LEMMI: A Live Evaluation of Computational Methods for Metagenome Investigation

Mathieu Seppey[1] ORCID: 0000-0003-3248-011X

Mosè Manni[1] ORCID: 0000-0002-4146-6523

Evgeny M. Zdobnov[1]*

[1]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

Metagenomics, taxonomic classification, read binning, reference database, benchmarking, reproducibility, bioboxes, bioinformatics, pipeline.

Abstract

LEMMI (https://lemmi.ezlab.org) is a benchmarking platform of computational tools for metagenome composition assessments that introduces three live aspects: a continuous integration of tools, their multi-objective ranking, and an effective distribution through software containers. We see this platform as a community-driven effort: method developers can showcase novel approaches and get unbiased benchmarks for publications, while users can make informed choices and obtain standardized and easy-to-use tools.

Metagenomics has made possible the study of previously undiscovered uncultured microorganisms. To use metagenomics to probe the vast hidden microbial landscape, we need effective bioinformatics tools, notably taxonomic classifiers for binning the sequencing reads (i.e. grouping and labeling) and profiling the corresponding microbial community (i.e. defining the relative abundance of taxa). This is computationally challenging, requiring time and resources to query shotgun sequencing data against rapidly expanding genomic databases. Users have to choose a solution among the plethora of methods that have been developed in a quest for accuracy and efficiency, for instance lowering the runtime and memory usage by reducing the reference material while maintaining the representative diversity[1,2]. However, individual tools are task-specific and their features rarely overlap entirely, making the establishment of universal rankings of methods ineffective. To date, at least one hundred published methods can be identified[3], among which new developments that may revolutionize the field cohabit with re-implementations of already explored strategies, fragmenting the community of users and hindering experimental reproducibility. Therefore, there is a need for a flexible platform that enables robust comparisons of individual methods and multi-step pipelines while considering differing objectives and computational limitations (e.g. accuracy versus runtime). Recommendations[4,5] include defining a continuous evaluation strategy, using containers (i.e. isolated software packages) to allow reproducibility and long-term availability of tools[6], reporting computational resources consumption (available hardware limits the choice to otherwise less efficient methods), exploring parameters, and avoiding rankings based on a unique metric. In the particular case of taxonomic classification, a benchmark that uses the same reference for all methods is necessary to perform a valid evaluation of their respective algorithms.

Given the lack of objectivity found in individual method publications, several groups have published comparative studies[7], including the Critical Assessment of Metagenome Interpretation[8] (CAMI), which conducted a challenge over several months leading to a collaborative publication, and introduced valuable methodologies related to benchmarking[9–11]. While necessary to recognize and guide innovation in the development of methods, such publications are ephemeral and do not help identify the most up-to-date methods in a fast-evolving field. As shown in Supplementary Table 1, positive evaluations by CAMI have not greatly impacted the adoption of methods in the year following its publication. To overcome these limitations, we introduce LEMMI (Figure 1), a web-based platform that hosts a semi-automated benchmarking pipeline for taxonomic profilers and binners. It explicitly addresses key problems for the community: (i) closing the time gap between benchmarking publications by continuously evaluating new methods, (ii) allowing heterogeneous tools and pipelines to be evaluated together with a multi-objective exploration and ranking of their performances, (iii) reporting computational resources, (iv) exploring parameters and references, (v) facilitating the dissemination of easy-to-use software packages, and (v) producing evaluations in a neutral and controlled environment to ensure published methods have a reliable benchmark.

LEMMI uses a containerized approach resembling a previously suggested format, bioboxes[9]. LEMMI is the first solution in the field to fully embrace the advantages of containerization to manage a benchmarking workflow (Supplementary Figure 1a) and to guarantee stable access to evaluated tools. Making informed choices when designing analyses or pipelines requires a thorough exploration of the parameter space of available algorithms. Multiple runs of a container enable such exploration. LEMMI does not segregate methods into profilers and binners, it is “results-oriented” so comparisons can be made between multi-functional tools as well as their combinations into pipelines to generate the two kinds of output. Therefore, these two-dimensional results can be visualized from different perspectives through a dynamic multi-criteria ranking (Figure 2-3, Supplementary Figure 2).

