LEMMI: A continuous benchmarking platform for metagenomics classifiers

Mathieu Seppey[1] ORCID: 0000-0003-3248-011X
Mosè Manni[1] ORCID: 0000-0002-4146-6523
Evgeny M. Zdobnov[1]* ORCID: 0000-0002-5178-1498

[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

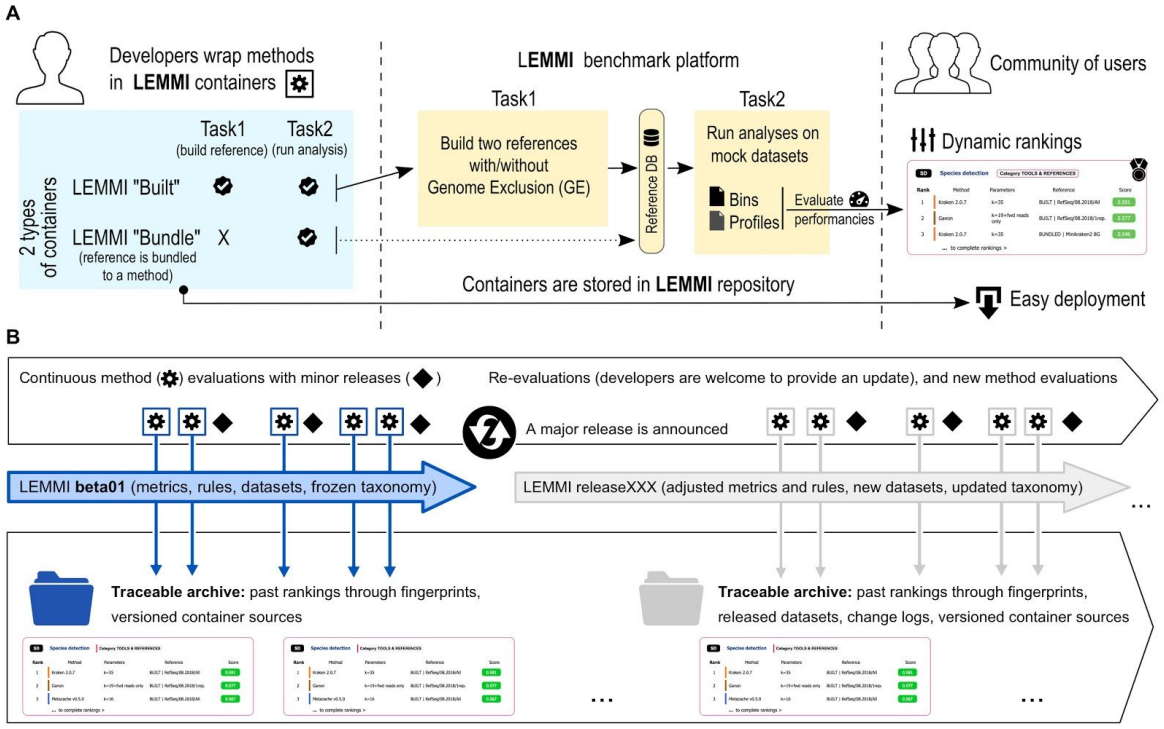
Supplemental Note, Figures, and Tables

Supplemental Note S1

Assembly state is not an efficient criterion for subsampling RefSeq, for benchmarking or analysis purposes.

