and the k-mer content of each group is important. The false positive rate of a bloom filter depends mainly on its bitvector size

and the number of inserted elements. In general, the more base pairs a particular cluster has, the higher the number of distinct k-mers. That requires the bloom filter to be bigger to achieve low false positive rates when querying. In ganon indices, the group with most unique k-mers will define the size and the maximum false positive rate of the final IBF since they have to be equal-sized by definition. Thus the previous clustering step is crucial to achieve a good trade-off between the number of groups, their sizes and k-mer content. The lower the taxonomic level the more fragmented the clustering will be. For example: if a reference set has 2000 species groups, there will be at least the same number of clusters. The higher the taxonomic level, the lower the number of clusters, since they can be grouped together, producing smaller filters.

The IBF has an inherent capability of updating since it is fragmented into many sub-parts. Adding new sequences to a previously generated IBF is as easy as setting the bit positions of the k-mers originated from the new sequences, once we know to which cluster it should be added to, or appending new clusters to the existing filter. To remove sequences from the IBF, we reset corresponding bit positions of the affected bins and re-insert the k-mers from the updated content.

At the end of the build process, ganon index will consist of an IBF based on a maximum classification level chosen (taxonomic rank or assembly) and auxiliary files for the classification step.

2.2 Classifying

The ganon read classification is based on the well-studied k-mer counting lemma [25, 26]. All k-mers from given reads are looked up on the indices previously generated. If a minimum number of matches between the read and the reference are achieved, a read is set to classified. Based on incremental error rates, multiple classifications for each read are filtered out and only the best ones are selected. When such filtering cannot define a single origin for a read, an optional LCA step is applied to join multiple matching reads into their lowest common ancestor node in the taxonomic tree.

2.2.1 K-mer counting lemma

The k-mer counting lemma can be defined as the minimum number of k-mer sequences from a read that should match against reference k-mers to be considered present in a set with a certain number of errors allowed. Given a read R with length l, the number of possible k-mers with length k in such read can be defined as:

$$kmers_R = l_R - k + 1 \tag{1}$$

An approximate occurrence of R in a set of references has to share at least

$$kcount_R = kmers_R - k \cdot e \tag{2}$$

k-mers, where e is the maximum number of errors/mismatches allowed.

2.2.2 Filtering

A read can be assigned to multiple references with different error rates, thus a filtering step is necessary to decrease the number of false assignments. The applied k-mer counting lemma provides k-mer counts for each read against the reference sequences. From this count it is possible to estimate the maximum number of mismatches that a read has. For example: for $k = 19$ and $length = 100$, a read with 50 19-mers matching a certain reference will have at most 2 mismatches. This calculation can be achieved by solving the Equation 2 equation for e.

Assuming that reads with fewer mismatches have a higher chance of being correct, the following filtering is applied: first, only matches against references with no mismatches are kept (all k-mers matching). If there are no such assignments, only matches with 1 error are kept. If there are none, only matches with 2 errors are kept and so on up to the maximum number of errors allowed (e in Equation 2). Similar filtration methods (also known as mapping by strata) were previously used in read mappers such as Yara. If a read is classified in several references within the same range of errors, they are all going to be reported since it is not possible to define which one has a higher chance of being correct based on the k-mer count information.

Ganon indices contain groups of reference sequences clustered by taxonomy or assembly group, depending on how the index was created. All k-mers from the reads are extracted and compared against such indices applying the k-mer counting lemma to select candidates, based on a user defined maximum number of errors. All matches within the error rate are filtered as described above and one or more matches are reported. Given our clustering approach, some groups can share the same identification target (e.g. one species was split in two or more groups due to a large amount of sequences). Those cases are treated specially by reporting only the match with more k-mer similarities since they belong to the same classification group.

Ganon also provides a way to further filter the unique classifications with a different error rate for reads that matched against just one reference group. Such filter will be applied after the standard filtration and will re-classify a low scored read to its parent taxonomic level if it scores below a certain threshold. This can be applied for filtering at low levels (e.g. assembly) since the classification in those levels should be more precise with less mismatches. This feature is also useful to avoid classifications which only happen due to a lack of related genomes (e.g. low score match on a species which is the only representative of a lineage). Additionally, ganon supports classification based on multiple indices in a user-defined hierarchy, with independent error rates for each index. At the end, an optional LCA method can be applied to solve reads with multiple matches with a more conservative and less precise taxonomic classification, reporting one match for each read.

3 Results

We evaluate ganon against a well-established method for read classification: kraken [16] and also against a version called krakenuniq [27] which uses the basic kraken algorithm and also allows classification on more specific levels after taxonomic assignments (e.g. up to assembly or sequence level). We further compare the results against Centrifuge [28] which uses the Burrows-Wheeler transform (BWT) and the Full-text index in Minute space (FM-)index for indexing and aims at reducing the index size by compressing similar sequences together. Here we consider only the direct read classification capabilities of the tools. Further functionalities like the estimation of a presence of a certain organism or abundance estimation are not covered.

Ganon and the other evaluated tools are reference-based, meaning all classifications are made based on previously generated sequence data. The choice of the underlying database is therefore crucial. We use the same sequences and taxonomic database version for all tools when creating their indices to guarantee a fair comparison. The NCBI RefSeq repository was the chosen source of the reference sequences since it provides a curated and very extensive set of sequences. Two subsets of RefSeq were extracted: a set of only complete genomes from the groups Archaea, Bacteria, Fungi and Viral (RefSeq CG) and complete set of all genomes from the same groups (RefSeq ALL) both dating from 19-December-2018 (Table 1). Taxonomic information was obtained for the same dates as the sequences.

