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

Vitor C. Piro (1,2) Temesgen Hailemariam Dadi (3) Enrico Seiler (3) Knut Reinert (3) 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 metagenomic studies, providing a broader catalog of reference organisms in a variety of studies. 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 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 As a result, many tools still rely on static outdated datasets and clearly underperform under the currently available wealth of genome sequences.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 short 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 under 25 minutes to index the complete (i.e. high quality) archaeal-bacterial genomes and under 3 hours for all archaeal-bacterial genomes (30GB and 380GB of raw data, respectively). Ganon 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. Using data from the CAMI challenge, ganon shows strongly improved precision (in some cases more than double) while having equal or better sensitivity compared to state-of-the-art tools. Ganon's memory footprint is less than half of similar tools using the same reference set. Using a larger set of references with a broader diversity, ganon classified 25% more reads achieving 97% precision at species 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

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 [Tausch, S. H.] or after sequencing, is usually the first step prior to abundance estimation, profiling and assembly. Many tools have been specifically developed for this task over the last years with different strategies [Pavlopoulos, Oulas, A.; McIntyre, A. B. R.; Lindgreen, S.; Peabody, M. A.; Sczyrba, A.] to achieve good performance classifying huge amount of short reads against a predefined and static set of reference sequences. Many of those use taxonomy based classifications [Balvočiūtė, M.], providing a backbone arrangement of already obtained sequences which can be helpful for researchers to understand the composition of samples better.

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 [Parks, D. H., Mukherjee, S.], 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 [Nasko, D. J.]. The problem that was at first mainly focused on the speed of short read classification is now shifting towards managing the huge reference sizes and their frequent updates [Li, X.].

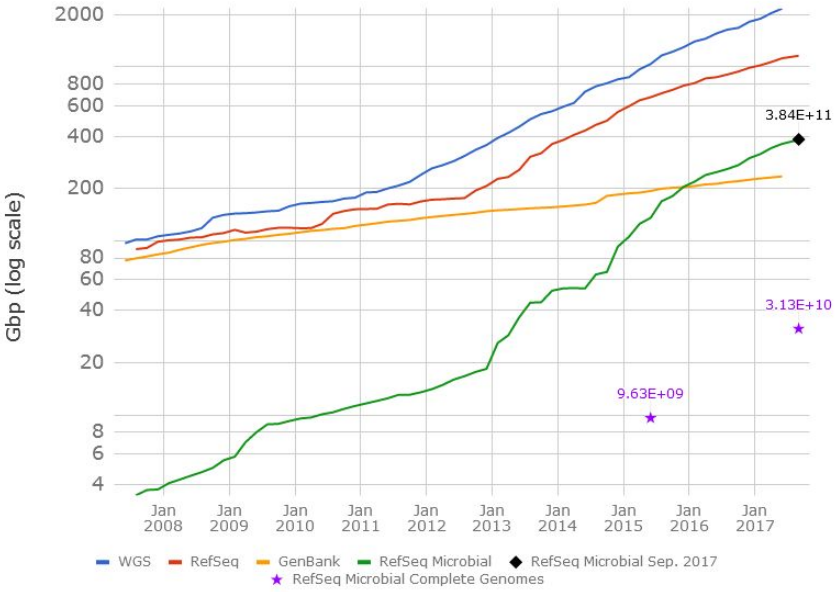

Figure 1: Number of available sequences in NCBI repositories from 2007 to 2017. Microbial stands for Archaeal and

Bacterial organisms. 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 for the last 10 years in the GenBank [Benson, D. A.] and RefSeq [Haft, D. H.] repositories from NCBI. The growth is exponential. RefSeq sequences from Archaeal and Bacterial genomes are highlighted for being a commonly used reference set for metagenomic classification. In an interval of 2 years (2015-2017) the RefSeq Microbial of Complete Genomes grew more than three times, with almost two times more species contained in the most recent set. Looking at the same data point (end of 2017), the complete RefSeq Microbial had >12 times sequences and >5 times species compared to the Complete Genomes 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 analysed 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.

Therefore, the choice of the database and their subsets to perform reference-based classification is an important step and known issue in metagenomics [Breitwieser, F. P. 2017]. 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 it is current that many analyses are performed in outdated resources.

The sequence classifiers MetaPhlAn [Truong, D. T.] and Kaiju [Menzel, P.] 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 metagenomic sequencing in complex environments, organisms with low abundance are easily missed, since their genomic content may not be completely covered. In addition, current methods which work with complete genome sequences are struggling with the current amount of available data [Nasko, D. J.].

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 [Dadi, T. H.] to represent very large amounts of sequences into 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 is open source and available at: https://gitlab.com/rki_bioinformatics/ganon

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 with the lowest common ancestor algorithm [Huson, D. H.]. In the following sections we will explain each of those steps in detail.