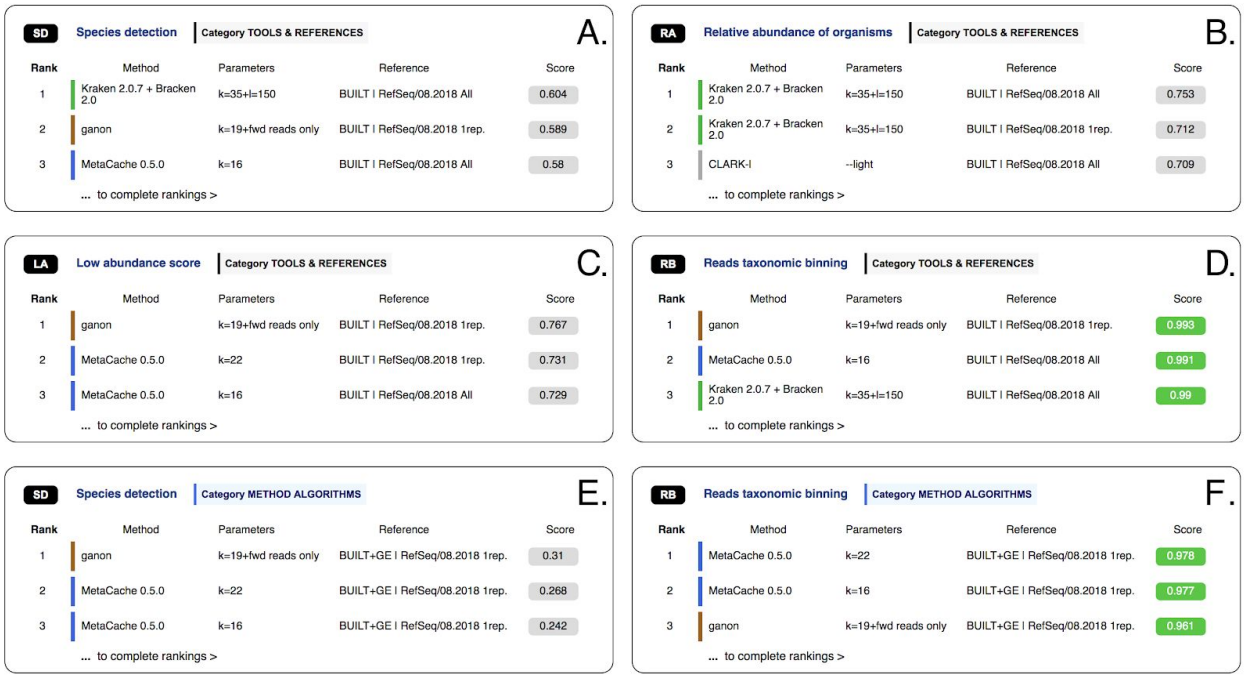
Assemblies deposited on the RefSeq repository are flagged with “Complete genome”, “Chromosome”, “Scaffold”, or “Contig” to describe the completion state of sequencing and assembly procedures. It is likely that the accelerating production of new assemblies, enabled notably by metagenomics, leads to fewer new entries being polished enough to obtain the complete genomes flag, decreasing the representativity of this category over time (41% of the species taxid in the LEMMI/RefSeq repository in mid-2018, only 19% when ignoring viruses, which are not used in the beta01 release). The first LEMMI datasets (LEMMI_LOWDIV and LEMMI_MEDDIV) were created using only complete genomes sequences to work with the best representative sequences (Supplemental Table S2). We noticed that the Minikraken databases (obtained in October 2017, updated in November 2018 and April 2019 for Kraken 2) performed well on these, while being unable to recover most of the species in the CAMI1 datasets, in contrast to using Kraken 2 with all LEMMI/RefSeq genomes available in mid-2018, confirming that species found in CAMI1 datasets are represented in the RefSeq assembly repository (Supplemental Fig. 8). We realized that regular versions of Kraken databases are built using only sequences with the complete genomes state (Wood and Salzberg 2014). Eventually, LEMMI_HIGHDIV_201802_001 (Supplemental Table S2) was designed to account for the whole diversity found in RefSeq by sampling all assembly states. Overall, any reference based on complete genomes, such as Minikraken, will obtain a medium score in the current release of LEMMI, performing very well on LEMMI_LOWDIV and LEMMI_MEDDIV and very poorly on CAMI1 datasets. To mitigate the bias towards “complete genome” assemblies, all future LEMMI datasets will be created by sampling evenly across all available taxonomic identifiers regardless of the assembly status.

Supplemental Figure S1



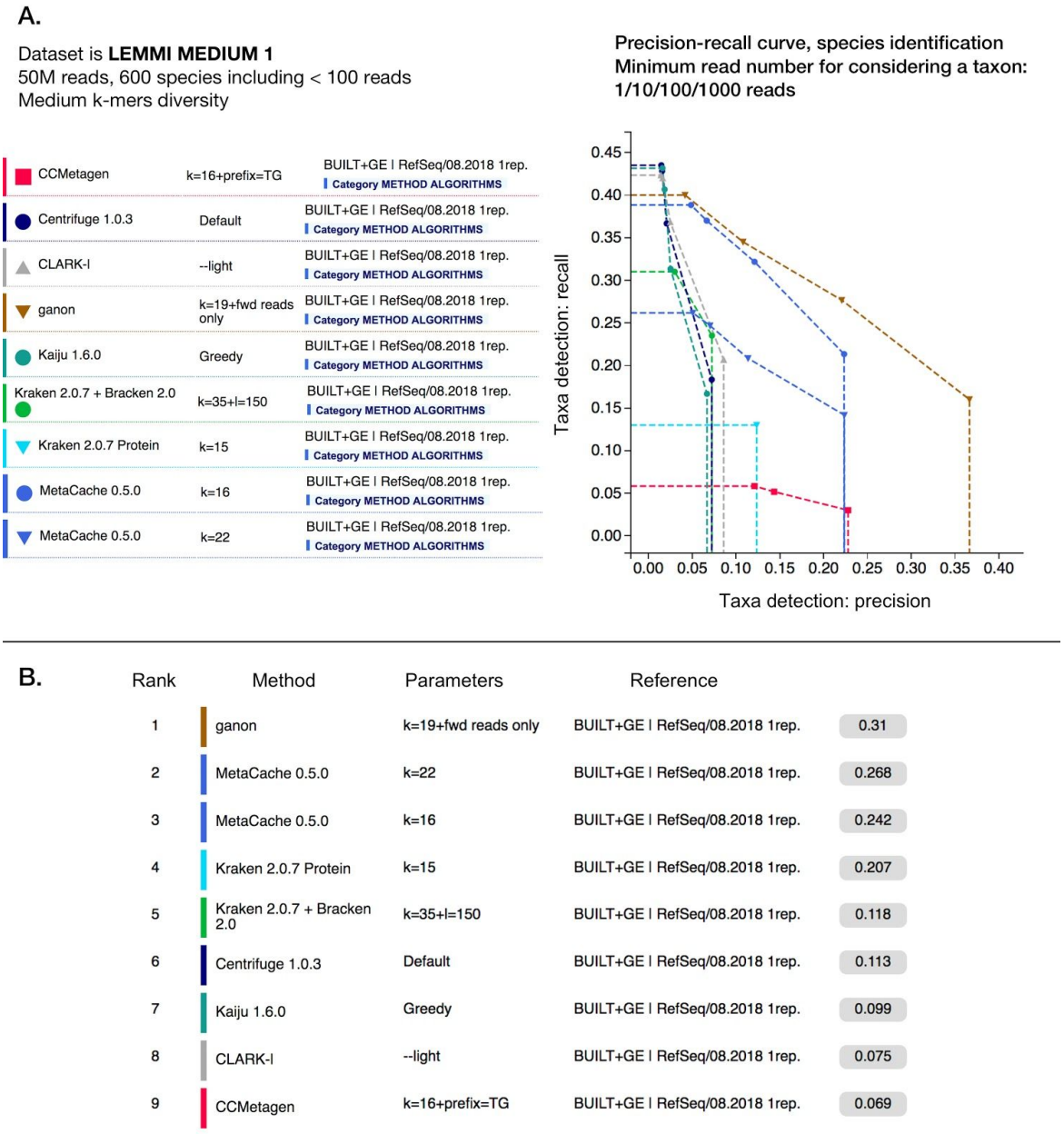
Supplemental Fig. S1 | LEMMI workflow and sustainable life cycle. (A) Developers prepare a container for their method following the provided guideline, to complete two tasks: building a reference using provided FASTA files (first task), and analyzing FASTQ samples to return a profile and binned reads (second task). Developers can also suggest a pre-packaged “bundled” reference instead of performing the first task. Their containerized method is then managed by the LEMMI administrator to be run within the LEMMI platform to process all datasets required to appear in the ranking. Multiple runs to explore parameters and references can be conducted using a single container. Method users can browse the results to define which methods best suit their needs and obtain the corresponding containers to conduct their own tests or actual analyses, with the guarantee of unified file formats and similar behaviors. (B) The release beta01 of LEMMI is the first major release. Every successful evaluation is integrated into the rankings, which are traceable through time and for which the source of the container is publicly available. Feedback from the community and progress in the field will eventually lead to the end of this first release to allow an update of both the platform and the datasets. While entering the next major release, still relevant methods will be systematically re-evaluated and new submissions will continue to populate the rankings.