Some method implementations provide pre-packaged reference databases while others provide scripts to generate them. LEMMI evaluates these pre-packaged references to include methods whose value lies in providing a curated database (e.g. marker genes[2]) and to keep track of reference genome catalogs available to method users. However, in the LEMMI context, the scripts for producing a reference database starting from custom sequences allow the algorithm underlying the method to be evaluated independently from the taxa that constitute its default reference database. The use of different reference databases is likely a major source of result discrepancies (Supplementary Note 1). Consequently, methods considering any nucleotide or protein files have to make these scripts available in the container. As for parameters, this enables the exploration of references, for instance by restraining the source material to what was available at a given date or to specific metadata (e.g. assembly states). The present release of LEMMI embeds all bacterial and archaeal RefSeq[12] assembly entries having both nucleotide and corresponding protein files to offer both types of sequences as a possible reference. Not every method can process the 125,000 genomes included in the full LEMMI/RefSeq repository with the resources provided (245 GB of RAM). It is therefore relevant both to assess which subsets constitute good tradeoffs in terms of runtime, memory use, and accuracy of the predictions, and to track the impact of continuous database growth on different methods[13]. LEMMI uses its repository as source material provided to each method to create their reference database but also for sampling mock microbial communities to generate in-silico paired-end short reads used to measure the accuracy of predictions made by each method. It implements a genome exclusion approach to simulate unknown organisms at various taxonomic ranks and to prevent overfitting by excluding the source of the analyzed reads from the reference (Supplementary Figure 3, see Online Methods).

Datasets generated in-house by LEMMI contain randomly sampled bacteria and archaea, the most widely explored domains in metagenome research, representing communities of variable complexity in terms of number of species, abundance distribution, and k-mer content (Supplementary Table 2). They are complemented by datasets used in previously published benchmarking studies (CAMI, mockrobiota[14]). This is essential for comparisons with past results and to identify benchmark-specific biases in the dataset creation (Supplementary Note 2). The content of these external datasets is not under the control of the platform, however, and genome exclusion is not possible. The LEMMI/RefSeq repository may therefore constitute an overfitted reference, as is the case for the pre-packaged references regarding all datasets. Consequently, LEMMI offers two benchmarking categories for different needs (Figure 2). When referring to “TOOLS & REFERENCES”, users can get an overview of all tools and references that can be combined, giving an outline of the ability of currently available approaches to capture the known microbial diversity. The second category, “METHOD ALGORITHMS”, considers only methods and datasets that permit the creation of a reference, using genome exclusion, that will be identical for each run. This guarantees a fair and trusted benchmark for developers to support their algorithm improvement claims. The score assigned to each entry visible in LEMMI rankings is averaged over all tested datasets to summarize their performances. In addition, detailed values of 17 metrics are plotted individually for each dataset and taxonomic rank, namely genus and species (Figure 3, Supplementary Table 3).

LEMMI envisions a sustainable life cycle (Supplementary Figure 1b), taking advantage of contributions and feedback from the community of method developers and users to evolve and adapt to new expectations in a fast-moving field. It ensures a traceable archive of past rankings through the use of unique “fingerprints” to be reported in publications. The current release of LEMMI uses the NCBI taxonomy[15] but is ready to integrate alternative taxonomies, as there are calls to revise NCBI[16] and method implementations should be agnostic regarding taxonomic authority (i.e. allowing a variable number of ranks and identifiers) to explore new approaches meant to improve the classification resolution (e.g. reaching bacterial strains, viral operational taxonomic units[17]). Renewal of LEMMI datasets (e.g. also including long read technologies) along with the release of their yet undisclosed composition will occur periodically. As a proof of concept, we populated LEMMI with a selection of established or promising methods (Supplementary Table 4). The platform welcomes updates and new contributions from developers on https://lemmi.ezlab.org, where documentation, support, a discussion board, and evaluated containers are available.