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 [Bloom, B. H.], 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 (IBF), a technique previously described for the DREAM-Yara tool [Dadi, T. H.] and also part of the SeqAn library [Reinert, K. 2017]. TaxSBP is used to separate the sequences into taxonomic groups and distribute them better into equal-sized clusters.

TaxSBP

TaxSBP [https://github.com/pirovc/taxsbp] uses the NCBI Taxonomy database [Federhen, S.] 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 [Codenotti, B.]. 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. 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 supports 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 [Dadi, T. H.] 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.

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 Dadi et. al.

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. That means that 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 limit of the sum of the base pair length of its elements. Sequences defined by NCBI with an accession version are the smallest subunit to be clustered.

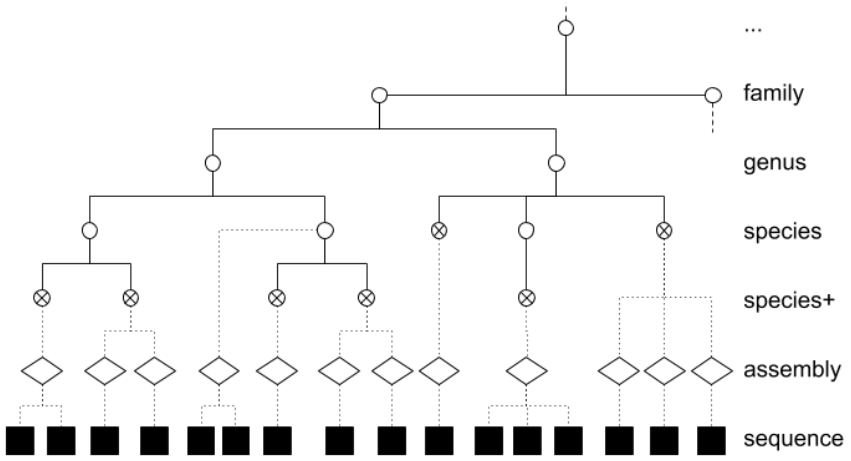

Figure 2: Classification levels and taxonomic distribution**.** Empty circles are inner nodes of the tree; marked circles are leaf nodes (also referenced in this manuscript as "taxid" nodes); full lines represent taxonomic relations, dotted lines represent the extension of the taxonomic classification 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 sequences 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 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, requiring bigger filters. The higher the taxonomic level, the lower the number of groups sharing k-mers, thus 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. To remove sequences from the IBF, the sub-vectors affected should be rebuilt.

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.

Classifying

The ganon read classification is based on the well-studied k-mer counting lemma [Jokinen, P.; Reinert, K. 2015]. All k-mers from given reads are looked up on the indices previously generated. If a minimum number of mismatches between the read and the reference is achieved, a read is set as 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.

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:

$$kmers_R = l_R - k + 1$$

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

$$kcount_R = kmers_R - k \cdot e$$

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

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 $kcount_R$ 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 mismatches are kept and so on up to the maximum number of errors allowed (e in equation $kcount_R$). Similar filtration methods were previously used in read mappers such as Yara. If a read is classified in several references within the same range of mismatches, they are all going to be

reported since it is not possible to define which one has 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 by a different error rate for reads that matched against just one reference group. Such filter will be applied afters the standard filtration and will reclassify a low scored read to its parent taxonomic level. That is useful for a filtering in very low levels (e.g. assembly) since the classification in those levels should be more precise with less mismatches. Ganon supports classification based on multiple indices in a user-defined hierarchy.

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.

Results

We evaluate ganon against a well-established method for read classification: kraken [Wood, D. E.] and also against a new version called krakenhll [Breitwieser, F. P. 2018] 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 [Kim, D.] which uses the Burrows-Wheeler transform (BWT) and the Ferragina-Manzini (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, abundance estimation or posterior assembly 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 complete genomes from Archaeal and Bacterial organisms dating from 26-September-2017 and a complete set of all genomes of Archaeal and Bacterial organisms dating from 25-September-2017 (Table 1). Taxonomic information was obtained on the same dates as the sequences.