Supplemental Figure S2



Supplemental Fig. S2 | The LEMMI homepage. The main page presents multiple rankings, each corresponding to the top three configurations (methods associated with a reference and specific parameters) according to metrics chosen for addressing various experimental objectives (A-F) in one of the benchmark categories (TOOLS & REFERENCES and METHOD ALGORITHMS). These lists constitute different entry points to the dynamic ranking page where the full repertoire of configurations can be explored beyond these predefined criteria.

Supplemental Figure S3



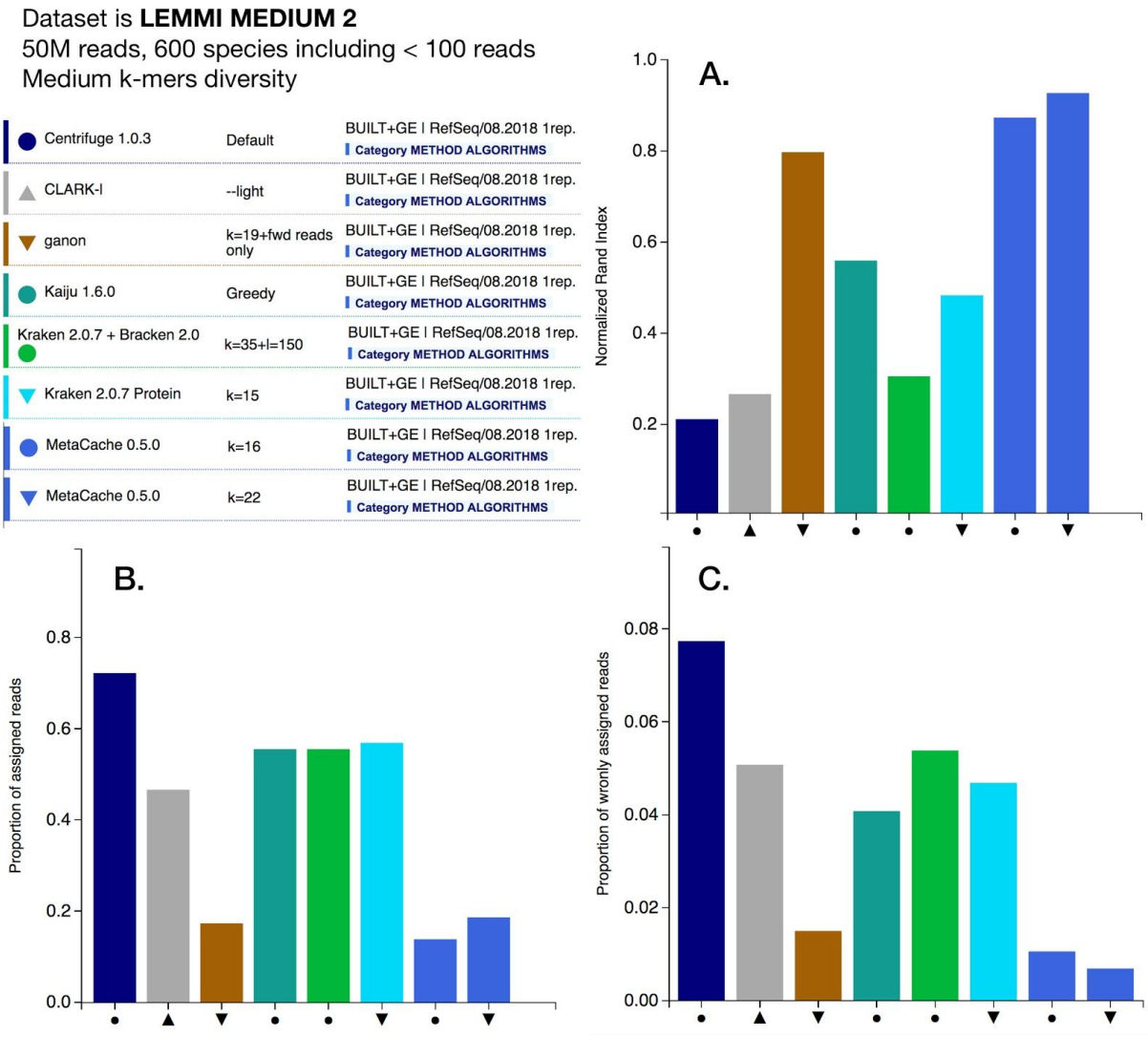
B.