Benchmarking has become a must-have requirement for publishing novel methods. To bring credibility and facilitate the adoption by their target audience, it is essential that methods appear side by side with established competitors in a trusted independent ranking. The technology of containerization has a strong future in the bioinformatics community[9,18]. Therefore, LEMMI will encourage developers to consider biocontainers to disseminate their work and to standardize the results formats so users will obtain easy-to-use and stable implementations of up-to-date methods as they appear in the benchmark.

Online Methods

Structure of the pipeline

An in-house python3-based[19,20] controller coordinates the many subtasks required to generate datasets, run the candidate containers, and compute the statistics. Snakemake 5.3.1[21] 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[22]: https://mpld3.github.io.

LEMMI containers

The LEMMI containers are implemented for Docker 18.09.0-ce. They partially follow the design[9] introduced by http://bioboxes.org/ 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 method developer has to build the container on their own environment, while ensuring that both tasks can be run by an unprivileged user and return the desired outputs. A tutorial is available on https://gitlab.com/ezlab/lemmi/wikis/user-guide. The containers or the sources to recreate them are made available to the users.

Computing resources

During the benchmarking process, the container is loaded on a dedicated server and given 245 GB of RAM and 32 cores. Reaching the memory limit will cause the container to be killed ending the benchmarking process. All inputs and outputs are written on a local disk and the container is not given access to the Internet.

Taxonomy

The NCBI taxonomy is used to validate all entries throughout the process and unknown taxids are ignored (unclassified). The framework etetoolkit[23] (ETE3) 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 (download date for the current release was 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. The taxonomic lineage for the seven main levels was extracted with ETE3 (superkingdom, phylum, class, order, family, genus, species). Viruses were not used for generating references and datasets in this proof of concept study and the number of genomes in the reference RefSeq/08.2018/All is 124,289. To subset the repository and keep one representative per species as inputs for the reference construction (for entries labelled as RefSeq/08.2018/1rep.), the list of bacterial and archaeal genomes 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, for a total of 18,907 files. When subsetting the repository in the genome exclusion mode (i.e. for the METHOD ALGORITHMS category, Supplementary Figure 3), if the entry to be selected was part of the reads, the next representative in the list was used instead when available.

LEMMI datasets

To sample the genomes included in the LEMMI datasets, a custom python script was used to randomly select representative genomes in the RefSeq assembly repository, among bacterial and archaeal content (Supplementary Table 2). Their abundance was randomly defined following a lognormal distribution (mean=1, standard deviation in Supplementary Table 2). In the case of LEMMI_LOWDIV datasets, additional low coverage species (abundance corresponding to < 100 reads) were manually defined while in LEMMI_MEDDIV datasets, low coverage species were produced as part of the random sampling procedure. Each species abundance was normalized according to the species average genome size (as available in the LEMMI/RefSeq repository) to get closer to organisms abundance (considering one genome copy per cell), and the total was normalized to one to constitute a relative abundance profile. BEAR[24] was used to generate paired-end reads, 2x150 bp, and DRISEE[25] 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 taxonomic ranks, and taxonomic 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[26] 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/ (accessed 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[27] 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.