	Base pairs	# assemblies	# sequences
RefSeq CG	46.986.899.184	19.623	33.029
RefSeq ALL	587.607.072.429	147.713	15.201.684

Table 1: **Reference sequences used for evaluations.** Data was downloaded using https://github.com/pirovc/genome_updater

Both reference sets have some taxonomic groups over-represented with several assemblies for a single species. For example: the *Escherichia coli* species group is being represented by 634 assemblies, accounting for almost 7% of all base pairs in the RefSeq CG. This is even more pronounced on RefSeq ALL, with 13,259 *E. Coli* assemblies representing more than 11% of the base pairs in the whole set. In RefSeq CG, the 92 most over-represented species have as many base pairs as the remaining 11,372 species. In RefSeq ALL this ratio is 14 to 29,047 (Supplementary Figures 1-4). This unbalanced distribution of references may not only bias analysis but also introduces redundancy to the set when aiming classification at taxonomic levels. For such reason, when not classifying at assembly level, we removed over-represented assemblies from our reference set, keeping only the 3 biggest assemblies for each taxonomic group (Table 2).

For the classification we use data produced for the first CAMI Challenge [6]. Sets of simulated

	Base pairs	# species	# leaf taxids	# assemblies	# sequences
RefSeq CG top 3	$2,9e10$ (62%)	11.464 (100%)	14.071 (100%)	15.171 (77%)	24.290 (74%)
RefSeq ALL top 3	$2,1e11$ (36%)	29.061 (100%)	51.292 (100%)	56.805 (38%)	4.400.402 (29%)

Table 2: **Reference sequences after over-representation filtering.** Percentages in brackets show amount of data left compared to the original set (Table 1)

and real datasets mimicking commonly used environments and settings were obtained, representing multiple closely related strains, plasmid and viral sequences. Those samples are divided in 3 categories: low, medium and high complexity with increasing number of organisms and different sequencing profiles providing a well-produced and challenging dataset to analyze. The simulated reads were generated based on public data (NCBI, SILVA46) and an exact ground truth is provided with an origin for each read down to sequence level. The real dataset was obtained from newly sequenced genomes of 700 microbial isolates and 600 circular elements and a ground truth is provided at a taxonomic level. Here we use one high complexity sample from both categories to perform evaluations (Supplementary Table 4).

The classification evaluation was performed in a binary fashion. Every read has an assembly or taxonomic identification defined by the ground truth. If a certain tool gives the correct classification for a read at its defined level or at a correct higher level, this read is marked as a true positive. For example: if a read has an species group assigned in the ground truth but the tool outputs it at its correct genus, the read is a true positive for the genus level. False positives are reads with a wrong direct classification or a false high level classification. That way a read classified in the wrong taxonomic group will be considered a false positive, even if it is correctly placed on a higher hierarchy rank (e.g. wrong species of the same genus). Reads with too-specific classifications are also counted as false positives (e.g. tool classifies a read at species but the ground truth only defines it at family level).

3.1 Indexing

The set of sequences from RefSeq CG and RefSeq ALL complete and top 3 (Table 1 and 2) were used as input to generate the indices for each evaluated tool. Here evaluation is done by total run-time, memory consumption and final index size (Table 3).

Max. depth	Method	RefSeq CG top 3			RefSeq ALL top 3		
classification		time	Memory (GiB)	Index size (GiB)	time	Memory (GiB)	Index size (GiB)
taxid	centrifuge	06:51:25	269	12	-	-	-
	ganon	00:07:14	76	72	00:46:48	288	285
	kraken	04:53:31	200	184	-	-	-
		RefSeq CG			RefSeq ALL		
		time	Memory (GiB)	Index size (GiB)	time	Memory (GiB)	Index size (GiB)
assembly	centrifuge	12:32:08	1441	20	-	-	-
	ganon	00:10:13	102	96	03:37:12	752	748
	krakenuniq	09:36:45	329	191	11 days*	2500*	-

Table 3: **Build times, memory consumption and index sizes.** Tools taking more than 24 hours to build were not considered. 48 threads were used for all tools. Computer specifications: 56 x Intel Xeon Processor (Skylake, IBRS), 2593 MHz with two terabytes of memory. Parameters used are in the supplementary material. * Approximate time and memory consumption for kraken with a similar reference set obtained from [11]

We built two different indices in different depths of classification: at assembly level using the complete set of references and at taxonomic leaf nodes (taxid) using only the top 3 assemblies for each taxonomic group. Centrifuge indices allow classification up to a taxonomic or sequence level (assembly must be deduced from sequences) and krakenuniq indices allow classifications up to assembly and sequence level. Kraken can do taxonomic level classification. krakenuniq was not evaluated on taxid level since it runs exactly the same base algorithm as kraken in this configuration.

When indexing the RefSeq CG (Table 3), the evaluated tools took between 7 minutes and 12

hours, with ganon being the fastest and centrifuge the slowest. Ganon shows a significant overall reduction in memory consumption and run-time compared to the other tools: 56 times faster than krakenuniq, the second fastest at assembly level. Centrifuge achieves the lowest index size with the cost of having the highest memory and time consumption in every scenario evaluated.

