LEMMI: A Live Evaluation of Computational Methods for Metagenome Investigation

Mathieu Seppey* ORCID: 0000-0003-3248-011X

Mosè Manni* ORCID: 0000-0002-4146-6523

Evgeny M. Zdobnov*

*Department of Genetic Medicine and Development, University of Geneva Medical School and Swiss Institute of Bioinformatics, Geneva, Switzerland

Corresponding author: E-mail: evgeny.zdobnov@unige.ch

Keywords

Metagenome, metagenomics, marker genes, classification, binning, taxonomy, reference, benchmarking, reproducibility, containerization, bioboxes, CAMI, pipeline.

https://lemmi.ezlab.org/

Abstract

Metagenomics promotes our understanding of microbial communities. The methodology profoundly relies on computational interpretation of the sequencing reads in the light of our evolving understanding of microbial taxonomy. The fast growing volumes of such readouts and the reference databases prompted development of a multitude of computational procedures over recent years. Only a subset of these has appeared in independent benchmarking assessments, which are also quickly becoming obsolete. This is obscuring an informed choice of a method for biologists as well as the impact of certain innovations for method developers.

Here we present the LEMMI benchmarking platform (https://lemmi.ezlab.org) enabling the continuous integration of taxonomic profilers and binners, their assessment on a variety of data, as well as dynamic presentation of their ranking according to alternative objectives of required analyses or technical limitations. The platform is container-based and exploits the ability of many methods to construct a reference database on demand to compare them under highly controlled conditions and with identical inputs. Curated references can also be provided by method developers, enabling the advantages of particular databases to be highlighted. LEMMI creates a link between method developers and their users. It provides automated and unbiased benchmarking that are valuable to both. Moreover, the standardised and easy-to-use containers created by us or developers can be downloaded by metagenomic tool users, who will get exactly the same software as it appears in evaluation.

Introduction

Over the last decade, assessing complete microbial communities has emerged as a key component of medical and environmental research. Microbiologists, who had previously relied on culture to conduct their experiments, started to perceive the true magnitude of the unseen majority that does not grow when taken away from their habitat (Whitman et al., 1998). Sequencing has now become common practice for investigating targeted genomes using high-throughput technologies (Goodwin et al., 2016), which has enabled metagenome research, a young but rapidly growing area interested in reaching all organisms found in a microbial community through sequencing. One of its important goals is to identify and quantify all organisms present in a sample. This can, for instance, refine the action of clinicians and environmental scientists by associating key organisms to the stability of a given environment (Ainsworth et al., 2015), detecting a variety of pathogens (Petty et al., 2014), or linking variations in microbial composition to host phenotypic differences (Qin et al., 2012). The approach that first demonstrated the feasibility of targeting a ubiquitously conserved marker gene to obtain a prokaryotic phylogeny, the 16s ribosomal ribonucleic acids (16s rRNA) gene (Woese et al., 1977), is now commonly applied to bacterial classification. A more recent approach that has become popular with the decreasing costs of whole genome sequencing (WGS) consists of exploiting the whole DNA or RNA content of the sample without targeted amplification. WGS avoids the bias induced by the affinity of the primers in PCR-based methods, allows to get beyond the limited phylogenetic resolution of 16s markers (Eloe-Fadrosh et al., 2016), and to reach bacteria, viruses, fungi and other eukaryotes in a single analysis. This methodology is known as WGS-metagenomics, or simply metagenomics. To assign a taxonomic label and estimate the relative abundance of reads or genome copies, both approaches require raw or assembled reads to be queried against a reference of known sequences mapped to a taxonomic classification (Federhen, 2012; McDonald et al., 2012; O'Leary et al., 2016; Quast et al., 2012; Wang et al., 2007). Taxonomy-independent binning of reads or contigs sharing similar features that may correspond to *ab-initio*-defined "species", usually called operational taxonomic units (OTUs), are outside the current scope of the present work.

The complexity inherent in metagenomes has triggered the prolific development of methods dedicated to this field. Alignment-based techniques such as BLAST (Camacho et al., 2009) can be applied to the output of amplicon-based and metagenomics sequencing (Huson et al., 2007), but they are limited by an inability to query billions of reads against thousands of known genomes in a reasonable time. Alignment-free approaches, that proxy alignment by composition of the reads in terms of exact subsequences (n- or k-mers) as well as Burrows-Wheeler transform indexes, can process reads orders of magnitude faster (Ounit et al., 2015). In spite of this, the rate at which sequences representing new organisms are made available (RefSeq: 24,000 bacteria in May 2014, 53,000 in December 2018. https://www.ncbi.nlm.nih.gov/refseq/statistics/) now exceeds the potential of many tool users to have access to computing resources able to deal with comprehensive references for running their analyses. Consequently, other methods have focused on systematically reducing the reference material while trying to keep the representation of the existing diversity as accurate as possible (Kim et al., 2016; Nasko et al., 2018), and others have focused on producing curated references (Truong et al., 2015).

The number of published metagenome assessment methods has soared, raising the question of how the methods compare and how to make an informed choice when designing specialized pipelines, e.g. for clinical practitioners. As in other fields of bioinformatics (The Assemblathon, Bradnam et al., 2013; CASP, Moult et al., 2018), comparative benchmarking has become a “must-have”. Furthermore, it is a publishing requirement for novel method papers to include a benchmarking section in which the tool is compared to existing competitors, focusing on the innovation and strengths of the newcomer (Dilthey et al., 2018; Müller et al., 2017; Piro et al., 2018). While this kind of “benchmarking” is a useful starting point, its real ability to judge the actual merits of the tool in addressing the whole range of scenarios it is expected to cover remains questionable. For a fairer comparison, independent benchmarking studies (i.e. in which the authors do not directly promote their tool) have been published on several occasions (Lindgreen et al., 2016; McIntyre et al., 2017; Peabody et al., 2015). A key component of any benchmarking effort is that of the common input sample used during the analysis, which has to be described as comprehensively as possible to constitute the ground truth of the evaluation. Mock *in-silico* datasets created from public genomes are a convenient method, but

have the disadvantage of containing matches to existing references. This can be mitigated by using artificially evolved strains (Lindgreen et al., 2016) or clade exclusion, i.e. removing from the reference the sequences that are in the reads, when the methods can reprocess it (Peabody et al., 2015). To fully exclude this bias, the Critical Assessment of Metagenome Interpretation (CAMI), one of the largest efforts to date in benchmarking various metagenome-related problems, used organisms unreleased at the time of the evaluation to conduct a challenge taking place over months, open to the community, leading to a collaborative publication that introduced many valuable resources required for efficient benchmarking (Sczyrba et al., 2017). While informative to assess the potential of technical advances and crucial to define the common goals that need to be pursued by all tools, one-shot publications have not succeeded in clearly identifying the best implementations of most methods by not following the pace of a fast-evolving field. Many projects have remained out of such benchmarks, which has prevented novel or updated methods from getting the visibility they deserve at the time they could be beneficial. Moreover, a point ignored by these studies is the ability of a method to last beyond the time of evaluation and publication of a benchmarking paper. For instance, despite being considered by CAMI as relevant methods, some tools were not well adopted by the community during the year following the publication of the benchmark results (Table 1). This may be in part explained by a lack of technical maintenance, documentation, or advertising, as many bioinformatics tools cannot be easily reused some time after their publication (Mangul et al., 2018). Finally, as taxonomic classification is not only affected by the choice of the method (i.e. algorithm), but strongly directed by the choice of the reference sequences, a fair and comprehensive evaluation needs to judge the ability of different solutions to exploit expanding reference material, monitoring the effects on different families of methods of the exponential growth of sequence databases (Nasko et al., 2018).