Analysis of the results

The profile and binning reports are processed with OPAL 0.2.8[10] and AMBER 0.7.0[11] against the ground truth to obtain a wide range of metrics. Binning reports are processed to obtain a file for each taxonomic rank (genus/species), moving reads up from lowest level. The profiles of the candidate methods and the ground truth are filtered to discard low coverage taxa at different thresholds (below values corresponding to 1/10/100/1000 reads) and all metrics are computed for all values. When a container is not able to provide a profile as output, the LEMMI platform generates one using the proportion of reported reads. Taxa detection metrics are based on OPAL and thus on the profile output. Methods reporting a profile with 0.0 for low abundance taxa despite being present in their binning files will shift the balance from recall to precision (Supplementary Note 1). The low abundance score takes into account both the profile and binning output and is a custom metric calculated separately to evaluate the ability of the method 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 single dataset (false positives always 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 ≠ T2_low, < 100 reads coverage,). Each taxon belonging to T1_low identified in D1 increases the low abundance score given to the method for D1 (recall) only when it is not identified in D2 if absent from T2. Otherwise, a correct prediction of the taxon in D1 is canceled and does not improve the score (acting as proxy for low abundance precision). This is illustrated with Supplementary Fig. 3. The score (0.0 - 1.0) is processed from both sides (D1, D2), to obtain an independent score for each of the paired dataset. This metric is only defined for the pair of LEMMI_LOWDIV and the pair of LEMMI_MEDDIV datasets (low abundance species: n=10, n=8, n=98, n=138 for LEMMI_LOWDIV_001, LEMMI_LOWDIV_002, LEMMI_MEDDIV_001, LEMMI_MEDDIV_002 respectively). The runtime corresponds to the time in minutes during which the container is loaded. The memory is the peak value of total_rss memory reported when the container is loaded to complete one task. Methods unable to deliver part of the expected outputs are assigned 0.0 for the corresponding metrics (e.g. methods unable to provide a read binning report).

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 LEMMI user through the interface) and subtracted from 1.0, to obtain a range between 0.5 and 1.0. This approach allows the user 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 set to be 0 and 1 respectively. Each value is calculated for genus and species at 1, 10, and 100 reads low coverage filtering level and used or ignored according to the choice of the LEMMI user regarding these parameters. The final score displayed in the ranking is the harmonic mean of all metrics, taken into account 0, 1, or 3 times depending on the weight assigned to the metric by the LEMMI user.

Acknowledgement

We would like to thank all members of the Zdobnov group for discussions and feedback. This work was supported by the Swiss National Science Foundation funding 31003A_166483 to E.Z.

Author contributions

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

Competing interests

The authors declare no competing interests.

Reference

1. Kim, D., Song, L., Breitwieser, F. P. & Salzberg, S. L. Centrifuge: rapid and sensitive classification of metagenomic sequences. *Genome Res.* **26**, 1721–1729 (2016).
2. Truong, D. T. *et al.* MetaPhlAn2 for enhanced metagenomic taxonomic profiling. *Nat. Methods* **12**, 902–903 (2015).
3. Jacobs, J. Microbe Land. https://microbe.land/2018/12/13/97-metagenomics-classifiers/ Accessed 15/04/2019.
4. Mangul, S. *et al.* Systematic benchmarking of omics computational tools. *Nat. Commun.* **10**, (2019).
5. Maier-Hein, L. *et al.* Why rankings of biomedical image analysis competitions should be interpreted with care. *Nat. Commun.* **9**, (2018).
6. Mangul, S., Martin, L. S., Eskin, E. & Blekhman, R. Improving the usability and archival stability of bioinformatics software. *Genome Biol.* **20**, (2019).
7. Gardner, P. P. *et al.* Identifying accurate metagenome and amplicon software via a meta-analysis of sequence to taxonomy benchmarking studies. *PeerJ* **7**, e6160 (2019).
8. Sczyrba, A. *et al.* Critical Assessment of Metagenome Interpretation—a benchmark of metagenomics software. *Nat. Methods* **14**, 1063–1071 (2017).
9. Belmann, P. *et al.* Bioboxes: standardised containers for interchangeable bioinformatics software. *GigaScience* **4**, (2015).
10. Meyer, F. *et al.* Assessing taxonomic metagenome profilers with OPAL. *Genome Biol.* **20**, (2019).
11. Meyer, F. *et al.* AMBER: Assessment of Metagenome BinnERs. *GigaScience* **7**, (2018).
12. O'Leary, N. A. *et al.* Reference sequence (RefSeq) database at NCBI: current status, taxonomic expansion, and functional annotation. *Nucleic Acids Res.* **44**, D733–D745 (2016).
13. Nasko, D. J., Koren, S., Phillippy, A. M. & Treangen, T. J. RefSeq database growth influences the accuracy of k-mer-based lowest common ancestor species identification. *Genome Biol.* **19**, (2018).