Ganon indexed the RefSeq ALL (top 3) in 46 minutes and the complete RefSeq ALL in 3 hours and 37 minutes. We could not run centrifuge, kraken and krakenuniq for RefSeq ALL on our infrastructure, given computational limitations or long execution time (limited on 24 hours). A recent publication [11] reported that kraken and consequently krakenuniq need 11 days to build a database for the bacterial RefSeq version 80, which is smaller but an approximate of the RefSeq ALL here evaluated, with 64 cores of E7-8860v4 CPUs and three terabytes of memory.

3.2 Updating

Ganon is the only tool among the evaluated ones which allows updates on previously generated indices. We evaluated this functionality with Bacterial sequences added to RefSeq CG dating from 19-December-2018 to 21-January-2019 (comprising 2.77 Gbp, 1307 sequences, 370 species from which 213 are new to the reference set and 716 new assemblies). Updating the ganon index based on RefSeq CG with this dataset took under 5 minutes, less than half of the time necessary to create the index (Table 3).

3.3 Classifying

The classification results were evaluated in terms of sensitivity and precision at each taxonomic level based on the binary classification previously described. The values are always cumulative up to the evaluated level, for example: sensitivity and precision at the genus level will account for all classifications up to genus level (species and species+). Every taxonomic classification in between and different than the predefined ranks (superkingdom, phylum, class, order, family, genus and species) are counted towards the closest high level parent rank in its lineage. Species+ represents all taxonomic groups more specific than species (e.g. subspecies, species group, no rank) with species in the lineage (Figure 2).

The results for CAMI simulated and real datasets should be interpreted considering each tool maximum depth of classification: centrifuge, ganon and kraken at the taxonomic leaf nodes (taxid). centrifuge, ganon and krakenuniq up to assembly level. Given the availability of the ground truth, only simulated data was evaluated up to assembly level while real data was evaluated up to species+ level. Centrifuge outputs at sequence level, thus an extra step of applying a LCA algorithm for non-unique matches was necessary to generate results at assembly and taxid levels. Ganon classifies at a specific rank which the database was built for. kraken outputs at taxid level and krakenuniq at assembly level.

Figure 3 compares the results for one simulated high complexity dataset using the indices based on RefSeq CG top 3 and ganon for RefSeq ALL top 3. Using the same set of references, all tools performed similarly in terms of sensitivity (Table 4), with ganon achieving higher F1-Score from species to superkingdom. In terms of precision, ganon shows superior performance (>5%) in every scenario when compared to its competitors. Using the indices based on the RefSeq ALL, ganon show big improvements compared to the analysis based on the RefSeq CG, given the broader diversity covered by this set. Sensitivity went up by more than 20% at genus. Precision is highly improved with more references available, with ganon reaching >98% at species+. For higher ranks (species and species+) results were mainly limited by the availability of references sequences (black lines on Figure 3 - Sensitivity), with higher values when more references were present (genus and below).

The same analysis was performed on real data, provided as the challenge data on CAMI. This set is more challenging since most of the species in the sample are novel and not present in the public repositories of reference sequences. As stated by the CAMI results [6], most tools performed poorly in this dataset in terms of sensitivity, as show in Figure 4 and Table 5. The results follow the same trend from the previous analysis, with the tools being similarly sensitive while ganon being more precise with less false positives in all scenarios. Here the impact of a larger set of references (RefSeq ALL) is more evident, significantly improving the results in both sensitivity and precision. Genus level sensitivity went from around 5% to almost 50% with a significant improvement in precision. The boost in precision with ganon in this scenario can be explained by the longer read

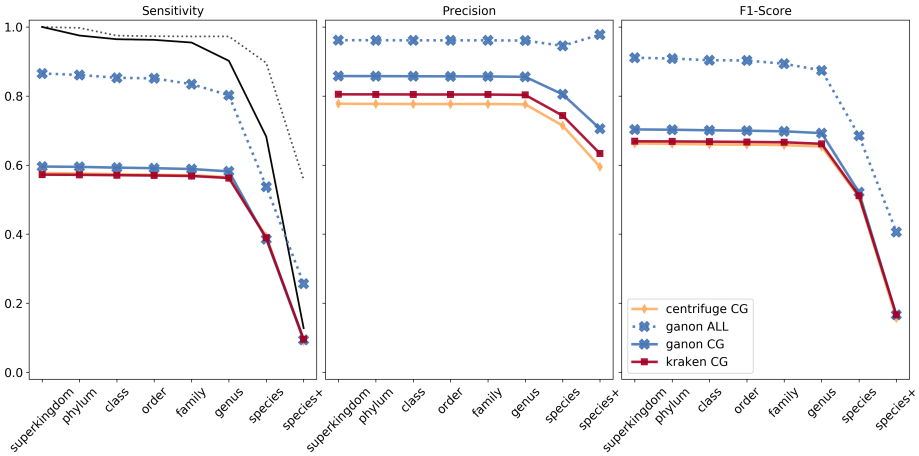

Figure 3: **Precision, Sensitivity and F1-Score for the simulated reads.** Data for the simulated high complexity sample number 1. RefSeq CG top 3 results are represented with continuous lines and RefSeq ALL top 3 with dotted lines. Black lines show the maximum sensitivity possible given the references available in the set.