	Base pairs	**# species**	**# assemblies**	**# sequences**
RefSeq CG	31.344.025.155	2.991	8.053	15.250
RefSeq ALL	383.786.471.018	15.527	95.891	9.861.599

Table 1: Reference Sequences used to generate indices

For the classification we use data produced for the first CAMI Challenge [Sczyrba, A.]. Sets of simulated 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 analyse. 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 (Table 2).

	simulated			real		
	# samples	Read len.	Total size	# samples	Read len.	Total size
low	1	2 x 100bp	15 Gbp	1	2 x 150bp	15 Gbp
medium	2	2 x 100bp	31.5 Gbp	2	2 x 150bp	40 Gbp
high	5	2 x 100bp	75 Gbp	5	2 x 150bp	75 Gbp

Table 2: Evaluated datasets

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 original level or at a higher level, this read is marked as a true positive. For example: if a read has an assembly origin but the tool outputs it at its correct species, the read is a true positive for the species level. False positives are reads with a wrong direct classification or a false high level classification. Reads with too-specific classifications are also counted as false positives (e.g. tool classifies a read at assembly level but the ground truth only defines it at species).

Indexing

The set of sequences from RefSeq CG and RefSeq ALL (Table 1) 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).

We built three different indices for ganon in different depths of classification: assembly, taxonomic leaf nodes (taxid) and species level. The size of the index was defined based on a minimum false positive rate. Ganon indices get smaller in higher taxonomic ranks, due to redundancy of sequences within the taxonomic groups. Centrifuge and krakenhll were run with default parameters. Centrifuge indices allow classification up to a taxonomic or sequence level (assembly must be deduced from sequences) and krakenhll indices allow classifications up to assembly and sequence level. Kraken, which can do only taxonomic level classification, was evaluated in two ways: default build and reduced build size limited to 64GB, a size comparable to the ganon-taxid index.

When indexing the RefSeq CG (Table 3, Figure 3), the evaluated tools took between 28 minutes and 7 hours, with ganon being the fastest and centrifuge the slowest. Building ganon indices for taxid or assembly level classification requires a higher memory usage compared to the species level due to the necessity of bigger filters. However ganon shows a significant overall reduction in memory consumption and run-time compared to the other tools: 6 times faster than kraken 64, the second fastest. Centrifuge achieves the lowest index size with the cost of having the highest memory and time consumption.

	Maximum depth of classification	RefSeq CG			RefSeq ALL		
		time	Memory (GB)	Index size (GB)	time	Memory (TB)	Index size (GB)
centrifuge	sequence	7:02:17	283	**14**	Ca. 11d	0.73	**211**
ganon	species	**0:27:47**	**52**	48	2:55:31	**0.27**	259
ganon	taxid	0:28:07	69	66	**2:53:32**	0.35	338
ganon	assembly	0:29:56	93	88	2:54:58	0.47	451
kraken	taxid	5:21:52	177	147	Ca. 11d*	2.5*	N.A
kraken 64	taxid	2:55:25	177	64	N.A.	N.A.	N.A.
krakenhll	sequence/ assembly	5:40:47	177	147	N.A.	N.A.	N.A.

Table 3: Build times using 24 threads. *data obtained from Nasko, D. J (2018) based on bacterial RefSeq version 80. kraken and krakenhll indices for RefSeq ALL are not available (N.A.) due to very large memory requirements. Computer specifications: 128 logical cpus from 8 x Intel Xeon E5-4667 v4 (2,2 GHz), 512 GiB RAM, 3 TiB SSD running Debian 8.11

Indexing the RefSeq ALL was more demanding for all tools. Centrifuge and kraken took around 11 days with high memory consumption, ranging from 730GB to 2.5TB, respectively. Due to computational limitations, we could not run kraken on our infrastructure. Here, kraken values are approximate (obtained from Nasko DJ et al) and could not be further evaluated using this reference set. Ganon built such indices in less than 3 hours, a speed-up of approximately 90 times over the other tools. Ganon also requires 2 to 7 times less memory when compared to centrifuge and kraken, respectively. Centrifuge, however, achieves the smallest index size for the RefSeq ALL data.