Rank	Method	Parameters	Reference	
1	ganon	k=19+fwd reads only	BUILT+GE | RefSeq/08.2018 1rep.	0.31
2	MetaCache 0.5.0	k=22	BUILT+GE | RefSeq/08.2018 1rep.	0.268
3	MetaCache 0.5.0	k=16	BUILT+GE | RefSeq/08.2018 1rep.	0.242
4	Kraken 2.0.7 Protein	k=15	BUILT+GE | RefSeq/08.2018 1rep.	0.207
5	Kraken 2.0.7 + Bracken 2.0	k=35+l=150	BUILT+GE | RefSeq/08.2018 1rep.	0.118
6	Centrifuge 1.0.3	Default	BUILT+GE | RefSeq/08.2018 1rep.	0.113
7	Kaiju 1.6.0	Greedy	BUILT+GE | RefSeq/08.2018 1rep.	0.099
8	CLARK-l	--light	BUILT+GE | RefSeq/08.2018 1rep.	0.075
9	CCMetagen	k=16+prefix=TG	BUILT+GE | RefSeq/08.2018 1rep.	0.069

Supplemental Fig. S3 | Evaluation using an identical reference. (A) Precision-recall curve in species identification of a mix of methods using identical references built using one representative genome per species taxid, excluding the source of the reads (curves with less than four data points indicate that filtering did not affect precision and recall at all thresholds, thus overlapping). Not all species can be recovered as their only representative was used to produce reads. Therefore, the best recalls illustrated here are close to the maximum that can

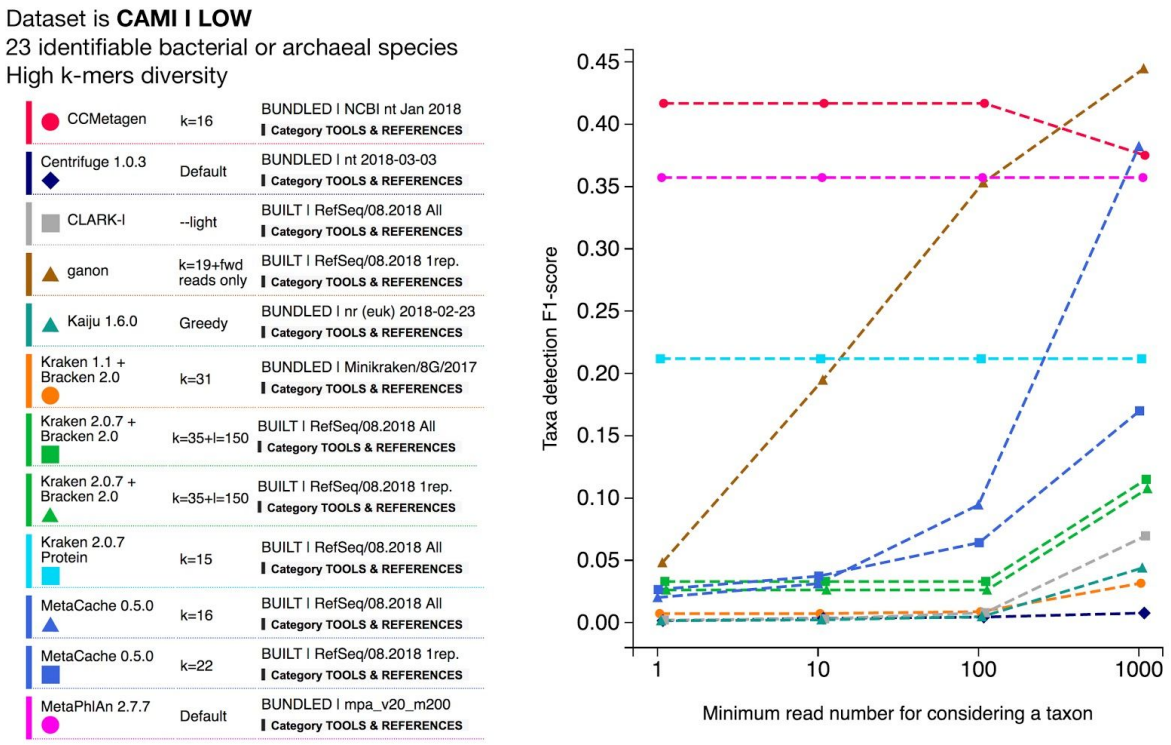
be reached at the species level under this scenario. (B) Ranking of methods over all datasets under the scenario described above given the LEMMI preset “Species detection”. Precision and recall are considered equally and taxa represented by less than 100 reads are ignored.