Reference set	Method	Precision			Sensitivity			F1-Score		
		genus	species	species+	genus	species	species+	genus	species	species+
RefSeq CG top 3	centrifuge	77.60%	71.44%	59.47%	56.49%	**39.34%**	9.04%	65.38%	50.74%	15.70%
	ganon	**85.56%**	**80.49%**	**70.54%**	**58.17%**	38.57%	9.42%	**69.26%**	**52.15%**	16.61%
	kraken	80.31%	74.33%	63.33%	56.26%	38.95%	**9.59%**	66.17%	51.12%	**16.66%**
RefSeq ALL top 3	ganon	96.05%	94.53%	97.77%	80.22%	53.76%	25.65%	87.43%	68.54%	40.64%

Table 4: **Precision, Sensitivity and F1-Score values for the simulated reads.** Data for the simulated high complexity sample number 1. Ganon shows higher precision in all levels with little variation on sensitivity. The use of a bigger reference set with RefSeq ALL significantly improves results. Detailed parameters and results for all ranks are in the supplementary material.

lengths (150 bp for real data, against 100 bp for simulated) where more mismatches are allowed, showing how the progressive filter can make a good distinction between false and true positives.

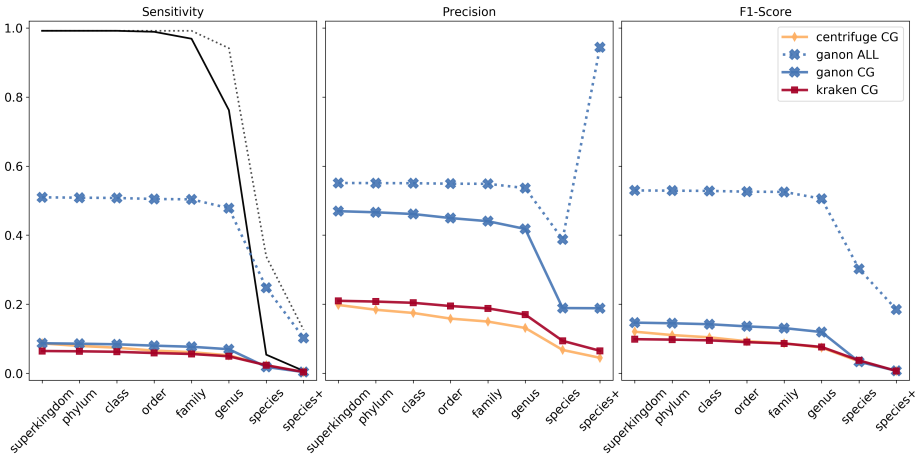

Figure 4: **Precision, Sensitivity and F1-Score for the real reads.** Data for the real high complexity sample number 1. RefSeq CG top 3 results are represented with continuous lines and RefSeq ALL top 3 with dotted lines. Black lines show the maximum sensitivity possible given the references available in the set.

When comparing ganon at assembly level classification against centrifuge and krakenuniq, the

Reference set	Method	Precision			Sensitivity			F1-Score		
		genus	species	species+	genus	species	species+	genus	species	species+
RefSeq CG top 3	centrifuge	13.12%	6.76%	4.49%	5.19%	2.32%	0.37%	7.44%	3.45%	0.68%
	ganon	**41.79%**	**18.89%**	**18.84%**	**6.97%**	1.83%	0.36%	**11.95%**	3.35%	**0.71%**
	kraken	17.01%	9.40%	6.52%	4.92%	**2.34%**	**0.37%**	7.63%	**3.74%**	0.70%
RefSeq ALL top 3	ganon	53.63%	38.81%	94.41%	47.80%	24.80%	10.24%	50.55%	30.26%	18.47%

Table 5: **Precision, Sensitivity and F1-Score values for the real reads.** Data for the real high complexity sample number 1. Ganon shows higher precision in all levels with little variation on sensitivity. The use of a bigger reference set with RefSeq ALL significantly improves results. Detailed parameters and results for all ranks are in the supplementary material.

same trend is present (Table 6). Ganon has approximately 20% less false positives assignments (3.8 million reads out of 75 million reads) in comparison with krakenuniq, the second best, while having a very similar number of true positives.

Method	Precision	Sensitivity	F1-Score
centrifuge	30.77%	**11.82%**	17.08%
ganon	**37.25%**	11.52%	**17.60%**
krakenuniq	32.45%	11.67%	17.17%

Table 6: **Precision, Sensitivity and F1-Score values for the simulated reads against RefSeq CG at assembly level** Data for the simulated high complexity sample number 1. Ganon shows higher precision with little variation on sensitivity. The maximum sensitivity that could be achieved given the reference set is 26%. Detailed parameters are in the supplementary material.

In all scenarios evaluated, ganon consistently provides a higher precision within the same reference set (from 1-30% higher) while keeping sensitivity values high, with little variation (around 1-2%). High precision translates to fewer reads with a wrong classification. In the simulated set against RefSeq ALL top 3 (Table 4) ganon has only 3% of false assignments out of the classified reads. A lower number of false positives are crucial to reduce bias in further analysis (e.g. abundance estimation).