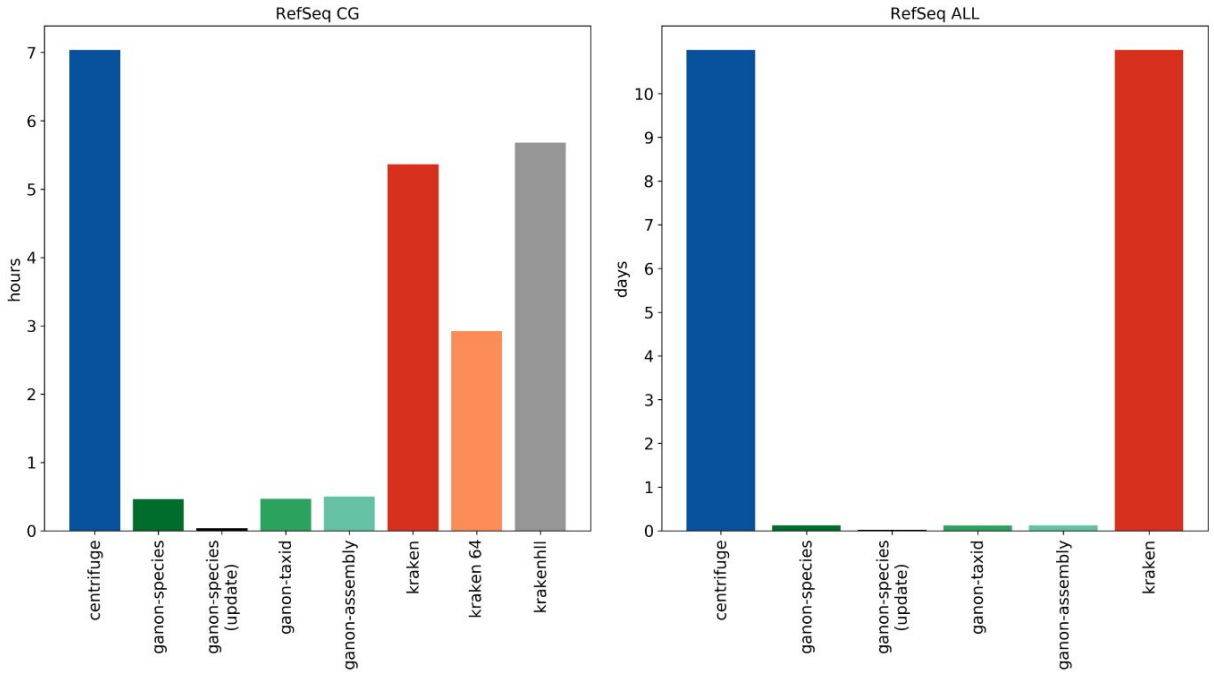

Figure 3: build and update times for indexing RefSeq CG (hours) and RefSeq ALL (days)

Ganon is the only tool among the evaluated ones which allows updates on previously generated indices. We evaluated this functionality with all *E. Coli* sequences added to RefSeq dating from 26-September-2017 to 19-December-2017 (0.23 Gbp, 155 sequences). Updating the ganon-species index based on RefSeq CG with this dataset took 2 min 11 seconds with the same memory requirements for building the index (Table 3). On RefSeq ALL, ganon took around 28 minutes to update the index.

Classifying

The classification results were evaluated in terms of sensitivity and precision at each taxonomic level based on the binary classification. The values are always cumulative up to the evaluated level, for example: sensitivity and precision at the genus level will consider all the classifications up to genus level (species, species+ and assembly). 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.

The results for the CAMI simulated and real datasets can be interpreted in groups (Table 4). First, by database: all evaluated tools for the RefSeq CG and only ganon and centrifuge for RefSeq ALL, since kraken and krakenhll requirements for such dataset were not practicable in our computational environments. Second, by tool considering their maximum depth of classification: centrifuge-taxid, ganon-taxid, kraken and kraken 64 classify reads up to the taxonomic leaf nodes (taxid). ganon-assembly, centrifuge-assembly and krakenhll up to assembly level. ganon-species was also included to show how the results changes when classifying at a higher level with a small index. Given the availability of the ground truth, 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.

depth of classification		simulated		real	
		RefSeq CG	RefSeq ALL	RefSeq CG	RefSeq ALL
species	ganon-species	✔	✔	✔	✔
taxid	centrifuge-taxid	✔	✔	✔	✔
	ganon-taxid	✔	✔	✔	✔
	kraken	✔	✖*	✔	✖*
	kraken 64	✔	✖*	✔	✖*
assembly	centrifuge-assembly	✔	✔	✖**	✖**
	ganon-assembly	✔	✔	✖**	✖**
	krakenhll	✔	✖*	✖**	✖**

Table 4: Comparison groups. *) kraken and krakenhll indices for RefSeq ALL are not available. **) real reads do not contain assembly level ground truth data

Figure 4 compares the results for one of the simulated high complexity datasets using the indices based on RefSeq CG and RefSeq ALL. With RefSeq CG all tools performed similarly in terms of sensitivity within their depth of classification, with ganon-species achieving a better classification at maximum species level. In terms of precision, ganon shows superior performance in every scenario when compared to its competitors. At assembly level ganon has approximately half of the false positives of krakenhll and centrifuge while having a very similar number of true positives. Using the indices based on the RefSeq ALL both ganon and centrifuge show big improvements compared to the RefSeq CG, given the broader diversity covered by this set. In terms of sensitivity both

tools performed once more very similarly, while ganon achieved better precision with very few false positives. Ganon-species had the highest precision of approximately 97% at species level.