Supplemental Figure S4



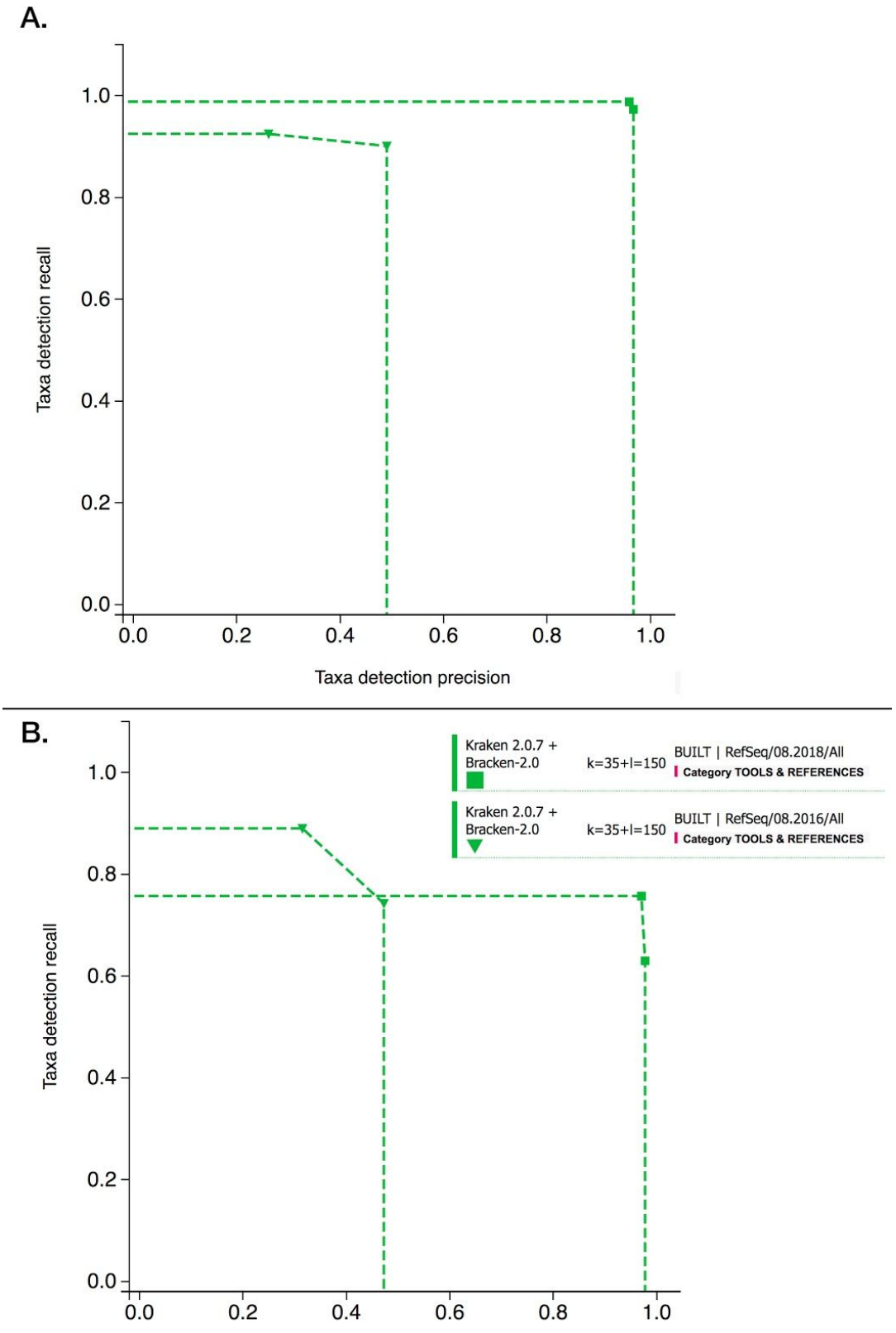
Supplemental Fig. S4 | Taxonomic binning. (A) Normalized Rand Index in species binning for a mix of methods using identical references built using one representative genome per species taxid, excluding the source of the reads. (B) Proportion of classified reads at the species level. (C) Proportion of reads assigned to a false positive species.