Table 7 compares the performance of the analyzed tools in terms of how many base pairs they can classify per minute (Mbp/m). Kraken is the tool with the best performance on classification base on Mbp/m. Ganon can be executed with different $offset$ values, skipping a certain number of k-mers, speeding up classification. $offset = 1$ means that all k-mers are being evaluated while $offset = 2$ means that every 2nd k-mer is being skipped. The trade-off between offset and precision/sensitivity for ganon results can be seen in Supplementary Figures 5 and 6. Speed variation between simulated and real reads is partly explained due to their classification rate: on average 70% of the simulated reads are classified while only 20% of the real reads are classified. Memory consumption is mainly based on the index size of each tool (Table 3), with little variation besides that. Centrifuge is the method with lowest memory requirements to classify (14 GB) followed by ganon (75 GB) and kraken (190 GB).

Method	simulated	real
centrifuge	274	791
ganon	461	649
kraken	1236	1859

Table 7: **Classification performance**. 74 million reads from simulated and real sets (both high complexity sample number 1) in Mbp/m with 48 threads against RefSeq CG top 3. Centrifuge performance estimation based on "Time searching" report value without considering post-processing time. Ganon performance based on classification time without optional LCA computation. Times are the average of 5 consecutive runs. Computer specifications and parameters used are in the Supplementary material.

3.4 Tara Oceans and multi-hierarchy analysis

Comparisons with similar methods were performed with one index including all organism groups of interest. However, ganon provides a multi-hierarchical classification system that can use several filters interchangeably, giving more flexibility to perform analysis. This feature together with the fast build and update times allow the creation of independent indices that can be gradually updated and used in any combination.

As a use case we analyzed one sample from TARA Oceans project [29] whose organisms are known for being underrepresented in the current repositories, even with recent advances in cataloging such environment. Besides Archaea (A), Bacteria (B), Fungi (F) and Virus (V) reference sets used from the previous evaluation, we also added Plasmids (PL) and Protozoa (P) reference sets, all from RefSeq. In addition, we built an index based on a set of recently assembled contigs from Tara Oceans project (TARA) [10]. The reads from sample ERR599159 were trimmed by quality and size and one million random reads were analyzed.

We first classified the sample against the "standard" RefSeq dataset used in the previous section, achieving less than 3% of classification on the RefSeq CG set and less than 6% in RefSeq ALL (Figure 5). As expected, a higher number of reads was achieved classifying them against the custom TARA reference set, with slightly more than 12% of reads classified.

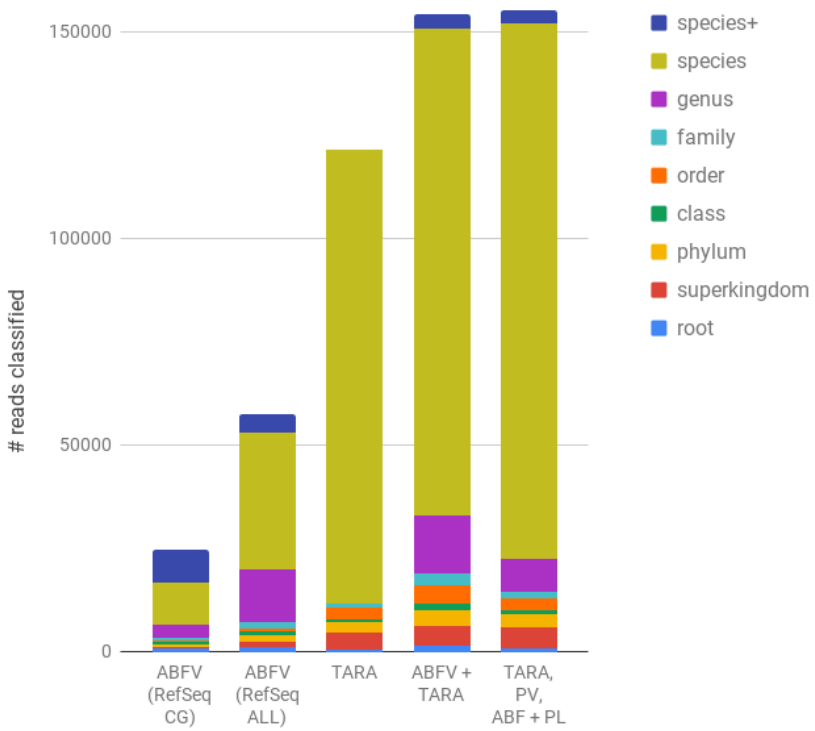

Figure 5: Tara Oceans sample analysis by taxonomic level with different combination of indices

Since the TARA set also contains Virus and Microbes, it is hard to interpret such results independently from each other. Here we could take advantage of ganon multi-filter analysis. We ran ganon classify combining ABFV (as independent indices) and TARA reference sets in one run. Such combination classified more than 15.4% of the reads. In addition we performed a multi-filter and multi-hierarchy run with ganon in the following order: first reads were classified against TARA set, second reads with no matches were classified against Protozoa and Virus. Lastly, all reads without matches were classified side-to-side against Bacteria, Archaea, Fungi and the Plasmid set. This combination achieved the highest classification rate 15.5%, mainly improving the ratio of sequences assigned to species level as show in Figure 5.