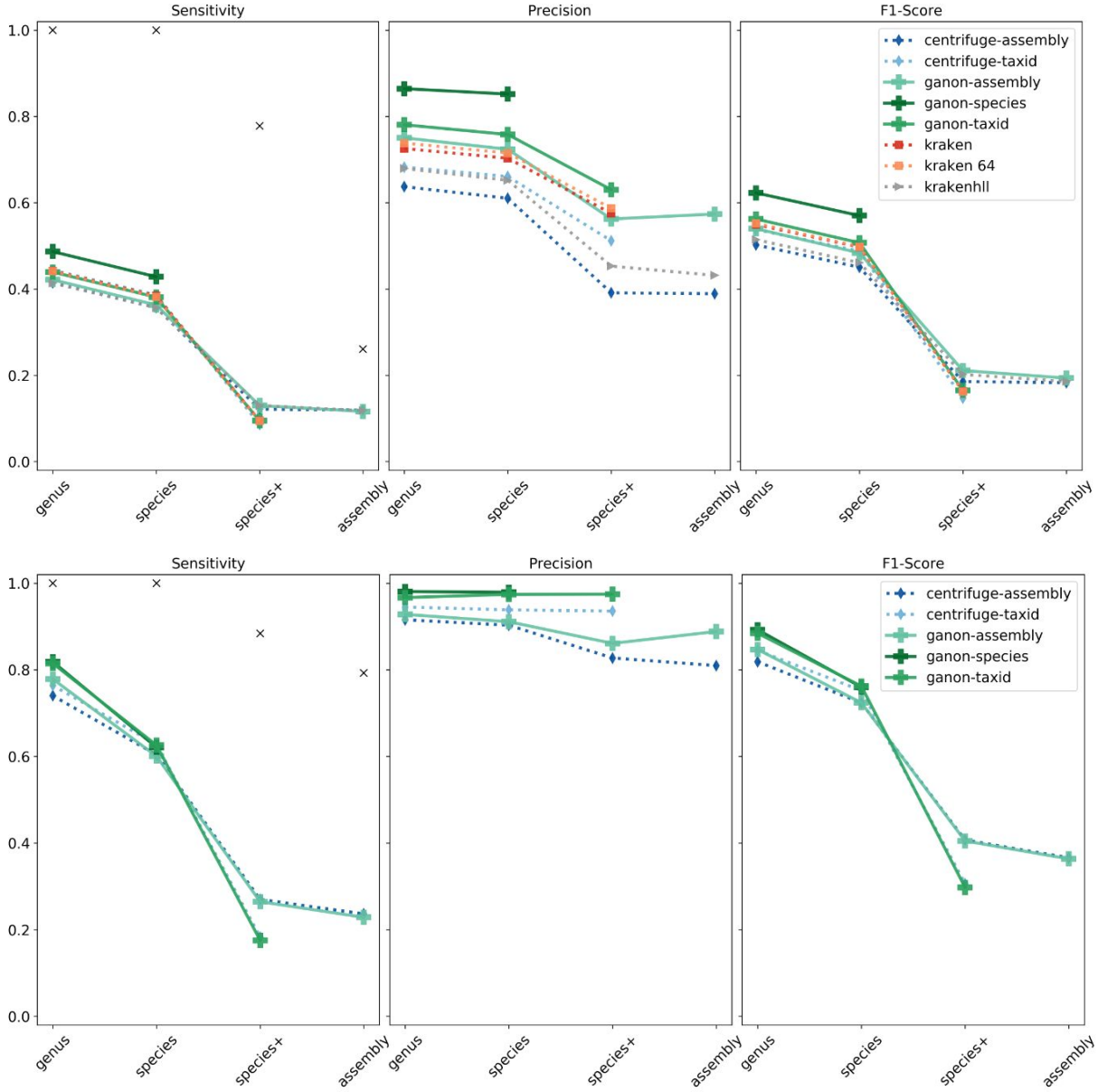

Figure 4: simulated reads - high complexity sample 1. Results on the top with RefSeq CG and on the bottom with RefSeq ALL. Results up to assembly level. Black markers show the maximum sensitivity possible given the ground truth classification.

The same analysis was performed on real data. This set is more challenging, since most of the organisms are novel and not yet present in the public repositories. As stated by the CAMI results [Sczyrba, A.], most tools performed poorly in this dataset in terms of sensitivity, as depicted in Figure 5. The results follow the same trend from the previous analysis, with both ganon and centrifuge being similarly sensitive while ganon being more precise with less false positives. Here the use of a larger set of references significantly improved the results in terms of sensitivity and precision, demonstrating

that the use of more references is crucial for newly sequenced environments. Ganon and centrifuge enable such analysis. In addition, our evaluated indices were based only on Archaeal and Bacterial genome sequences, while the real dataset also contains viral and other elements, explaining part of the bad performance of the tools.