Supplemental Figure S5



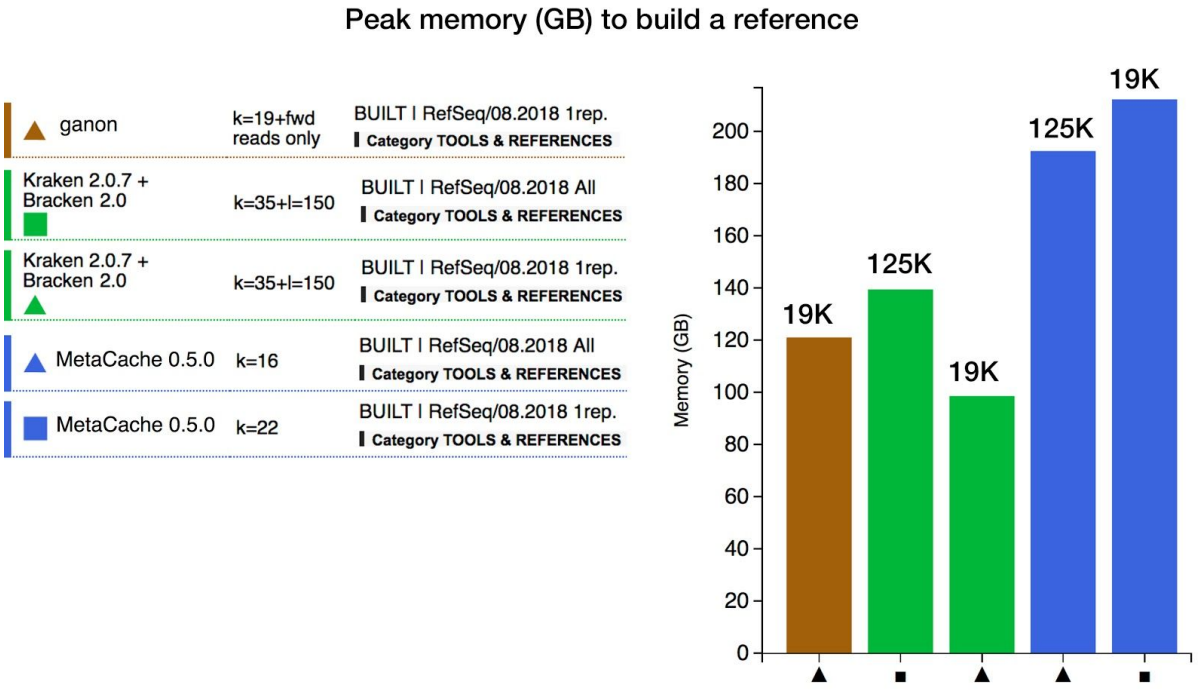
Supplemental Fig. S5 | A dataset with a low number of species. F1-score in species identification of a mix of methods using freely provided references or built using the maximal capacity of the tool with 245GB. The dataset is the low complexity set from the first CAMI challenge.

Supplemental Figure S6



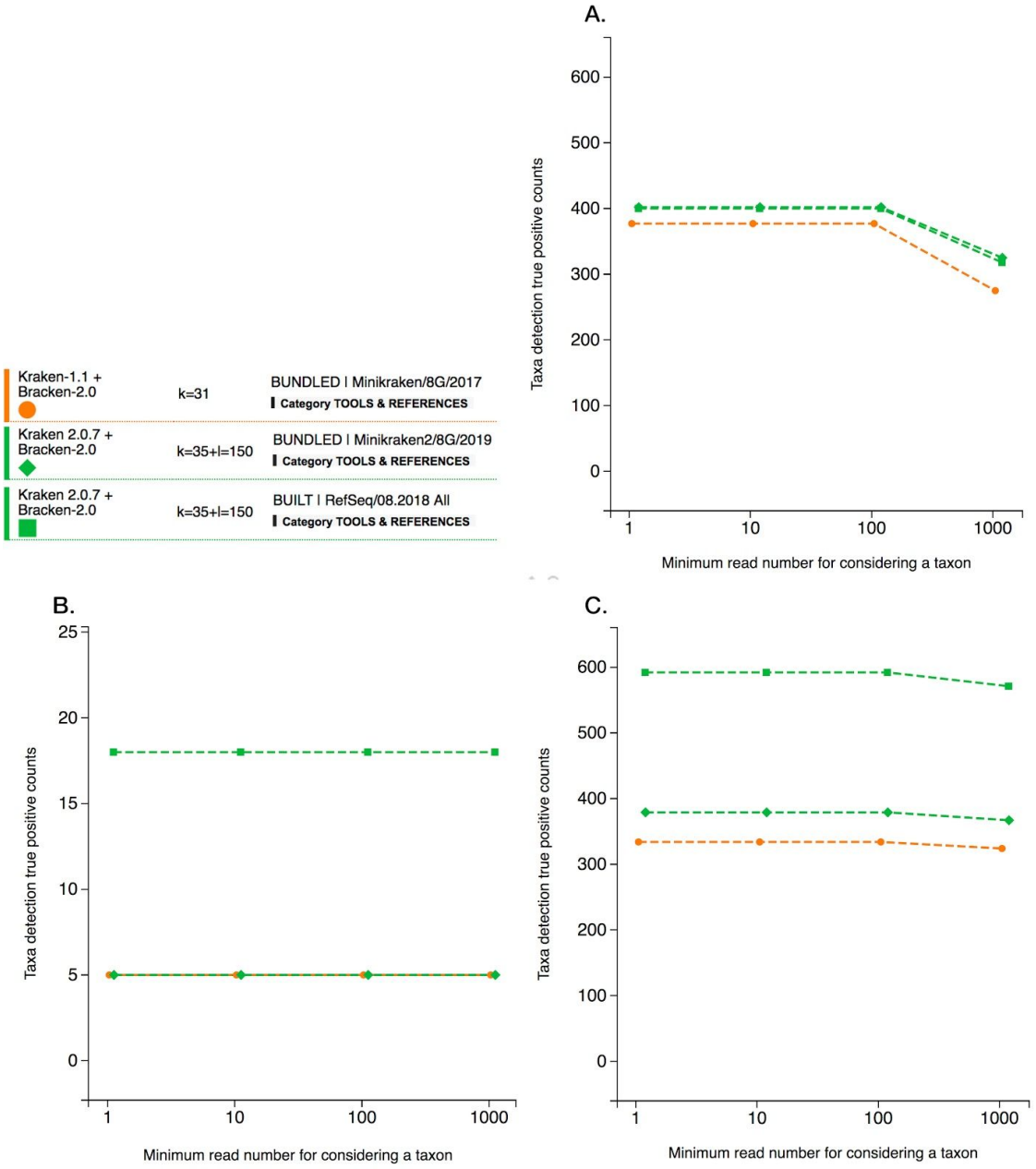
Supplemental Fig. S6 | Using LEMMI as a reference time machine. Kraken 2 analyses using the complete archaeal and bacterial content of the LEMMI repository from mid-2018, versus its state two years earlier. These two years doubled the number of genomes available as references. The taxonomic rank presented here is genus. An increase in the database size is not always beneficial in terms of recall, as reported previously (Nasko et al. 2018), but is beneficial in terms of precision. Overall, selecting the most up to date reference remains necessary to cover newly sampled taxa, as reflected by the ranking on main Figure 2. (A) Precision-recall curve in genus identification for the dataset LEMMI HIGH 1 (B) Precision-recall curve in genus identification for the dataset LEMMI MEDIUM 1.