Although such analyses are possible with similar approaches, ganon's fast indexing and built-in

multi-hierarchical classification allow exploratory data analysis in a very fast and convenient way. Combination of several indices in any order for classification is possible with a single command. The number of errors can be set for each index as well as individual output files, allowing diverse applications (e.g. host removal before classification). In addition. given the design of Interleaved Bloom Filters, it is possible to achieve better index to size ratio when building organism-specific indices, by clustering and adapting the size of bins and filter accordingly, speeding up the classification step.

4 Discussion

We presented ganon, a novel method to index big sets of genomic sequences and classify short reads for environmental samples based on a taxonomic oriented scheme. Ganon's strength lies in providing an ultra-fast indexing method for very large sets of reference sequences and a high precision in classification. That is possible due to a novel application of Interleaved Bloom Filters with a k-mer counting and filtering scheme. This is relevant in a time where the number of available sequences keeps increasing daily and current tools struggle to build and update such indices.

By indexing huge sets of reference sequences and turn them into searchable indices, ganon allows scientists to make most of their data. Short turnaround times for index building and updating is crucial for many bioinformatics applications (e.g. outbreak investigation), quickly going from raw data to taxonomic assignments. Ganon facilitates database maintenance, allowing short increment on a daily basis being the only realistic option up-to-date to keep-up with the fast pace of data generation. In addition, ganon indices are flexible and can be built for high taxonomic levels (e.g. genus), requiring less space and memory, consequently improving classification speed. A trade-off between filter size, clustering and false positive rate is also possible, sacrificing precision over performance or disk usage over classification speed.

Classification results presented here are on par with state-of-the-art methods with regard to sensitivity, while improving precision rates in every scenario of our evaluations, sometimes manifold. We attribute such improvement to an application of the k-mer counting lemma together with a progressive filtering step, which can better separate false from true positives. The unique filtering step also allows better selection of false positives when taxonomic groups are underrepresented in the reference set.

Even with ganon achieving improved results, in general terms, short read classification tools tested here perform similarly when based on the same underlying set of reference sequences. In addition, the more complete the reference set the better the classifications. The difference in sensitivity when using RefSeq ALL compared to only complete genomes is very significant and tends to get even bigger with more sequences added to this repository. Thus the choice of the database is crucial and should not be overlooked when analyzing metagenomic data. Even though centrifuge, kraken and krakenuniq could potentially perform well with more reference sequences, the time to index such set is highly prohibitive. Ganon manages to index a big set of references sequences and keep them updated in very short time with classifications results being as good as or better than the evaluated tools. Additionally, centrifuge does not provide a direct one-match-per-read result and an in-house post-processing step was necessary to achieve it. To the best of our knowledge, ganon is the only tool with update capabilities, which is performed in a fraction of the complete build time. That poses an advantage to keep up to date with the public repositories and even each single of their frequent updates.

Instead of reporting reads only at a fixed LCA level, ganon provides every output for a read at the level the index was built, also allowing assembly level classification. This is crucial for strain level analysis, where candidate organisms are more insightful for further investigations than a conservative identification. Multi-filter and multi-hierarchy support on classification tied to fast indexing of reference sets makes ganon a powerful tool for exploratory data analysis, enabling multiple combinations of indices and errors rates in an iterative manner.

Ganon's fast indexing performance is mainly due to the fact that k-mers are not being counted, instead all of them are being inserted into a space-efficient data structure (IBF) which also provides quick look-up times. However, data growth is not slowing down and in the long term such approach will reach a limit. For such reason, a k-mer aware clustering which could reduce redundancy would be beneficial together with a minimizer implementation which could reduce the index sizes. Such features are already planned and under development for future releases of the tool.

To conclude, we believe that ganon can be a useful tool for metagenomics analysis in a time where reference sequence repositories are growing fast. Ganon provides fast indexing, updatability, competitive results and improved precision for short read classification.

5 Acknowledgments

We would like to thank Tobias Loka for helpful discussions and Diogo Andrei Benvenutti for git project improvements and C++ contributions.

6 Author contributions

V.C.P. designed, implemented and tested the tool and the experiments and wrote the manuscript. T.H.D. created and implemented the IBF and reviewed the manuscript. E.S. improved and tested the IBF and reviewed the manuscript. K.R. and B.Y.R. designed and supervised the project, discussed the experiments and reviewed the manuscript. All authors read and approved the final manuscript.

7 Funding

This work was supported by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES) - Ciencia sem Fronteiras (BEX 13472/13-5 to VCP), by the BMBF (InfectControl 2020) and by the BMBF-funded de.NBI Cloud within the German Network for Bioinformatics Infrastructure (de.NBI) (031A537B, 031A533A, 031A538A, 031A533B, 031A535A, 031A537C, 031A534A, 031A532B).

References

[1] Simon H Tausch, Benjamin Strauch, Andreas Andrusch, Tobias P Loka, Martin S Lindner, Andreas Nitsche, and Bernhard Y Renard. LiveKraken—real-time metagenomic classification of illumina data. *Bioinformatics*, June 2018.

[2] Anastasis Oulas, Christina Pavloudi, Paraskevi Polymenakou, Georgios A. Pavlopoulos, Nikolas Papanikolaou, Georgios Kotoulas, Christos Arvanitidis, and loannis Iliopoulos. Metagenomics: Tools and Insights for Analyzing Next-Generation Sequencing Data Derived from Biodiversity Studies. *Bioinformatics and Biology Insights*, 9:BBI.S12462, January 2015.