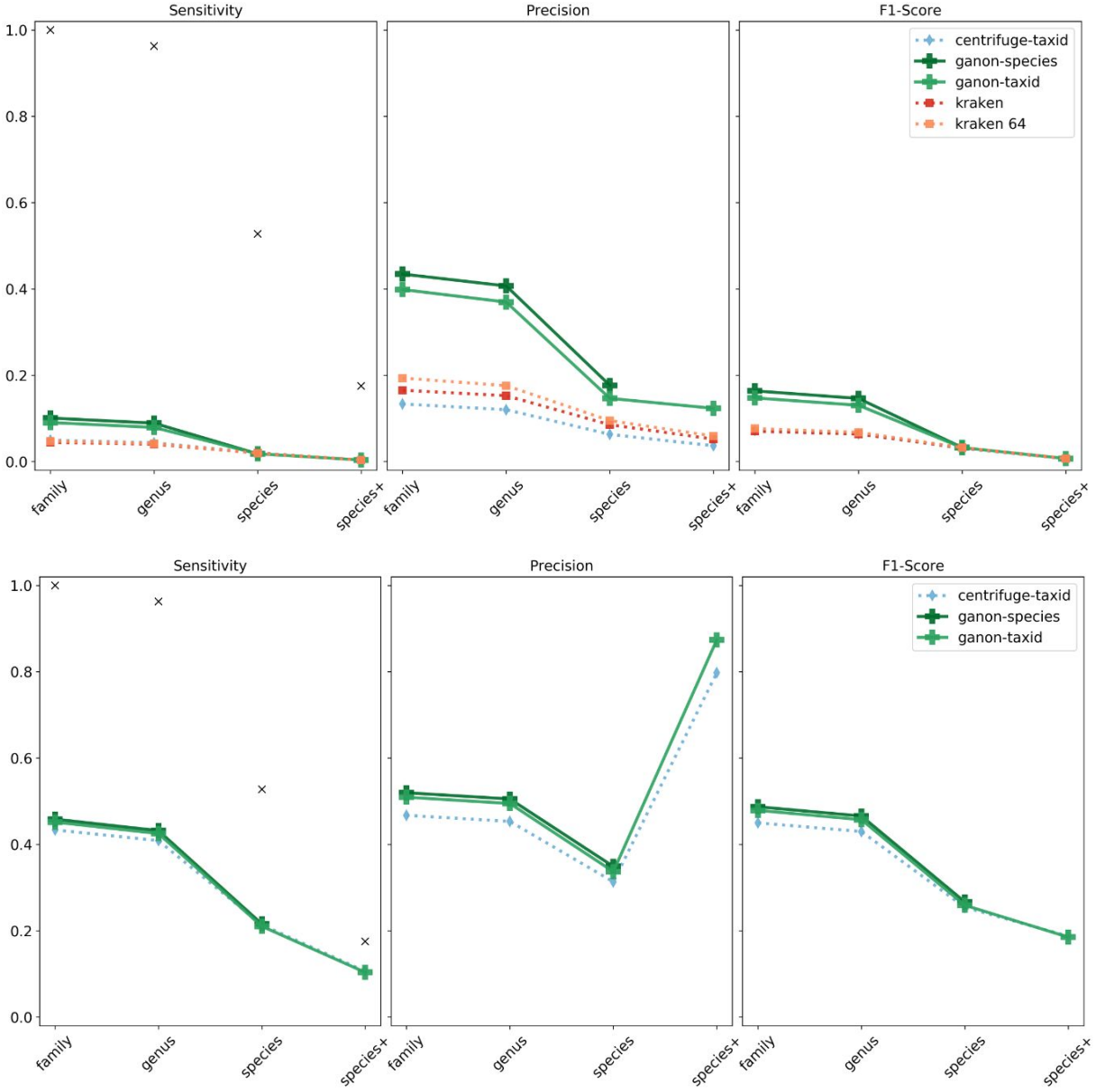

Figure 5: real reads - high complexity sample 1. Results on the top with RefSeq CG and on the bottom with RefSeq ALL. Results up to species+ level. Black markers show the maximum sensitivity possible given the ground truth classification.

Table 5 compares the performance of the analysed tools in terms of how many base pairs they can classify per minute. Memory usage is mainly based on the index size of each tool, with little variation besides that. Kraken is the tool with the fastest performance on the classification when using the RefSeq CG indices, followed by ganon-species and centrifuge. When classifying against RefSeq ALL, centrifuge has the fastest classification time followed by ganon-species. The slow down on bigger indices is reasonable, given that RefSeq ALL indices cover 5x more species, 11x more assemblies,

12x more base pairs and 646x more sequences than the smaller reference set (Table 1). Ganon classification time is around 5 times slower on species and taxid level compared to the RefSeq CG. Centrifuge performs around 3x slower in this reference set. A trend in the classification performance is clear, the bigger the index the slower the classification. This trend is evident when using very large indices based on the RefSeq ALL.