Given the limitations of previous benchmarking projects in addressing these critical aspects, we introduce LEMMI, a Web-based platform that hosts a semi-automated benchmarking pipeline that enables continuous tracking and evaluation of newly published methods for metagenome taxonomic classification (its first “Live” dimension). Many tools can complete multiple tasks (in the present context, binning and profiling), likely with unequal performances, which often leads tool users to seek the optimal trade-off for their specific field (i.e. Velsko et al., 2018). LEMMI does not present distinct, non comparable categories, but maintains a ranked evaluation of the strengths and weaknesses of each tool on multiple aspects of the metagenome classification problem. These aspects can be explored dynamically through the Web interface (the second “Live” dimension). When a method cannot deal with raw genomes as reference, a set of markers or preprocessed sequences can be provided as custom reference by the developer. Otherwise, LEMMI evaluates the ability of a tool to build a custom reference that can be used in the subsequent benchmarking process, therefore assessing methods under highly controlled and comparable conditions. By using a containerized approach resembling a previously suggested format (Belmann et al., 2015; Sczyrba et al., 2017) as the unique way of taking part to the evaluation, we strongly encourage developers to engage in standardizing their methodology in order to evaluate their tools or pipelines in a neutral environment under different conditions (references, datasets) and parameters (e.g. k-mer size, score thresholds) as well as being readily available to the tool users. The computational resources needed to process a reference and conduct mock sample analyses are evaluated together with the accuracy of the corresponding results to obtain a complete overview. In order to simulate unsequenced strains from public data, the LEMMI platform can generate mock reads from publicly available genomes, taking care of excluding the source material from the reference subsequently provided to the tools (referred as the “leave-out” approach). It also includes some of the available mock datasets previously seen in published studies (Bokulich et al., 2016; Sczyrba et al., 2017). The platform, with detailed results, documentation, and evaluated containers, is available on https://lemmi.ezlab.org/.

Table 1 | Apparent lack of correspondence between the CAMI ranking and citations. Number of citations for methods included in the profiling category of the first CAMI challenge compared to methods not included. The tools presented on the top section were ranked as first or second best for at least one metric in the category taxonomic profilers. The number of citations is based on the most cited paper that can be related to the method or an update according to Google Scholar (18/12/2018). Kraken was included in CAMI, but in a different category, Kaiju was first released in late 2015, Centrifuge and Kraken's companion tool Bracken were released in the first half of 2016.

Tool	Citations in 2018	Paper
Methods evaluated as profilers by CAMI:		
MetaPhlAN	134	Truong et al., 2015
Metaphyler	15	Liu et al., 2011
Clark	71	Ounit et al., 2015
common kmer (metapalette)	2	Koslicki and Falush, 2016
Focus	8	Silva et al., 2014
Taxy-pro	3	Klingenberg et al., 2013
Quikr	3	Koslicki et al., 2013
Methods not evaluated as profilers by CAMI:		
Kraken (when used with Bracken)	367	Wood and Salzberg, 2014
Centrifuge	63	Kim et al., 2016
Kaiju	78	Menzel et al., 2016

Results

Workflow

The central component of our benchmarking process is the method or a combination of methods wrapped into a container (see https://www.docker.com/) to complete the two tasks that will be called by the LEMMI pipeline during the assessment and that can be downloaded by the tool users (Fig. 1). One task performs the analysis of single or paired-end reads in FASTQ format using a reference and provides either a taxonomic profile describing relative abundance, a taxonomic binning report, or both. The other task is to process provided FASTA files (nucleotides and/or proteins) along with a mapping to the taxonomy to create an output folder containing a reference compatible with the method implemented in the previous task, to be kept and reused afterwards. The implementation of the latter is optional but highly recommended if the method does not require a specific curated reference. Otherwise, a preprocessed reference can be provided along with the container, e.g. to demonstrate the advantage of a curated database. The taxonomic rank of interest is provided as a parameter, as some tools process the reference in a rank-specific manner (Ounit et al., 2015). The taxonomic ranks that have to be supported in the initial version of the LEMMI platform are genus and species. To achieve the evaluation of the tool, the container is loaded and ran on the pipeline to complete the construction of a reference and the evaluation of several datasets. Such containers not only are essential to the benchmarking process, but are a useful resource when made available to the tool user. A detailed guideline to build containers is included in the LEMMI user guide on https://gitlab.com/ezlab/lemmi/wikis/userguide. The results of the benchmark are presented in details on the Web interface of the LEMMI platform (Fig. 2, Fig. 3, Fig. 5).

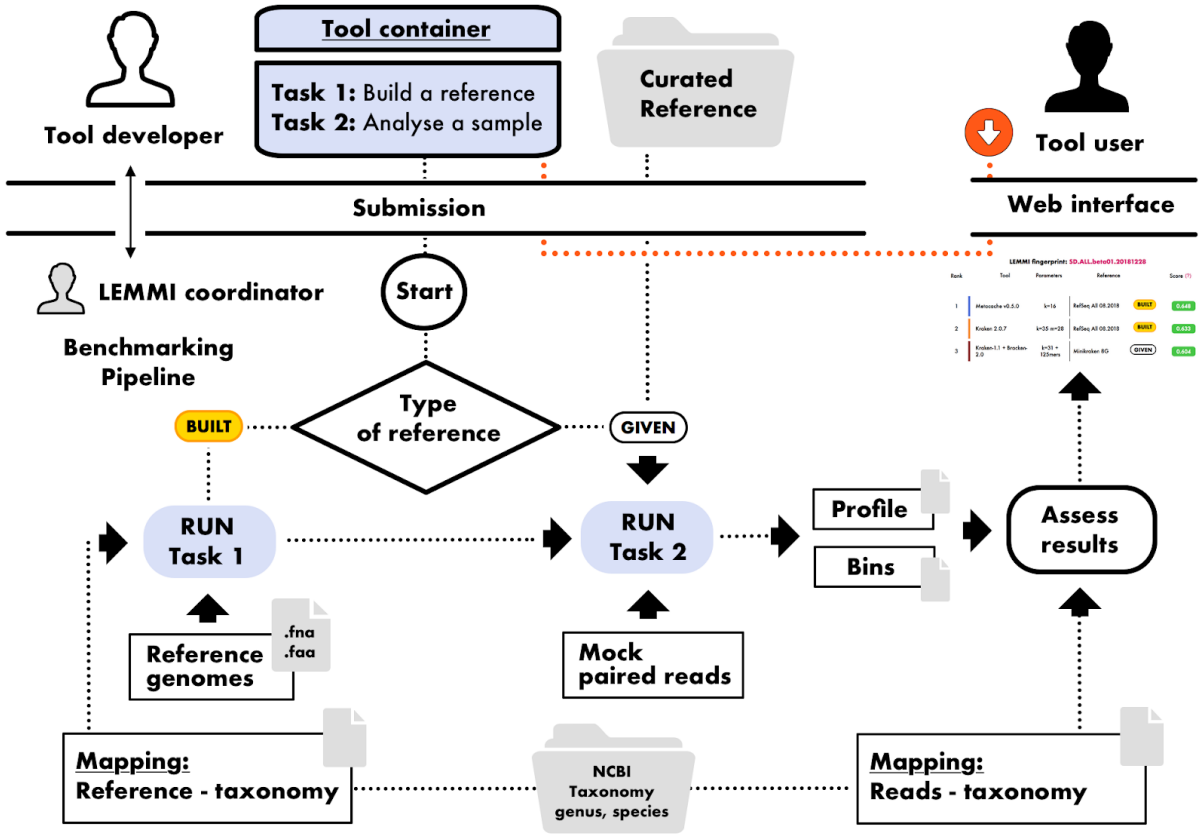

Fig. 1 | Workflow. Tool developers prepare a LEMMI container for their method following the provided guideline, to complete two tasks: building a reference on provided genomic FASTA files (task 1), and analysing FASTQ samples to provide a profile and binned reads (task 2). They can provide a curated reference if their method is unable to complete task 1. Their containerized method is then managed by the LEMMI coordinator to complete all tasks required to process all datasets and appear in the ranking. Multiple runs to explore parameters and references can be conducted using a single container. The tool users can browse the results to define which methods best suit their needs and obtain the corresponding containers to conduct additional tests or actual analyses, with the guarantee of identical file formats and similar behaviors shared by all methods.