[3] Alexa B. R. McIntyre, Rachid Ounit, Ebrahim Afshinnekoo, Robert J. Prill, Elizabeth Hénaff, Noah Alexander, Samuel S. Minot, David Danko, Jonathan Foox, Sofia Ahsanuddin, Scott Tighe, Nur A. Hasan, Poorani Subramanian, Kelly Moffat, Shawn Levy, Stefano Lonardi, Nick Greenfield, Rita R. Colwell, Gail L. Rosen, and Christopher E. Mason. Comprehensive benchmarking and ensemble approaches for metagenomic classifiers. *Genome Biology*, 18(1):182, dec 2017.

[4] Stinus Lindgreen, Karen L. Adair, and Paul P. Gardner. An evaluation of the accuracy and speed of metagenome analysis tools. *Scientific Reports*, 6:19233, jan 2016.

[5] Michael A. Peabody, Thea Van Rossum, Raymond Lo, and Fiona S. L. Brinkman. Evaluation of shotgun metagenomics sequence classification methods using in silico and in vitro simulated communities. *BMC Bioinformatics*, 16(1):362, dec 2015.

[6] Alexander Sczyrba, Peter Hofmann, Peter Belmann, David Koslicki, Stefan Janssen, Johannes Dröge, Ivan Gregor, Stephan Majda, Jessika Fiedler, Eik Dahms, Andreas Bremges, Adrian Fritz, Ruben Garrido-Oter, Tue Sparholt Jørgensen, Nicole Shapiro, Philip D Blood, Alexey Gurevich, Yang Bai, Dmitrij Turaev, Matthew Z DeMaere, Rayan Chikhi, Niranjan Nagarajan, Christopher Quince, Fernando Meyer, Monika Balvočiūtė, Lars Hestbjerg Hansen, Søren J Sørensen, Burton K H Chia, Bertrand Denis, Jeff L Froula, Zhong Wang, Robert Egan, Dongwan Don Kang, Jeffrey J Cook, Charles Deltel, Michael Beckstette, Claire Lemaitre, Pierre Peterlongo, Guillaume Rizk, Dominique Lavenier, Yu-Wei Wu, Steven W Singer, Chirag Jain,

Marc Strous, Heiner Klingenberg, Peter Meinicke, Michael D Barton, Thomas Lingner, Hsin-Hung Lin, Yu-Chieh Liao, Genivaldo Gueiros Z. Silva, Daniel A Cuevas, Robert A Edwards, Surya Saha, Vitor C Piro, Bernhard Y Renard, Mihai Pop, Hans-Peter Klenk, Markus Göker, Nikos C Kyrpides, Tanja Woyke, Julia A Vorholt, Paul Schulze-Lefert, Edward M Rubin, Aaron E Darling, Thomas Rattei, and Alice C McHardy. Critical Assessment of Metagenome Interpretation—a benchmark of metagenomics software. *Nature Methods*, 14(11):1063–1071, oct 2017.

[7] Monika Balvočiūtė and Daniel H. Huson. SILVA, RDP, Greengenes, NCBI and OTT — how do these taxonomies compare? *BMC Genomics*, 18(S2):114, mar 2017.

[8] Donovan H. Parks, Christian Rinke, Maria Chuvochina, Pierre-Alain Chaumeil, Ben J. Woodcroft, Paul N. Evans, Philip Hugenholtz, and Gene W. Tyson. Recovery of nearly 8,000 metagenome-assembled genomes substantially expands the tree of life. *Nature Microbiology*, 2(11):1533–1542, nov 2017.

[9] Supratim Mukherjee, Rekha Seshadri, Neha J Varghese, Emiley A Eloe-Fadrosh, Jan P Meier-Kolthoff, Markus Göker, R Cameron Coates, Michalis Hadjithomas, Georgios A Pavlopoulos, David Paez-Espino, Yasuo Yoshikuni, Axel Visel, William B Whitman, George M Garrity, Jonathan A Eisen, Philip Hugenholtz, Amrita Pati, Natalia N Ivanova, Tanja Woyke, Hans-Peter Klenk, and Nikos C Kyrpides. 1,003 reference genomes of bacterial and archaeal isolates expand coverage of the tree of life. *Nature Biotechnology*, 35(7):676–683, jun 2017.

[10] Benjamin J. Tully, Elaina D. Graham, and John F. Heidelberg. The reconstruction of 2, 631 draft metagenome-assembled genomes from the global oceans. *Scientific Data*, 5:170203, jan 2018.

[11] Daniel J. Nasko, Sergey Koren, Adam M. Phillippy, and Todd J. Treangen. RefSeq database growth influences the accuracy of k-mer-based lowest common ancestor species identification. *Genome Biology*, 19(1), December 2018.

[12] Xin Li, Saleh A Naser, Annette Khaled, Haiyan Hu, and Xiaoman Li. When old metagenomic data meet newly sequenced genomes, a case study. *PLOS ONE*, 13(6):e0198773, jun 2018.

[13] Dennis A. Benson, Mark Cavanaugh, Karen Clark, Ilene Karsch-Mizrachi, James Ostell, Kim D. Pruitt, and Eric W. Sayers. GenBank. *Nucleic Acids Research*, 46(D1):D41–D47, January 2018.