	RefSeq CG	RefSeq ALL
centrifuge	428*	150*
ganon-species	481	80
ganon-taxid	379	71
ganon-assembly	324	30
kraken	952	-
kraken 64	1057	-
krakenhll	465	-

Table 5: Classification performance of 1 million reads in Mbp/m with 48 threads. * performance estimation based on "Multiseed full-index search" time report without considering post-processing time. Computer specifications: 128 logical cpus from 8 x Intel Xeon E5-4667 v4 (2,2 GHz), 512 GiB RAM, 3 TiB SSD running Debian 8.11

Discussion

We presented ganon, a novel method to index references and classify short reads for environmental samples based on a taxonomic oriented set of references. Ganon strength lies in providing an ultra-fast indexing method for very large sets of reference sequences and a high classification precision. 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.

Ganon indices are also flexible and can be built for high taxonomic levels, requiring less space and memory, consequently improving classification speed. Evaluations show that indices built to classify at maximum species level had the best results in terms of sensitivity and precision. A trade-off between filter size and false positive rate is also possible, sacrificing precision over performance.

Ganon classification results are on par with state-of-the-art methods with regard to sensitivity, while improving precision rates in every scenario of our evaluations. 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.

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 analysing metagenomic data. Even though centrifuge performed well with more reference sequences, the time to index such set is highly prohibitive. In addition the method does not provide a direct one-match-per-read result and an in-house post-processing step was necessary to achieve it. The indexing issue also applies to kraken and krakenhll. Ganon manages to index big set of references sequences in very short time and classifications results are as good or better than the evaluated tools. 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.

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.

References

Balvočiūtė, M., & Huson, D. H. (2017). SILVA, RDP, Greengenes, NCBI and OTT — how do these taxonomies compare? *BMC Genomics*, *18*(S2), 114. http://doi.org/10.1186/s12864-017-3501-4

Benson, D. A., Cavanaugh, M., Clark, K., Karsch-Mizrachi, I., Ostell, J., Pruitt, K. D., & Sayers, E. W. (2017). GenBank. *Nucleic Acids Research*, (November), 1–7. http://doi.org/10.1093/nar/gkx1094

Bloom, B. H. (1970). Space/time trade-offs in hash coding with allowable errors. *Communications of the ACM*, *13*(7), 422–426. http://doi.org/10.1145/362686.362692

Breitwieser, F. P., Lu, J., & Salzberg, S. L. (2017). A review of methods and databases for metagenomic classification and assembly. *Briefings in Bioinformatics*, (August), 1–15. http://doi.org/10.1093/bib/bbx120

Breitwieser, F. P., & Salzberg, S. L. (2018). KrakenHLL: Confident and fast metagenomics classification using unique k-mer counts. *bioRxiv*, 262956. http://doi.org/10.1101/262956

Codenotti, B., De Marco, G., Leoncini, M., Montangero, M., & Santini, M. (2004). Approximation algorithms for a hierarchically structured bin packing problem. *Information Processing Letters*, *89*(5), 215–221. http://doi.org/10.1016/j.ipl.2003.12.001

Dadi, T. H., Siragusa, E., Piro, V. C., Andrusch, A., Seiler, E., Renard, B. Y., & Reinert, K. (2018). DREAM-Yara: An exact read mapper for very large databases with short update time. *bioRxiv*, 256354. http://doi.org/10.1101/256354

Federhen, S. (2012). The NCBI Taxonomy database. *Nucleic Acids Research*, *40*(D1), D136–D143. http://doi.org/10.1093/nar/gkr1178

Haft, D. H., DiCuccio, M., Badretdin, A., Brover, V., Chetvernin, V., O'Neill, K., … Pruitt, K. D. (2018). RefSeq: an update on prokaryotic genome annotation and curation. *Nucleic Acids Research*, *46*(D1), D851–D860. http://doi.org/10.1093/nar/gkx1068

Huson, D. H., Auch, A. F., Qi, J., & Schuster, S. C. (2007). MEGAN analysis of metagenomic data. *Genome Research*, *17*(3), 377–86. http://doi.org/10.1101/gr.5969107

Jokinen, P., & Ukkonen, E. (1991). Two algorithms for approximate string matching in static texts. *Mathematical Foundations of Computer Science 1991*, 240–248. Retrieved from http://www.springerlink.com/index/p58155n8012x0477.pdf

Kim, D., Song, L., Breitwieser, F. P., & Salzberg, S. L. (2016). Centrifuge: rapid and sensitive classification of metagenomic sequences. *Genome Research*, *26*(12), 1721–1729. http://doi.org/10.1101/gr.210641.116