Tools and references included

We conceived the platform to encourage developers to submit their up-to-date methods by themselves. Nevertheless, we added a selection of tools as a proof of concept. We created demo LEMMI containers for Metaphlan2 (Truong et al., 2015), Kraken 2, Kraken 1 combined with Bracken (Lu et al., 2017; Wood and Salzberg, 2014), Metacache (Müller et al., 2017), and Kaiju (Menzel et al., 2016). We used provided reference for Metaphlan2 and Kraken 1, and built controlled reference for all other tools using RefSeq (O'Leary et al., 2016) assemblies nucleotide or protein files. When provided with 245GB of RAM, only Kraken 2 (any parameters) and Metacache (when k=16) were able to process the entirety of the 124,520 fasta files available in the LEMMI RefSeq repository for the bacterial and archeal domains (RefSeq All 08.2018). Therefore, a subset of 18,916 file constituting one file per species taxid was used as a smaller representative reference (RefSeq 1 rep. 08.2018). Furthermore, to evaluate the benefits of the increase of sequences over time, the repository was subsampled back in time, to keep only the sequences available at the end of 2015, for a total of 52,051 files for the bacterial and archeal domains (RefSeq All 12.2015). The latter still covers respectively 95% and 94% of the species diversity present in the in-house generated datasets, LEMMI_RefSeq_201805_001 and LEMMI_RefSeq_201805_002.

Interactive interface permits a quick overview to a detailed understanding

The visualisation of the results starts on the homepage with a list of methods (Fig. 2a) ranked according to a score expressing multiple metrics averaged over all available datasets. The LEMMI platform user can generate a different ranking by changing the weight of a variety of parameters to better fit their expectations (e.g. focused on recall, ignoring precision, giving more importance to abundance estimation or maximizing the amount of classified reads) (Fig. 2ab). The LEMMI platform offers different presets and a "fingerprint" to facilitate keeping track of rankings over time when mentioned in a publication by developers and tool users. Two different types of reference exist and

can be displayed individually, “BUILT” that stands for a reference constructed during the benchmarking process on a controlled source material, a firm guarantee of an unbiased evaluation, and “GIVEN” when an external reference was provided with the container. Among the criteria used to generate the ranking, a section dedicated to the resources needed to perform the computation can be enabled to penalize tools consuming too much time and memory (Fig. 2c). As all included datasets are not comparable in their composition, and to provide to the LEMMI platform user a real understanding of what is behind the ranking, a detailed view of all metrics is plotted for each dataset (to date, we offer five: LEMMI_RefSeq_201805_001, LEMMI_RefSeq_201805_002, Mockrobiota-17, CAMI_I_LOW, CAMI_I_HIGH_1), along with information about the composition of these datasets (Fig. 3abc). The metrics are displayed separately for the species and genus ranks, and the interface allows the LEMMI platform user to toggle the visibility of each tool separately, as well as zooming in and out to disentangle overlapping data points. Finally, several metrics are computed according to different levels of low coverage filtering (i.e. taxa having less than 10/100/1000 reads are ignored in both the candidate profile and the ground truth) to emphasize the benefits of applying such strategy on the output of high recall but low precision tools, such as Kaiju (Fig. 3b).

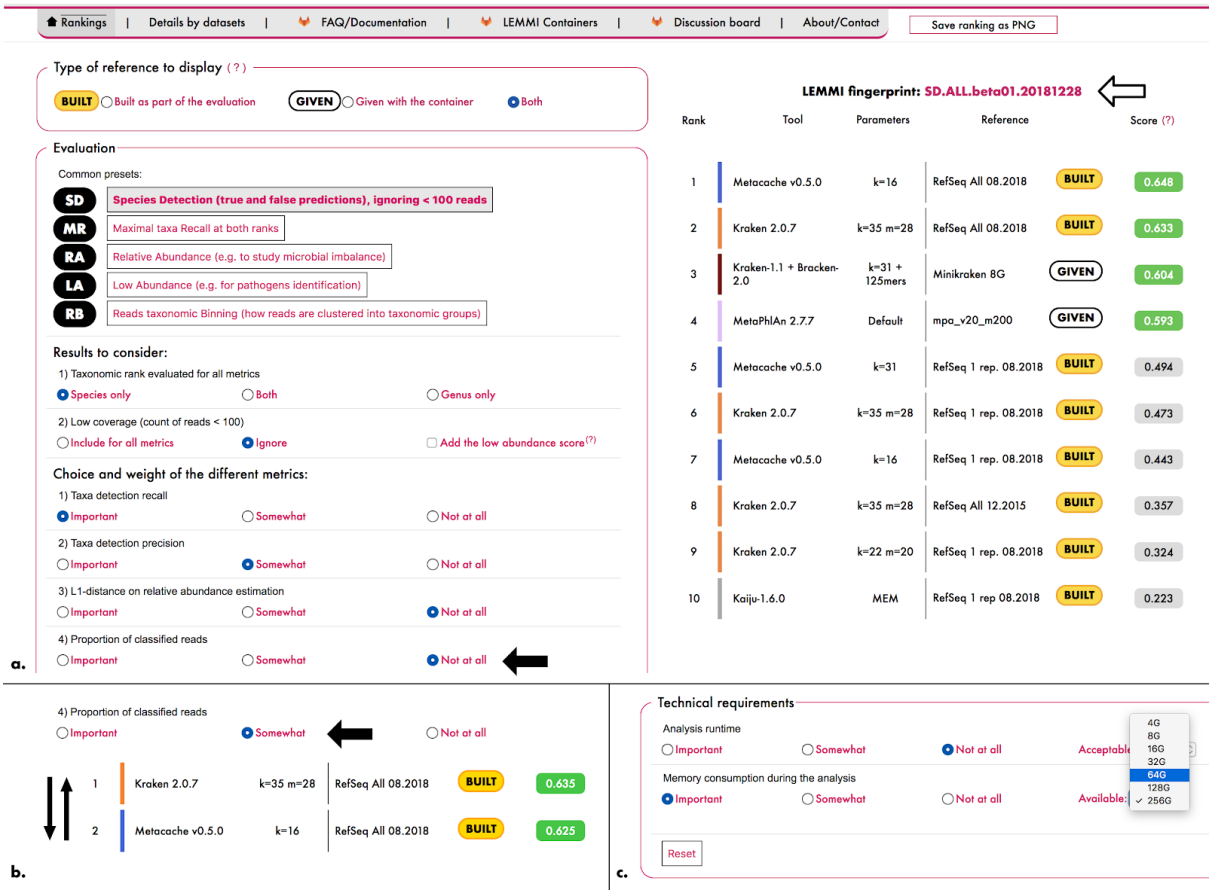

Fig. 2 | Main interface, the dynamic ranking. a, The interface on the left allows the LEMMI platform users to edit specific criteria to modify the default ranking. Five suggested sets of parameters are available, generating a unique fingerprint that can be used to refer to a specific ranking at a specific date and version (as indicated by the white arrow). **b**, The LEMMI platform users can freely edit parameters to correspond to their needs, thus affecting the ranking. The black arrows highlight a choice that swapped the top ranked tools. **c**, Among the metrics that can be selected to create a custom ranking, the time and memory consumption filters can be enabled, and a maximum acceptable value can be set.