Supplemental Figure S7



Supplemental Fig. S7 | Resources usage. Amount of memory used by ganon, Kraken 2, and MetaCache to construct their reference (other entries present in the LEMMI release beta01.20191118 are not shown). The number of genomes included, one representative per species (~19,000 files) or all representatives (~125,000 files) is indicated. MetaCache allocates the memory differently when using k=16 and k=22. It was not able to build the large reference with k=22, neither was ganon.

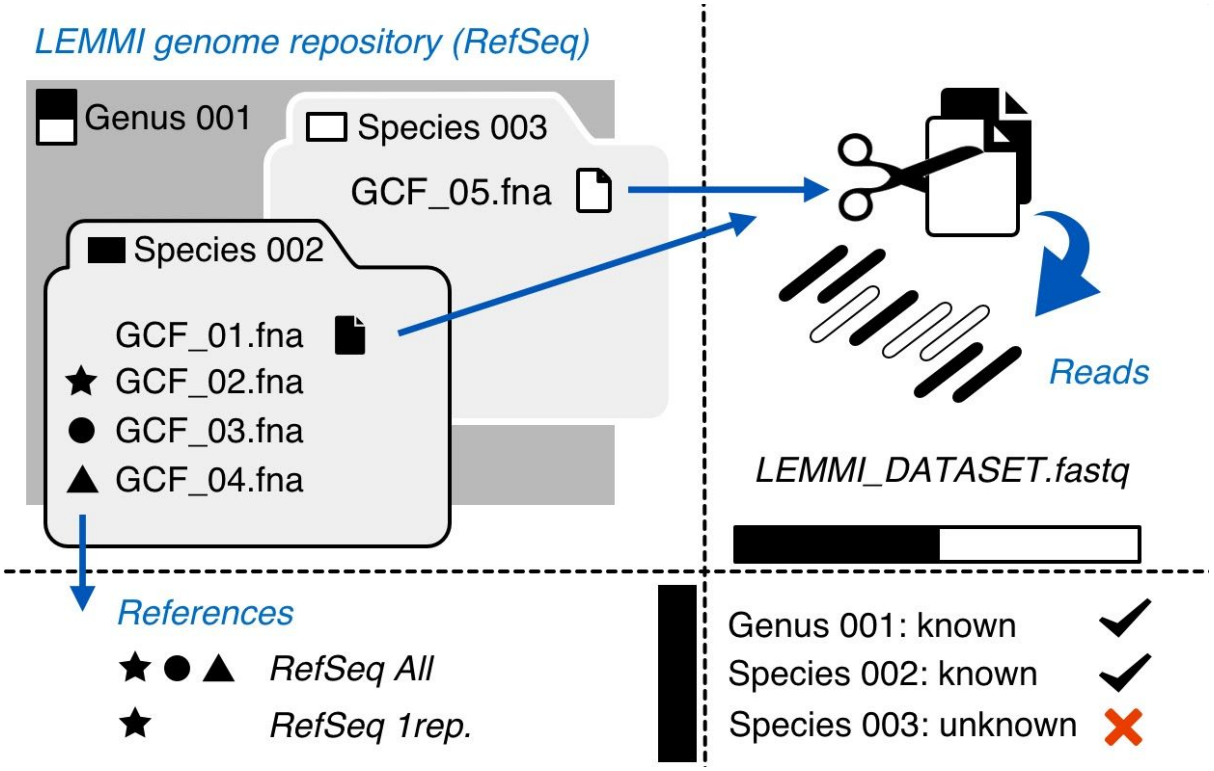
Supplemental Figure S8



Supplemental Fig. S8 | How the evaluation of the Minikraken database is biased by the design of the datasets used. Counts of species correctly identified when running Kraken and Kraken 2 with either Minikraken 2017/2019 or a comprehensive build of the LEMMI/RefSeq repository (mid-2018) without excluding any genome. (a) The dataset is