14. Bokulich, N. A. *et al.* mockrobiota: a Public Resource for Microbiome Bioinformatics Benchmarking. *mSystems* **1**, (2016).

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

16. Parks, D. H. *et al.* A standardized bacterial taxonomy based on genome phylogeny substantially revises the tree of life. *Nat. Biotechnol.* **36**, 996–1004 (2018).

17. Paez-Espino, D. *et al.* IMG/VR v.2.0: an integrated data management and analysis system for cultivated and environmental viral genomes. *Nucleic Acids Res.* **47**, D678–D686 (2019).

18. da Veiga Leprevost, F. *et al.* BioContainers: an open-source and community-driven framework for software standardization. *Bioinformatics* **33**, 2580–2582 (2017).

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

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

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

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

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

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

25. Keegan, K. P. *et al.* A Platform-Independent Method for Detecting Errors in Metagenomic Sequencing Data: DRISEE. *PLoS Comput. Biol.* **8**, e1002541 (2012).

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

27. Kozich, J. J., Westcott, S. L., Baxter, N. T., Highlander, S. K. & Schloss, P. D. 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 (2013).

Figures

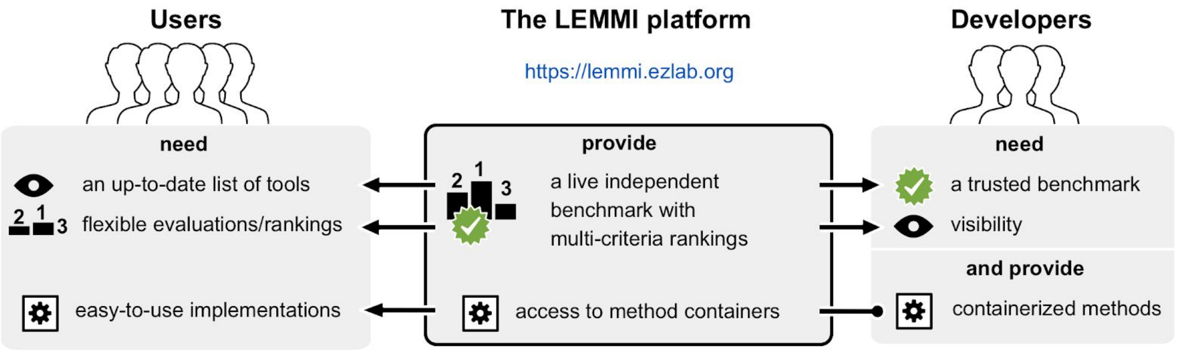
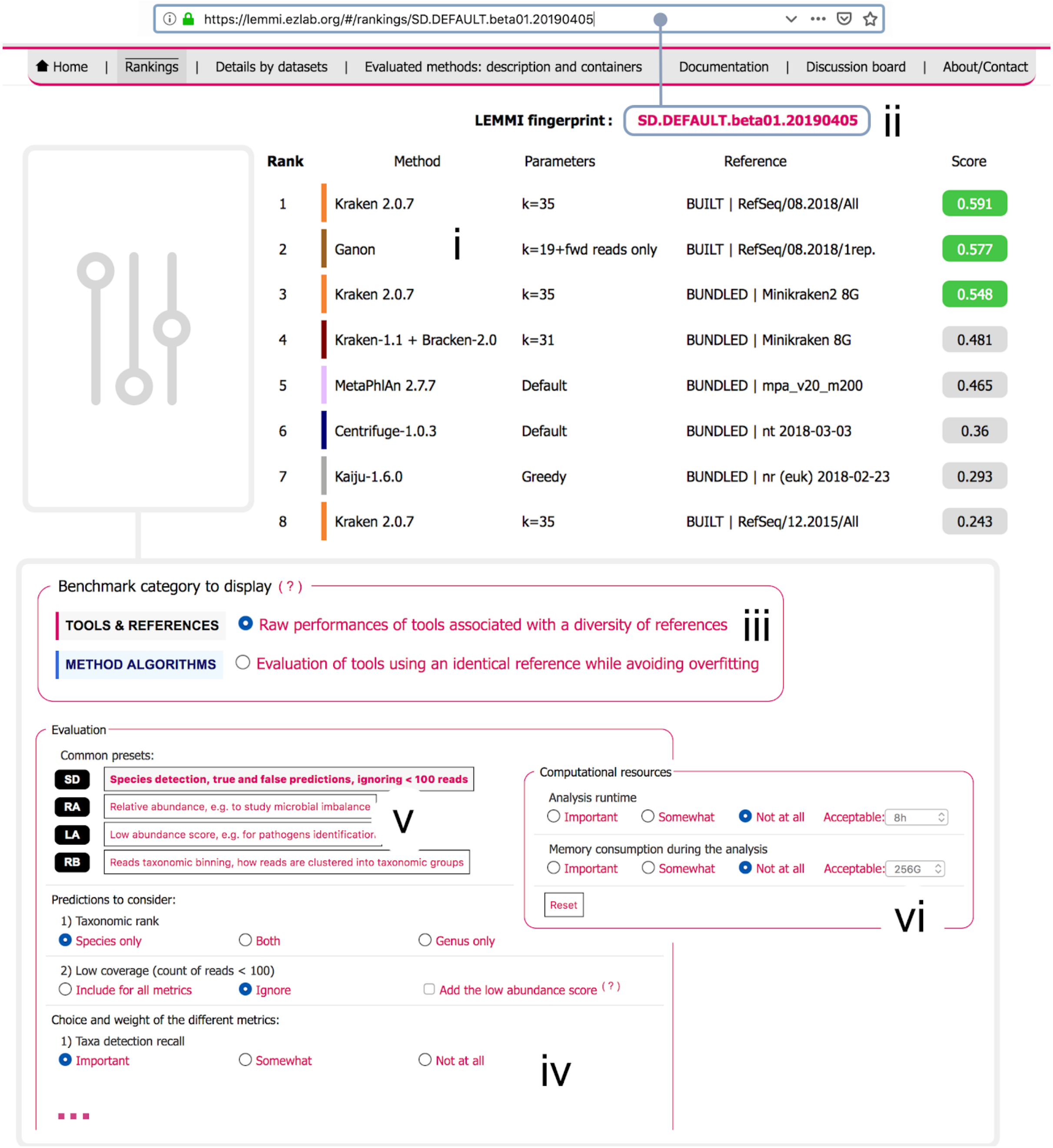