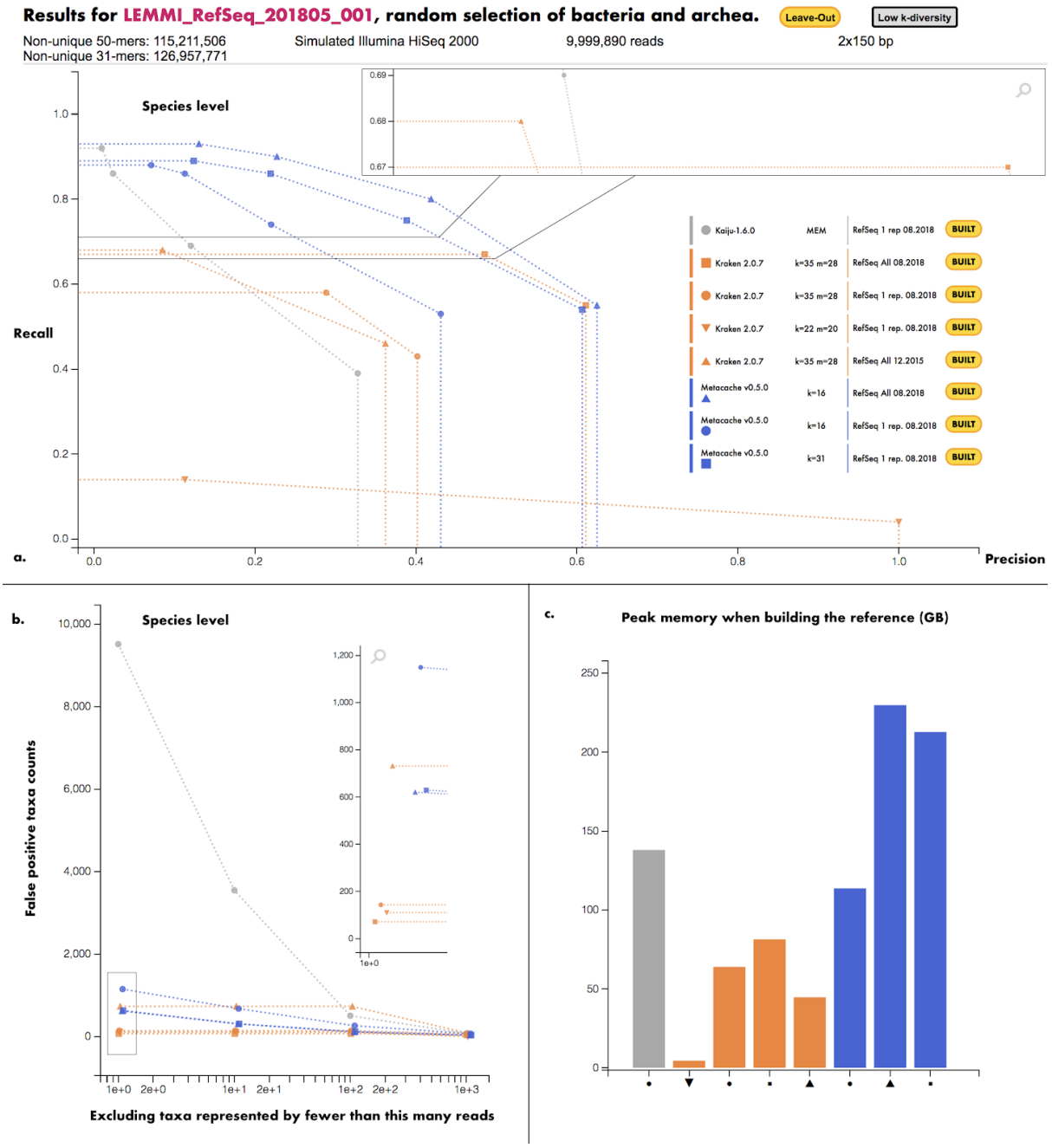

Fig. 3 | Example panels from the details section of an individual dataset, LEMMI_RefSeq_201805_001. Three of the 16 detailed plots available for each dataset at each taxonomic rank. The Web interface allows the LEMMI platform user to toggle the visibility of each tool and freely zoom in and out. **a**, List of tools for all panels. Precision and recall curve for eight combinations of references, parameters, and tools, illustrating that a greater precision and recall can be achieved when using the complete RefSeq repository instead of a single representative genome for each species. Metacache, ran here with two difference k-mer sizes, outcompetes Kraken 2 in terms of recall, even when using a reduced representation of the reference. It also reaches a better precision than Kraken 2 when using the complete RefSeq repository as reference. **b**, Counts of false

positive taxa detected for the same eight tools. When filtering under 100 reads, the false positives produced by Kaiju decrease dramatically. **c**, Amount of memory required by each tool to build their reference, showing the higher performances of kraken 2 to that aspect.

Evaluation under highly controlled conditions

While the GIVEN mode opens the benchmark to methods relying on manually curated references, which authorizes the evaluation to be primarily guided by the reference, the BUILT benchmarking mode of the LEMMI pipeline places all tools under the same conditions, excluding any bias that would have to be investigated on a case-by-case basis, incompatible with a systematic, semi-automated approach. Therefore, any deprecated taxid, missing reference, or on the contrary, similarity between reads and reference, will equally affect each and every candidate method for all datasets evaluated in this mode. However, as different families of methods may be affected differently by identical sequences in reads and reference (e.g. alignment-based or composition methods) and these are unlikely to represent close-to-real-life scenarios, the LEMMI pipeline introduces another level of control when assessing the datasets generated in-house using RefSeq assembly genomes. It functions in a leave-out mode, excluding the genomic or protein files corresponding to the accessions used to generate reads (Fig. 4). The two existing LEMMI_RefSeq datasets contains 100 species each issued from several genomes, which can still be fully recovered as they have multiple representatives in the RefSeq repository in addition to those used to produce the reads. These datasets are more complex (their non-unique 50-mers diversity is 115M) than the microbiota-17 dataset (68M), but almost an order of magnitude less diverse than the CAMI 1 high complexity datasets (940M). All results presented on Fig. 3abc are issued from these highly controlled runs, based on comparable reference and leave-out approach.

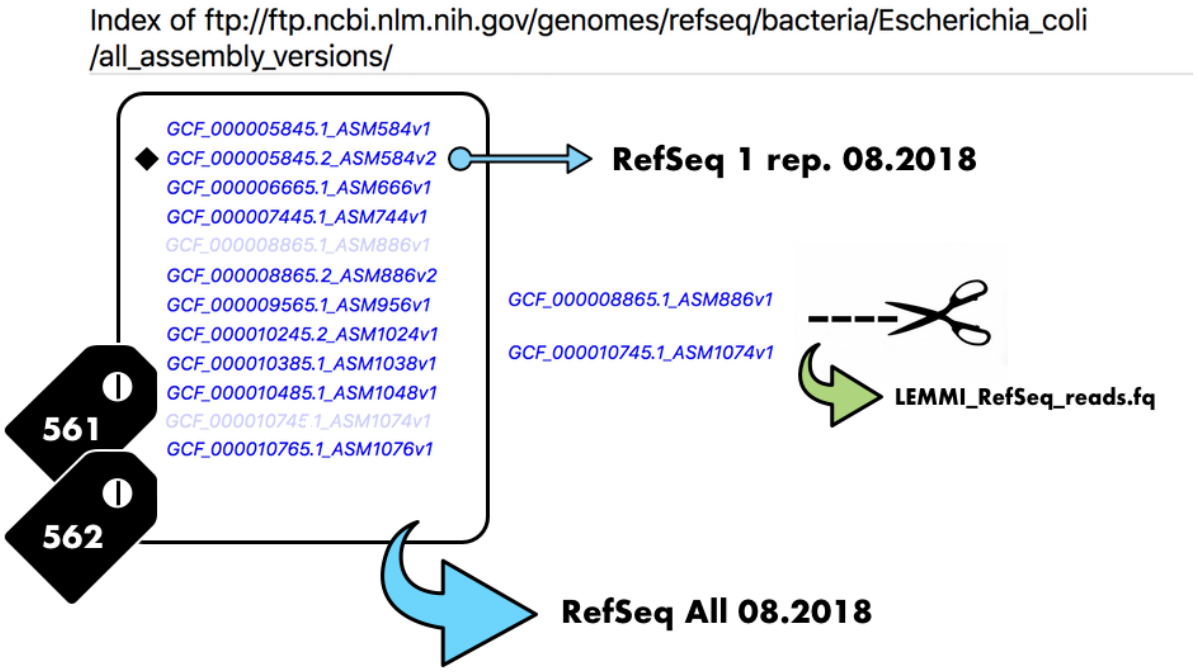

Fig. 4 | The leave-out approach used when running the BUILT mode on LEMMI in-house datasets. A toy example illustrating how representative strains of *Escherichia coli* are selected to create a scenario where the taxon can be identified at the species and genus level. Two accessions are taken out, and the corresponding files are chunked to generate the mock reads. All other files are used as material for building the reference if the method can handle a large amount of files. Otherwise, one representative is picked (◆). As all sequences share the same label (NCBI taxid=561|562, rank=genus|species), a correct classification is expected for the reads at species or higher ranks. This approach can be applied at any taxonomic rank, i.e. excluding all representatives of the species while keeping representatives of its genus (NCBI taxid=561) artificially creates unknown clades that can only produce false predictions at low levels.