Li, X., Naser, S. A., Khaled, A., Hu, H., & Li, X. (2018). When old metagenomic data meet newly sequenced genomes, a case study. *PLOS ONE*, *13*(6), e0198773. http://doi.org/10.1371/journal.pone.0198773

Lindgreen, S., Adair, K. L., & Gardner, P. P. (2016). An evaluation of the accuracy and speed of metagenome analysis tools. *Scientific Reports*, *6*, 19233. http://doi.org/10.1038/srep19233

McIntyre, A. B. R., Ounit, R., Afshinnekoo, E., Prill, R. J., Hénaff, E., Alexander, N., … Mason, C. E. (2017). Comprehensive benchmarking and ensemble approaches for metagenomic classifiers. *Genome Biology*, *18*(1), 182. http://doi.org/10.1186/s13059-017-1299-7

Menzel, P., Ng, K. L., & Krogh, A. (2016). Fast and sensitive taxonomic classification for metagenomics with Kaiju. *Nature Communications*, *7*, 11257. http://doi.org/10.1038/ncomms11257

Mukherjee, S., Seshadri, R., Varghese, N. J., Eloe-Fadrosh, E. A., Meier-Kolthoff, J. P., Göker, M., … Kyrpides, N. C. (2017). 1,003 reference genomes of bacterial and archaeal isolates expand coverage of the tree of life. *Nature Biotechnology*, *35*(7), 676–683. http://doi.org/10.1038/nbt.3886

Nasko, D. J., Koren, S., Phillippy, A. M., & Treangen, T. J. (2018). RefSeq database growth influences the accuracy of k-mer-based species identification. *bioRxiv*, 1–21. http://doi.org/10.1101/304972

Parks, D. H., Rinke, C., Chuvochina, M., Chaumeil, P.-A., Woodcroft, B. J., Evans, P. N., … Tyson, G. W. (2017). Recovery of nearly 8,000 metagenome-assembled genomes substantially expands the tree of life. *Nature Microbiology*, *2*(11), 1533–1542. http://doi.org/10.1038/s41564-017-0012-7

Pavlopoulos, Oulas, A., Pavloudi, C., Polymenakou, P., Papanikolaou, N., Kotoulas, G., … Iliopoulos, I. (2015). Metagenomics: Tools and Insights for Analyzing Next-Generation Sequencing Data Derived from Biodiversity Studies. *Bioinformatics and Biology Insights*, 75. http://doi.org/10.4137/BBI.S12462

Peabody, M. A., Van Rossum, T., Lo, R., & Brinkman, F. S. L. (2015). Evaluation of shotgun metagenomics sequence classification methods using in silico and in vitro simulated communities. *BMC Bioinformatics*, *16*(1), 362. http://doi.org/10.1186/s12859-015-0788-5

Reinert, K., Dadi, T. H., Ehrhardt, M., Hauswedell, H., Mehringer, S., Rahn, R., … Weese, D. (2017). The SeqAn C++ template library for efficient sequence analysis: A resource for programmers. *Journal of Biotechnology*, *261*(July), 157–168. http://doi.org/10.1016/j.jbiotec.2017.07.017

Reinert, K., Langmead, B., Weese, D., & Evers, D. J. (2015). Alignment of Next-Generation Sequencing Reads. *Annual Review of Genomics and Human Genetics*, *16*(1), 133–151. http://doi.org/10.1146/annurev-genom-090413-025358

Sczyrba, A., Hofmann, P., Belmann, P., Koslicki, D., Janssen, S., Dröge, J., … McHardy, A. C. (2017). Critical Assessment of Metagenome Interpretation—a benchmark of metagenomics software. *Nature Methods*, *14*(11), 1063–1071. http://doi.org/10.1038/nmeth.4458

Tausch, S. H., Strauch, B., Andrusch, A., Loka, T. P., Lindner, M. S., Nitsche, A., & Renard, B. Y. (2018). LiveKraken—real-time metagenomic classification of illumina data. *Bioinformatics*, (August), 1–3. http://doi.org/10.1093/bioinformatics/bty433

Truong, D. T., Franzosa, E. A., Tickle, T. L., Scholz, M., Weingart, G., Pasolli, E., … Segata, N. (2015). MetaPhlAn2 for enhanced metagenomic taxonomic profiling. *Nature Methods*, *12*(10), 902–903. JOUR. http://doi.org/10.1038/nmeth.3589

Wood, D. E., & Salzberg, S. L. (2014). Kraken: ultrafast metagenomic sequence classification using exact alignments. *Genome Biology*, *15*(3), R46. http://doi.org/10.1186/gb-2014-15-3-r46