Users
The LEMMI platform
https://lemmi.ezlab.org
Developers
need
an up-to-date list of tools
2 1 3
flexible evaluations/rankings
easy-to-use implementations
provide
2 1 3
a live independent benchmark with multi-criteria rankings
access to method containers
need
a trusted benchmark
visibility
and provide
containerized methods

Fig. 1 | The LEMMI platform facilitates the access to up-to-date implementations of methods for taxonomic binning and profiling. LEMMI creates a link between developers and users as it provides independent and unbiased benchmarks that are valuable to both. Developers need comparative evaluations to keep improving the methodology and to get their work published in peer-reviewed journals. Method users need a resource that keeps track of new developments and provides a flexible assessment of their performances to match their experimental goals, resources, and expectations. The containerized approach of LEMMI guarantees the transfer of a stable and usable implementation of each method from the developer to the final user, as they appear in the evaluation.


https://lemmi.ezlab.org/#/rankings/SD.DEFAULT.beta01.20190405
Home | Rankings | Details by datasets | Evaluated methods: description and containers | Documentation | Discussion board | About/Contact
i
ii
iii
iv
v
vi

LEMMI fingerprint : SD.DEFAULT.beta01.20190405

Rank	Method	Parameters	Reference	Score
1	Kraken 2.0.7	k=35	BUILT | RefSeq/08.2018/All	0.591
2	Ganon	k=19+fwd reads only	BUILT | RefSeq/08.2018/1rep.	0.577
3	Kraken 2.0.7	k=35	BUNDLED | Minikraken2 8G	0.548
4	Kraken-1.1 + Bracken-2.0	k=31	BUNDLED | Minikraken 8G	0.481
5	MetaPhlAn 2.7.7	Default	BUNDLED | mpa_v20_m200	0.465
6	Centrifuge-1.0.3	Default	BUNDLED | nt 2018-03-03	0.36
7	Kaiju-1.6.0	Greedy	BUNDLED | nr (euk) 2018-02-23	0.293
8	Kraken 2.0.7	k=35	BUILT | RefSeq/12.2015/All	0.243

Benchmark category to display (?)

TOOLS & REFERENCES ● Raw performances of tools associated with a diversity of references

METHOD ALGORITHMS ○ Evaluation of tools using an identical reference while avoiding overfitting

Evaluation

Common presets:

SD Species detection, true and false predictions, ignoring < 100 reads

RA Relative abundance, e.g. to study microbial imbalance

LA Low abundance score, e.g. for pathogens identification

RB Reads taxonomic binning, how reads are clustered into taxonomic groups

Predictions to consider:

1) Taxonomic rank
● Species only ○ Both ○ Genus only

2) Low coverage (count of reads < 100)
○ Include for all metrics ● Ignore □ Add the low abundance score (?)

Choice and weight of the different metrics:

1) Taxa detection recall
● Important ○ Somewhat ○ Not at all

...

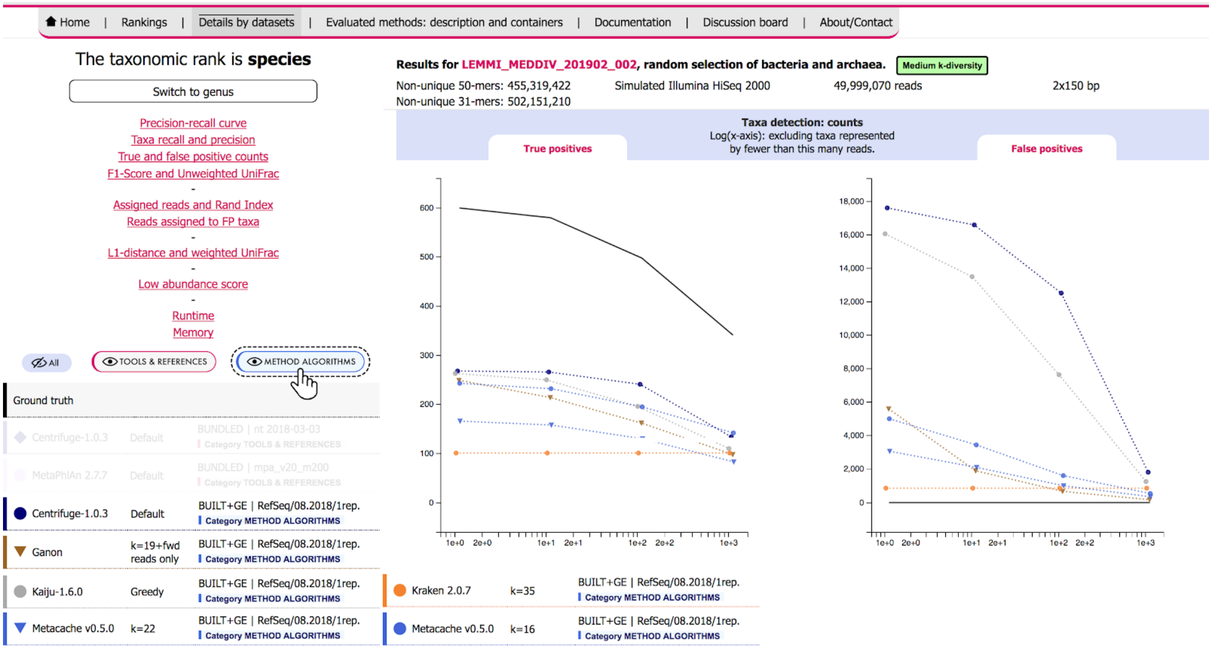
Computational resources

Analysis runtime
○ Important ○ Somewhat ● Not at all Acceptable: 8h

Memory consumption during the analysis
○ Important ○ Somewhat ● Not at all Acceptable: 256G