Multiple runs to explore several references and parameters

A LEMMI container can accept multiple parameters, defined by the method, which in the context of benchmarking allows successive runs to explore the range of possibilities offered by the tool. This feature is illustrated on Fig. 3a that shows the different effects of varying the k-mer size on Kraken 2 and Metacache. The latter is less negatively affected by a decrease in the value of k (from 31 to 16) than Kraken 2 (from 35 to 22), which loses most of its recall. Furthermore, the marked difference in memory consumption between runs with different k-mer sizes (e.g. k=31 and k=16) allows Metacache with k=16 to be run on a machine with less available memory, or to build a much larger reference which results in an increase of both precision and recall (Fig. 3ac). On the other hand, Kraken 2 is the ideal tool to explore the impact of varying the reference on last common ancestor (LCA) based classifiers, as it can quickly process large amount of files while still using less memory than all other tools (Fig. 3c). Our runs tend to agree with Nasko et al., 2018 who raised the potential issue of losing the ability to reach low ranks with LCA k-mers methods with the increased representation of the same species in the public repositories. When contrasting a reference limited to 2015 and a complete from 2018 to proceed to the analysis of the LEMMI_RefSeq_001 dataset, Kraken 2 identifies one extra species (68 vs 67) with a reference limited to 2015 (Fig. 3a), despite having only 95 species represented out of 100. The cost, however, is a greater amount of false positive (Fig. 3b). On the contrary to what is stated in Nasko et al., 2018, this effect is even more pronounced at the genus level, with six extra genera recovered from LEMMI_RefSeq_001 with the reference from 2015 (with 74 out of 78 genera represented). The second in-house dataset, LEMMI_RefSeq_002, does not replicate this finding, with the limited reference recovering three fewer species (when covering 94 out of the 100 species) and two fewer genera. Furthermore, this cannot be seen at all on the datasets issued from the CAMI challenge (Fig. 5a), as many of the species that compose its microbial community seem absent in RefSeq in 2015. Additionally, the tools ran with their embedded database performed very poorly on the CAMI sets at the species level (Fig. 5a). While we cannot exclude the absence of the representation of some species in the source material (Minikraken, created in October 2017 and mpa_v20_m200, created in November 2017), when compared to Kraken 2 and Metacache with a

complete reference from mid-2018 or even from 2015, and given the proportion of reads classified at the species level (Fig. 5c), it is likely that increasing the number of k-mers in the dataset content has a major impact on the ability of very reduced references to deal with the analysis (Minikraken is a subset of 5% of the k-mer in RefSeq, Metaphlan2 uses marker genes). Minikraken performs very well on the other, less complex datasets (Fig. 5d). It probably benefits from escaping the leave-out step when analysing the LEMMI in-house datasets, having the advantage of the source of the reads as reference.

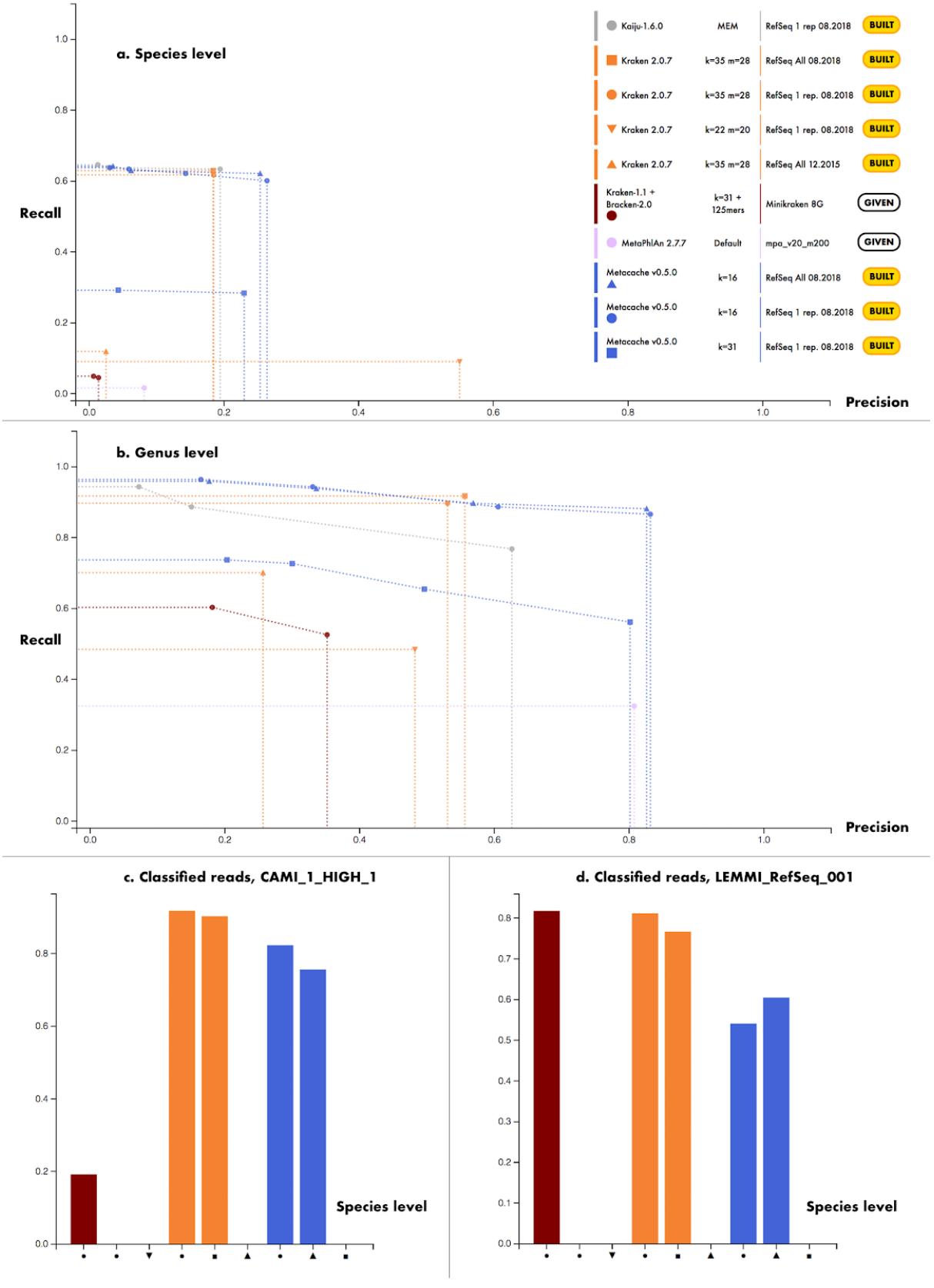

Fig. 5 | Performance of various tools when assessing the CAMI_1_HIGH_1 dataset. a, List of tools for all panels. Precision and recall curve at the species level, showing that Kraken 2 with a

reference from 2015, Kraken 1 with Minikraken, and Metaphlan 2 perform poorly compared with tools exploiting the entirety of RefSeq mid-2018. **b**, Equivalent at the genus level. **c**, Proportion of classified reads at the species level for Kraken 1 using Minikraken, Kraken 2 and Metacache based on several subsets of RefSeq. All other methods were hidden through the LEMMI Web interface, which helps to focus only on tools of interest. **d**. Equivalent for the LEMMI_RefSeq_001 dataset at the species level.