[14] Daniel H. Haft, Michael DiCuccio, Azat Badretdin, Vyacheslav Brover, Vyacheslav Chetvernin, Kathleen O'Neill, Wenjun Li, Farideh Chitsaz, Myra K. Derbyshire, Noreen R. Gonzales, Marc Gwadz, Fu Lu, Gabriele H. Marchler, James S. Song, Narmada Thanki, Roxanne A. Yamashita, Chanjuan Zheng, Françoise Thibaud-Nissen, Lewis Y. Geer, Aron Marchler-Bauer, and Kim D. Pruitt. RefSeq: an update on prokaryotic genome annotation and curation. *Nucleic Acids Research*, 46(D1):D851–D860, jan 2018.

[15] Florian P. Breitwieser, Jennifer Lu, and Steven L. Salzberg. A review of methods and databases for metagenomic classification and assembly. *Briefings in Bioinformatics*, September 2017.

[16] Derrick E Wood and Steven L Salzberg. Kraken: ultrafast metagenomic sequence classification using exact alignments. *Genome Biology*, 15(3):R46, mar 2014.

[17] Duy Tin Truong, Eric A. Franzosa, Timothy L. Tickle, Matthias Scholz, George Weingart, Edoardo Pasolli, Adrian Tett, Curtis Huttenhower, and Nicola Segata. MetaPhlAn2 for enhanced metagenomic taxonomic profiling. *Nature Methods*, 12(10):902–903, oct 2015.

[18] Peter Menzel, Kim Lee Ng, and Anders Krogh. Fast and sensitive taxonomic classification for metagenomics with Kaiju. *Nature Communications*, 7:11257, apr 2016.

[19] Temesgen Hailemariam Dadi, Enrico Siragusa, Vitor C Piro, Andreas Andrusch, Enrico Seiler, Bernhard Y Renard, and Knut Reinert. DREAM-Yara: an exact read mapper for very large databases with short update time. *Bioinformatics*, 34(17):i766–i772, September 2018.

[20] Knut Reinert, Temesgen Hailemariam Dadi, Marcel Ehrhardt, Hannes Hauswedell, Svenja Mehringer, René Rahn, Jongkyu Kim, Christopher Pockrandt, Jörg Winkler, Enrico Siragusa, Gianvito Urgese, and David Weese. The SeqAn C++ template library for efficient sequence analysis: A resource for programmers. *Journal of Biotechnology*, 261(July):157–168, nov 2017.

[21] Daniel H Huson, Alexander F Auch, Ji Qi, and Stephan C Schuster. MEGAN analysis of metagenomic data. *Genome research*, 17(3):377–86, mar 2007.

[22] Burton H. Bloom. Space/time trade-offs in hash coding with allowable errors. *Communications of the ACM*, 13(7):422–426, 1970.

[23] Scott Federhen. The NCBI Taxonomy database. *Nucleic Acids Research*, 40(D1):D136–D143, jan 2012.

[24] Bruno Codenotti, Gianluca De Marco, Mauro Leoncini, Manuela Montangero, and Massimo Santini. Approximation algorithms for a hierarchically structured bin packing problem. *Information Processing Letters*, 89(5):215–221, mar 2004.

[25] Petteri Jokinen and Esko Ukkonen. Two algorithms for approximate string matching in static texts. *Mathematical Foundations of Computer Science 1991*, pages 240–248, 1991.

[26] Knut Reinert, Ben Langmead, David Weese, and Dirk J Evers. Alignment of Next-Generation Sequencing Reads. *Annual Review of Genomics and Human Genetics*, 16(1):133–151, aug 2015.

[27] F. P. Breitwieser, D. N. Baker, and S. L. Salzberg. KrakenUniq: confident and fast metagenomics classification using unique k-mer counts. *Genome Biology*, 19(1):198, December 2018.

[28] Daehwan Kim, Li Song, Florian P Breitwieser, and Steven L Salzberg. Centrifuge: rapid and sensitive classification of metagenomic sequences. *Genome Research*, 26(12):1721–1729, dec 2016.

[29] S. Sunagawa, L. P. Coelho, S. Chaffron, J. R. Kultima, K. Labadie, G. Salazar, B. Djahanschiri, G. Zeller, D. R. Mende, A. Alberti, F. M. Cornejo-Castillo, P. I. Costea, C. Cruaud, F. d'Ovidio, S. Engelen, I. Ferrera, J. M. Gasol, L. Guidi, F. Hildebrand, F. Kokoszka, C. Lepoivre, G. Lima-Mendez, J. Poulain, B. T. Poulos, M. Royo-Llonch, H. Sarmento, S. Vieira-Silva, C. Dimier, M. Picheral, S. Searson, S. Kandels-Lewis, C. Bowler, C. de Vargas, G. Gorsky, N. Grimsley, P. Hingamp, D. Iudicone, O. Jaillon, F. Not, H. Ogata, S. Pesant, S. Speich, L. Stemmann, M. B. Sullivan, J. Weissenbach, P. Wincker, E. Karsenti, J. Raes, S. G. Acinas, P. Bork, E. Boss, C. Bowler, M. Follows, L. Karp-Boss, U. Krzic, E. G. Reynaud, C. Sardet, M. Sieracki, and D. Velayoudon and. Structure and function of the global ocean microbiome. *Science*, 348(6237):1261359–1261359, may 2015.