Reset

Fig. 2 | LEMMI web interface. (i) The LEMMI users obtain a list of entries suited to their needs through the dynamic ranking interface that allows them to select and weight the criteria that are important. (ii) A unique fingerprint allows custom rankings to be shared and restored at any time on the LEMMI platform through the address bar. (iii) The benchmark category can be selected in the dashboard. (iv) Prediction accuracy metrics can be chosen along with their importance (Weight for “Important” is 3, “Somewhat” is 1, and “Not at all” is 0). This will cause the list to be updated with a score that represents a weighted average of all selected metrics. (v) Several presets corresponding to common expectations are available. (vi) Computational resources and the time required to complete the analysis can be included as an additional factor to rank the methods.


Home | Rankings | Details by datasets | Evaluated methods: description and containers | Documentation | Discussion board | About/Contact
Switch to genus
Precision-recall curve
Taxa recall and precision
True and false positive counts
F1-Score and Unweighted UniFrac
Assigned reads and Rand Index
Reads assigned to FP taxa
L1-distance and weighted UniFrac
Low abundance score
Runtime
Memory
All
TOOLS & REFERENCES
METHOD ALGORITHMS
Medium k-diversity
True positives
False positives
600
500
400
300
200
100
0
18,000
16,000
14,000
12,000
10,000
8,000
6,000
4,000
2,000
0
1e+0
2e+0
1e+1
2e+1
1e+2
2e+2
1e+3

The taxonomic rank is **species**

Results for LEMMI_MEDDIV_201902_002, random selection of bacteria and archaea.

Non-unique 50-mers: 455,319,422 Simulated Illumina HiSeq 2000 49,999,070 reads 2x150 bp
Non-unique 31-mers: 502,151,210

Taxa detection: counts
Log(x-axis): excluding taxa represented by fewer than this many reads.

Ground truth			
◆	Centrifuge-1.0.3	Default	BUNDLED | nt 2018-03-03 Category TOOLS & REFERENCES
●	MetaPhlAn 2.7.7	Default	BUNDLED | mpa_v20_m200 Category TOOLS & REFERENCES
●	Centrifuge-1.0.3	Default	BUILT+GE | RefSeq/08.2018/1rep. Category METHOD ALGORITHMS
▼	Ganon	k=19+fwd reads only	BUILT+GE | RefSeq/08.2018/1rep. Category METHOD ALGORITHMS
●	Kaiju-1.6.0	Greedy	BUILT+GE | RefSeq/08.2018/1rep. Category METHOD ALGORITHMS
▼	Metacache v0.5.0	k=22	BUILT+GE | RefSeq/08.2018/1rep. Category METHOD ALGORITHMS
●	Kraken 2.0.7	k=35	BUILT+GE | RefSeq/08.2018/1rep. Category METHOD ALGORITHMS
●	Metacache v0.5.0	k=16	BUILT+GE | RefSeq/08.2018/1rep. Category METHOD ALGORITHMS

Fig. 3 | “Details by datasets” interface. Plots for 17 metrics are available and the LEMMI user can toggle each line representing a method associated with a reference and specific parameters individually. This is exemplified here by two plots showing the true and false positive counts of taxa detection as a function of filtering the reads below a given threshold. It is also possible to display either all entries belonging to the category “TOOLS & REFERENCES” or all entries belonging to the category “METHOD ALGORITHMS” (as indicated by the cursor). Plots can be zoomed in to disentangle overlapping points and permit focused interpretations. The pages exist for all taxonomic ranks investigated, i.e. genus and species in the current release.