Discussion

A continuous benchmark that includes reference processing

In this paper, we describe a Web interface that maintains a dynamic and multi-criteria ranking for tools dedicated to metagenome taxonomic profiling and binning. It lies on the top of a hosted evaluation workflow (Fig. 1) exploiting containers that wrap candidate methods able to construct their reference using provided genomic files (optional) and conduct their analyses using this previously built reference or a provided one. The analyses are conducted on a variety of mock samples gathered from different external sources or generated in-house by the pipeline. While forthcoming benchmarking efforts are expected to focus on new technological and field-specific challenges with interdisciplinary scientific teams (Bremges and McHardy, 2018), our solution aims at filling the major time gap existing between benchmarking publications and the absence of a solution to monitor the uninterrupted flow of new developments with an efficient way to visualize the results, filter the performances, and examine all coexisting metrics at once. Its continuous integration approach allows novel methods to appear in an independent evaluation at the time of their release, loosening the dependency to others' benchmarking agenda. This will not only provide benefits to developers, by helping to publish and promote their tools, but most of all to tool users who have to choose the appropriate method, and who are sometimes at a loss when seeing multiple forked repositories based on similar tools claiming technical refinements (Breitwieser et al., 2018; Břinda et al., 2015) along with newcomers' publications that remain absent from benchmarks. Once populated with a significant number of recent methods, the LEMMI platform will help to unravel several practical questions associated with the choice of references, methods, and parameters, to maximize the benefits of selecting metagenomic classifiers.

A standardized problem for heterogeneous tools

During the design of the LEMMI platform, the choice was made to define a unique category of problem (i.e. providing both taxonomic binning and profiling) to allow efficient comparisons of tools or combination of tools (e.g. kraken + bracken (Lu et al., 2017; Wood and Salzberg, 2014)) which have multiple features, strength and weaknesses. They can be combined in a pipeline to constitute a complete solution to run analyses only once and obtain various results, evaluated here as a whole, ultimately promoting the navigation by dynamic ranking offered by the LEMMI platform (tools, parameters, and reference as one entry in the list). If a solution is not able to provide a required information, it is maintained in the list with a poor score only when the corresponding metric is considered as important. Great care was taken here to not compare apples with oranges by defining standardized inputs, outputs, controlled reference whenever possible and clear expectations. One of the major advantages of our platform is that by accepting both tools that can process imposed genomic material as input and those that rely on a curated reference, clearly tagged as such, we are able to include in the evaluation most of the methods that exist and to provide detailed metrics including computational resources usage. As a proof of concept, we present some of the well-adopted softwares in the field of metagenome sciences.

Future plans and perspectives

As sequencing technologies improve and new methods appear with new claims, moving the scope of the standardized problem stated above, the minimal offer for a generalist benchmarking solution has to evolve. The initial release of this platform focuses on datasets representing bacterial and archeal assessment of short read sequences, at the genus and species level as defined by the NCBI taxonomy. However, we have designed our platform to be flexible in order to include new datasets, new metrics, and be ready to potentially incorporate more profound changes such as a different taxonomic classification system (Parks et al., 2018). After establishing the state of the art of the methods available to date, the next milestone will be to offer an assessment at lowest levels by defining a way to match different representative genomes of the same strain, which cannot be achieved by the NCBI taxonomy anymore (https://www.ncbi.nlm.nih.gov/books/NBK431007/).

By contrasting the advantages of the LEMMI datasets with those offered by reusing previously published datasets, we demonstrated the potential of creating both the reference and the reads from a shared controlled source, to disconnect the outcome of the evaluation from the publication state of specific sequences, allowing a continuous enrichment of the platform with new tools still facing identical benchmarking conditions. The now published CAMI 1 datasets were useful to highlight the impact of complex datasets on the outcome of the analysis below the genus level when using reduced references. It shows that our offer of in-house datasets has to be completed with complex datasets, encompassing unknown clades corresponding to different taxonomic ranks, for the LEMMI platform to provide rankings of tools able to model a use in close-to-real-life scenarios. Even though our datasets are generated from public data, the details of the sampling cannot remain permanently hidden. Therefore, the composition of our in-house datasets will be made public in time, to be replaced by newly generated ones, freezing the current ranking and moving the platform to the next release. This moment will give the opportunity to decide whether each method should be migrated (i.e. by rerunning the container on the new release), updated, or discarded. The periodicity of these releases has still to be defined, but should correspond to the introduction of new features required to follow the method

developments and expand the LEMMI platform offer. The improvements that are likely to be included in the near future are sequences representing long reads technologies and benchmarking methods for the analysis of metaviromes.

To conclude, we think that appearing side-by-side in a trusted ranking with established competitors under clear and validated criteria is essential to bring credibility to any newly published method in order to be adopted by their target audience. This represents a trustable baseline before introducing additional scenarios designed to promote the novelty of the method on specific, previously poorly addressed problems. Therefore, we strongly encourage developers or advanced tool users to encapsulate their scripts in a container compatible with our platform and take the lead to appear in the present ranking (https://lemmi.ezlab.org/contact.html), especially prior to enter a publishing process. Achieving this and providing a container publicly will ensure that the method is sustainable for being used on most user's environment, who will get "what they see on benchmark", including the crucial part dedicated to custom reference processing, which is not always straightforward. The technology of containerization is gaining interest among the bioinformatics community (da Veiga Leprevost et al., 2017), with Docker as main technology. The approach used by the LEMMI platform will certainly be revised in due time to cope with the evolution and requests of the tool users and developers, with the ultimate goal of promoting a standardized way to efficiently evaluate and share bioinformatics solutions.

Methods

Structure of the pipeline

An in-house python3-based controller (McKinney, 2010; Oliphant, 2015) coordinates the many subtasks required to generate datasets, run the candidate containers, and compute the statistics (Fig. 1). Snakemake 5.3.1 (Koster and Rahmann, 2012) is used to supervise individual subtasks such as generating a dataset or running one evaluation. The process is semi-automated through configuration files, designed to allow a potential full automation through a Web application. To be easily deployable, the benchmarking pipeline itself is wrapped in a Docker container. The plots presented on the user interface are generated with the mpld3 library (Hunter, 2007, https://mpld3.github.io)

LEMMI containers

The LEMMI containers are implemented for Docker 18.09.0-ce. They partially follow the design introduced by http://bioboxes.org/ (Belmann et al., 2015; Sczyrba et al., 2017) as part of the CAMI challenge effort. The required output files are compatible with the profiling and binning format created for the CAMI challenge. Two tasks have to be implemented in order to generate a reference and conduct an analysis. To take part in the benchmark, a tool developer has to build the container on their own environment, while ensuring that both tasks can be run by an unprivileged user. A tutorial is available on https://gitlab.com/ezlab/lemmi/wikis/userguide. The containers or the sources to recreate them are available to the user.

Computing resources

During the benchmarking process, the container is loaded on a dedicated server and given 245GB of RAM and 32 cores. Reaching the memory limit will cause the container to be killed and the end of the

benchmarking process. All inputs and outputs are written on a local disk and the container is not connected to the Internet.

Taxonomy

The NCBI taxonomy is used to validate all entries throughout the process and unknown taxids are ignored. The framework etetoolkit (ETE3) (Huerta-Cepas et al., 2016) is used to query the taxonomy. The database was downloaded on 03/09/2018 and remains frozen to this version until a new release of the LEMMI platform.

RefSeq repository

All RefSeq assemblies for bacteria, archaea, and viruses were downloaded from ftp.ncbi.nlm.nih.gov/refseq/release (last download, 08/2018) with the conditions that they contained both a protein and a nucleotide file and that their taxid has a corresponding entry in the ETE3 NCBI taxonomy database, for a total of 132,167 files of each sequence type. Their taxonomic lineage for the seven main levels was extracted with ETE3. To subset the repository and keep one representative per species as inputs for the reference construction (1 rep. 08.2018), the list was sorted according to the assembly states (1:Complete Genome, 2:Chromosome, 3:Scaffold, 4:Contig) and the first entry for each species taxid was retained.

LEMMI datasets

To sample the genomes included in the LEMMI_RefSeq datasets, a custom python script was used to randomly select representative genomes for 100 species having more than one genome in the RefSeq assembly repository (05/2018), among bacterial and archaeal content, using complete assemblies or chromosome level assemblies. Their abundance was randomly defined following a lognormal distribution (mean=1, standard deviation=2.75). Additional low coverage species (abundance corresponding to < 100 reads) were manually defined and the total value was normalized to one to constitute a relative abundance profile. BEAR (Johnson et al., 2014) was used to generate

10,000,000 paired-end reads, 2x150bp, and DRISEE (Keegan et al., 2012) was used to extract an error profile from the SRA entry ERX2528389 to be applied onto the generated reads. The ground truth profile for the seven ranks, and bins for species and genus were kept. The non-unique 50-mers and 31-mers diversity of the obtained reads were generated with Jellyfish 2.2.8 (Marçais and Kingsford, 2011) on the concatenated pair of reads using the following parameters: jellyfish count -m 31 -s 3G --bf-size 5G -t 8 -L 1 reads.fq.

Additional datasets

The CAMI datasets were obtained from https://data.cami-challenge.org/ (09/2018) along with the metadata describing their content, already in the expected file format. The binning details were reprocessed to obtain distinct lists at the species and genus rank. The mockrobiota-17 dataset (Kozich et al., 2013) was obtained through https://github.com/caporaso-lab/mockrobiota and reprocessed to obtain a taxonomic profile in the appropriate format. No binning detail is available for this dataset, therefore no assessment of this aspect is based on this dataset. 50-mers and 31-mers diversity were computed as detailed above.

Metrics

The profile and binning reports are processed with OPAL 0.2.8 (Meyer et al., 2018a) and AMBER 0.7.0 (Meyer et al., 2018b) against the ground truth to obtain a wide range of metrics. The profiles of the candidate tools and the ground truth are filtered to ignore low coverage taxa (below 10/100/1000) and all metrics are computed at all these thresholds. The low abundance score is a custom metric calculated separately to evaluate the ability of the tool to correctly identify organisms present at very low coverage, but penalizing methods likely to recover them by recurrent report of the same taxids owed to very poor precision. To achieve this, as precision of low abundance organisms cannot be defined for a dataset (all false positives have a true abundance of zero and cannot be categorized as low abundance), the metric is computed by pairing two datasets to judge if a prediction can be trusted. The datasets (D1 and D2) include sets of taxa T1 and T2 that contain a subset of low abundance taxa

(T1_low and T2_low, < 100 reads coverage, T1_low ≠ T2_low). Each taxon belonging to T1_low identified in D1 increases the low abundance score of the tool for D1 (recall) only when it is not identified in D2 if absent from T2. Otherwise, a correct prediction of the taxon in D1 does not improve the score (as proxy for low abundance precision). The score (0.0 - 1.0) is processed from both sides (D1, D2), to obtain an independent score for each dataset. This metric is only defined for all LEMMI_RefSeq datasets (low abundance species: n=10, n=8, for LEMMI_RefSeq_001 and LEMMI_RefSeq_002, respectively). The runtime corresponds to the time in seconds during which the container is loaded. The memory is the peak value of total_rss memory reported when the container is loaded.

Ranking score

All metrics that are not already values between 0.0 and 1.0, with 1.0 being the best score, are transformed. The L1 distance is divided by its maximum value of 2.0 and subtracted from 1.0, the weighted UniFrac score is divided by its maximum value of 16.0 and subtracted from 1.0. The unweighted UniFrac score is divided by an arbitrary value of 25,000 and subtracted from 1.0. The memory and runtime are divided by 2x the maximum value (as defined by the user through the interface) and subtracted from 1.0, to obtain a range between 0.5 and 1.0. This approach allows to segregate methods that remain below the limit from those that exceed it and get the value 0.0. Any transformed metric below 0.0 or above 1.0 is constrained back to the respective value. The final score displayed in the ranking is the harmonic mean of all metrics, taken into account zero, one, or three times depending on the weight assigned to the metric by the user.

Author contributions

MS and EZ conceived the study. MS created the platform. MS and MM conducted the analyses. MS and MM wrote the documentation. MS, MM and EZ wrote the manuscript.

Acknowledgement

We would like to thanks all members of the Zdobnov group, in particular Christopher Rands for his useful comments. This work was supported by the Swiss National Science Foundation funding 31003A_166483.

References

Ainsworth, T.D., Krause, L., Bridge, T., Torda, G., Raina, J.-B., Zakrzewski, M., Gates, R.D., Padilla-Gamiño, J.L., Spalding, H.L., Smith, C., et al. (2015). The coral core microbiome identifies rare bacterial taxa as ubiquitous endosymbionts. ISME J. *9*, 2261–2274.

Belmann, P., Dröge, J., Bremges, A., McHardy, A.C., Sczyrba, A., and Barton, M.D. (2015). Bioboxes: standardised containers for interchangeable bioinformatics software. GigaScience *4*.

Bokulich, N.A., Rideout, J.R., Mercurio, W.G., Shiffer, A., Wolfe, B., Maurice, C.F., Dutton, R.J., Turnbaugh, P.J., Knight, R., and Caporaso, J.G. (2016). mockrobiota: a Public Resource for Microbiome Bioinformatics Benchmarking. MSystems *1*.

Bradnam, K.R., Fass, J.N., Alexandrov, A., Baranay, P., Bechner, M., Birol, I., Boisvert, S., Chapman, J.A., Chapuis, G., Chikhi, R., et al. (2013). Assemblathon 2: evaluating de novo methods of genome assembly in three vertebrate species. GigaScience *2*.

Breitwieser, F.P., Baker, D.N., and Salzberg, S.L. (2018). KrakenUniq: confident and fast metagenomics classification using unique k-mer counts. Genome Biol. *19*.

Bremges, A., and McHardy, A.C. (2018). Critical Assessment of Metagenome Interpretation Enters the Second Round. MSystems *3*.

Břinda, K., Sykulski, M., and Kucherov, G. (2015). Spaced seeds improve *k* -mer-based metagenomic classification. Bioinformatics *31*, 3584–3592.

Camacho, C., Coulouris, G., Avagyan, V., Ma, N., Papadopoulos, J., Bealer, K., and Madden, T.L. (2009). BLAST+: architecture and applications. BMC Bioinformatics *10*, 421.

Dilthey, A., Jain, C., Koren, S., and Phillippy, A. (2018). MetaMaps - Strain-level metagenomic assignment and compositional estimation for long reads. BioRxiv 372474.

Eloe-Fadrosh, E.A., Ivanova, N.N., Woyke, T., and Kyrpides, N.C. (2016). Metagenomics uncovers gaps in amplicon-based detection of microbial diversity. Nat. Microbiol. *1*, 15032.

Federhen, S. (2012). The NCBI Taxonomy database. Nucleic Acids Res. *40*, D136–D143.

Goodwin, S., McPherson, J.D., and McCombie, W.R. (2016). Coming of age: ten years of next-generation sequencing technologies. Nat. Rev. Genet. *17*, 333–351.

Huerta-Cepas, J., Serra, F., and Bork, P. (2016). ETE 3: Reconstruction, Analysis, and Visualization of Phylogenomic Data. Mol. Biol. Evol. *33*, 1635–1638.

Hunter, J.D. (2007). Matplotlib: A 2D Graphics Environment. Comput. Sci. Eng. *9*, 90–95.

Huson, D.H., Auch, A.F., Qi, J., and Schuster, S.C. (2007). MEGAN analysis of metagenomic data. Genome Res. *17*, 377–386.

Johnson, S., Trost, B., Long, J.R., Pittet, V., and Kusalik, A. (2014). A better sequence-read simulator program for metagenomics. BMC Bioinformatics *15*, S14.

Keegan, K.P., Trimble, W.L., Wilkening, J., Wilke, A., Harrison, T., D'Souza, M., and Meyer, F. (2012). A Platform-Independent Method for Detecting Errors in Metagenomic Sequencing Data: DRISEE.

PLoS Comput. Biol. *8*, e1002541.

Kim, D., Song, L., Breitwieser, F.P., and Salzberg, S.L. (2016). Centrifuge: rapid and sensitive classification of metagenomic sequences. Genome Res. *26*, 1721–1729.

Klingenberg, H., Aßhauer, K.P., Lingner, T., and Meinicke, P. (2013). Protein signature-based estimation of metagenomic abundances including all domains of life and viruses. Bioinformatics *29*, 973–980.

Koslicki, D., and Falush, D. (2016). MetaPalette: a *k* -mer Painting Approach for Metagenomic Taxonomic Profiling and Quantification of Novel Strain Variation. MSystems *1*.

Koslicki, D., Foucart, S., and Rosen, G. (2013). Quikr: a method for rapid reconstruction of bacterial communities via compressive sensing. Bioinformatics *29*, 2096–2102.

Koster, J., and Rahmann, S. (2012). Snakemake--a scalable bioinformatics workflow engine. Bioinformatics *28*, 2520–2522.

Kozich, J.J., Westcott, S.L., Baxter, N.T., Highlander, S.K., and Schloss, P.D. (2013). Development of a Dual-Index Sequencing Strategy and Curation Pipeline for Analyzing Amplicon Sequence Data on the MiSeq Illumina Sequencing Platform. Appl. Environ. Microbiol. *79*, 5112–5120.

Lindgreen, S., Adair, K.L., and Gardner, P.P. (2016). An evaluation of the accuracy and speed of metagenome analysis tools. Sci. Rep. *6*.

Liu, B., Gibbons, T., Ghodsi, M., Treangen, T., and Pop, M. (2011). Accurate and fast estimation of taxonomic profiles from metagenomic shotgun sequences. BMC Genomics *12*, S4.

Lu, J., Breitwieser, F.P., Thielen, P., and Salzberg, S.L. (2017). Bracken: estimating species abundance in metagenomics data. PeerJ Comput. Sci. *3*, e104.

Mangul, S., Mosqueiro, T., Duong, D., Mitchell, K., Sarwal, V., Hill, B., Brito, J., Littman, R., Statz, B., Lam, A., et al. (2018). A comprehensive analysis of the usability and archival stability of omics computational tools and resources. BioRxiv 452532.

Marçais, G., and Kingsford, C. (2011). A fast, lock-free approach for efficient parallel counting of occurrences of k-mers. Bioinformatics *27*, 764–770.

McDonald, D., Price, M.N., Goodrich, J., Nawrocki, E.P., DeSantis, T.Z., Probst, A., Andersen, G.L., Knight, R., and Hugenholtz, P. (2012). An improved Greengenes taxonomy with explicit ranks for ecological and evolutionary analyses of bacteria and archaea. ISME J. *6*, 610–618.

McIntyre, A.B.R., Ounit, R., Afshinnekoo, E., Prill, R.J., Hénaff, E., Alexander, N., Minot, S.S., Danko, D., Foox, J., Ahsanuddin, S., et al. (2017). Comprehensive benchmarking and ensemble approaches for metagenomic classifiers. Genome Biol. *18*, 182.

McKinney, W. (2010). Data Structures for Statistical Computing in Python. 6.

Menzel, P., Ng, K.L., and Krogh, A. (2016). Fast and sensitive taxonomic classification for metagenomics with Kaiju. Nat. Commun. *7*.

Meyer, F., Bremges, A., Belmann, P., Janssen, S., McHardy, A.C., and Koslicki, D. (2018a). Assessing taxonomic metagenome profilers with OPAL. BioRxiv 372680.

Meyer, F., Hofmann, P., Belmann, P., Garrido-Oter, R., Fritz, A., Sczyrba, A., and McHardy, A.C. (2018b). AMBER: Assessment of Metagenome BinnERs. GigaScience *7*.

Moult, J., Fidelis, K., Kryshtafovych, A., Schwede, T., and Tramontano, A. (2018). Critical assessment

of methods of protein structure prediction (CASP)-Round XII. Proteins Struct. Funct. Bioinforma. *86*, 7–15.

Müller, A., Hundt, C., Hildebrandt, A., Hankeln, T., and Schmidt, B. (2017). MetaCache: context-aware classification of metagenomic reads using minhashing. Bioinformatics *33*, 3740–3748.

Nasko, D.J., Koren, S., Phillippy, A.M., and Treangen, T.J. (2018). RefSeq database growth influences the accuracy of k-mer-based lowest common ancestor species identification. Genome Biol. *19*.

O'Leary, N.A., Wright, M.W., Brister, J.R., Ciufo, S., Haddad, D., McVeigh, R., Rajput, B., Robbertse, B., Smith-White, B., Ako-Adjei, D., et al. (2016). Reference sequence (RefSeq) database at NCBI: current status, taxonomic expansion, and functional annotation. Nucleic Acids Res. *44*, D733–D745.

Oliphant, T.E. (2015). Guide to NumPy.

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

Parks, D.H., Chuvochina, M., Waite, D.W., Rinke, C., Skarshewski, A., Chaumeil, P.-A., and Hugenholtz, P. (2018). A standardized bacterial taxonomy based on genome phylogeny substantially revises the tree of life. Nat. Biotechnol.

Peabody, M.A., Van Rossum, T., Lo, R., and Brinkman, F.S.L. (2015). Evaluation of shotgun metagenomics sequence classification methods using in silico and in vitro simulated communities. BMC Bioinformatics *16*.

Petty, T.J., Cordey, S., Padioleau, I., Docquier, M., Turin, L., Preynat-Seauve, O., Zdobnov, E.M., and Kaiser, L. (2014). Comprehensive Human Virus Screening Using High-Throughput Sequencing with a User-Friendly Representation of Bioinformatics Analysis: a Pilot Study. J. Clin. Microbiol. *52*, 3351–3361.

Piro, V.C., Dadi, T.H., Seiler, E., Reinert, K., and Renard, B.Y. (2018). ganon: continuously up-to-date with database growth for precise short read classification in metagenomics. BioRxiv 406017.

Qin, J., Li, Y., Cai, Z., Li, S., Zhu, J., Zhang, F., Liang, S., Zhang, W., Guan, Y., Shen, D., et al. (2012). A metagenome-wide association study of gut microbiota in type 2 diabetes. Nature *490*, 55–60.

Quast, C., Pruesse, E., Yilmaz, P., Gerken, J., Schweer, T., Yarza, P., Peplies, J., and Glöckner, F.O. (2012). The SILVA ribosomal RNA gene database project: improved data processing and web-based tools. Nucleic Acids Res. *41*, D590–D596.

Sczyrba, A., Hofmann, P., Belmann, P., Koslicki, D., Janssen, S., Dröge, J., Gregor, I., Majda, S., Fiedler, J., Dahms, E., et al. (2017). Critical Assessment of Metagenome Interpretation—a benchmark of metagenomics software. Nat. Methods *14*, 1063–1071.

Silva, G.G.Z., Cuevas, D.A., Dutilh, B.E., and Edwards, R.A. (2014). FOCUS: an alignment-free model to identify organisms in metagenomes using non-negative least squares. PeerJ *2*, e425.

Truong, D.T., Franzosa, E.A., Tickle, T.L., Scholz, M., Weingart, G., Pasolli, E., Tett, A., Huttenhower, C., and Segata, N. (2015). MetaPhlAn2 for enhanced metagenomic taxonomic profiling. Nat. Methods *12*, 902–903.

da Veiga Leprevost, F., Grüning, B.A., Alves Aflitos, S., Röst, H.L., Uszkoreit, J., Barsnes, H., Vaudel, M., Moreno, P., Gatto, L., Weber, J., et al. (2017). BioContainers: an open-source and community-driven framework for software standardization. Bioinformatics *33*, 2580–2582.

Velsko, I.M., Frantz, L.A.F., Herbig, A., Larson, G., and Warinner, C. (2018). Selection of Appropriate Metagenome Taxonomic Classifiers for Ancient Microbiome Research. MSystems *3*.

Wang, Q., Garrity, G.M., Tiedje, J.M., and Cole, J.R. (2007). Naive Bayesian Classifier for Rapid Assignment of rRNA Sequences into the New Bacterial Taxonomy. Appl. Environ. Microbiol. *73*, 5261–5267.

Whitman, W.B., Coleman, D.C., and Wiebe, W.J. (1998). Prokaryotes: The unseen majority. Proc. Natl. Acad. Sci. *95*, 6578–6583.

Woese, C.R., Fox, G.E., and Pechman, K.R. (1977). Comparative Cataloging of 16S Ribosomal Ribonucleic Acid: Molecular Approach to Procaryotic Systematics. Int. J. Syst. Evol. Microbiol. *27*, 44–57.

Wood, D.E., and Salzberg, S.L. (2014). Kraken: ultrafast metagenomic sequence classification using exact alignments. Genome Biol. *15*, R46.