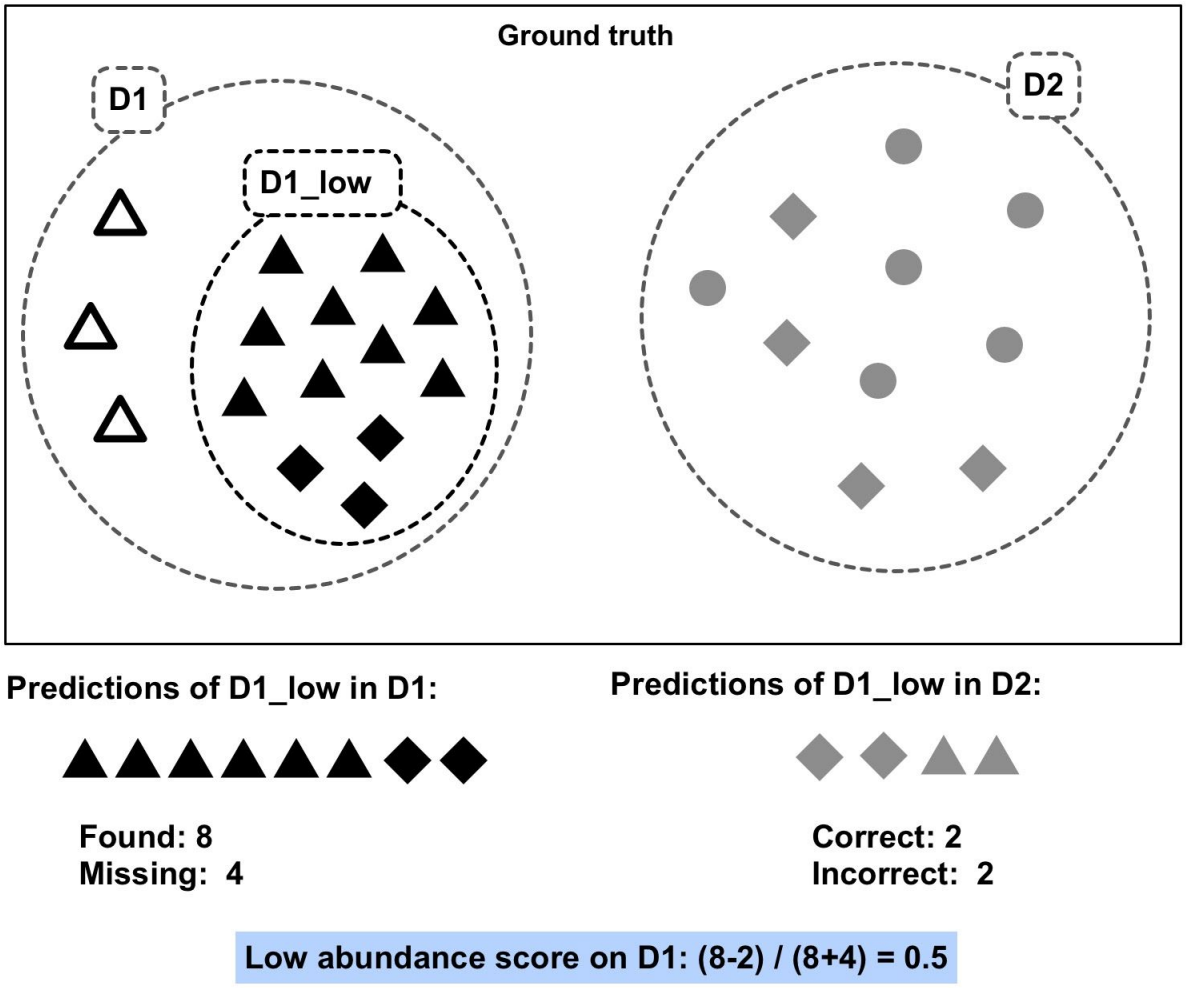
LEMMI MEDIUM 1, based only on assemblies flagged as “Complete Genome”. (b) The dataset is CAMI1 LOW. (c) The dataset is LEMMI HIGH 1, sampled using all assembly states in the LEMMI/RefSeq repository. The latter can be seen as the fairest evaluation of the Minikraken database.

Supplemental Figure S9



Supplemental Fig. S9 | Creating unknown taxa from public data. A toy example illustrating the "genome exclusion" approach used on LEMMI in-house datasets to avoid overfitting (i.e. having the source of the reads in the reference). Species 002 (black) and 003 (white) belonging to genus 001 have four representatives and one representative, respectively. One genome is taken from each species to simulate the reads, and the rest is used to build a comprehensive reference (RefSeq All) or a reduced one (RefSeq 1rep.) using one randomly selected representative per species. All candidate methods provided with this scenario are expected to identify the genus 001 and the species 002. Species 003 becomes an unknown species, with no sequence available as reference.

Supplemental Figure S10



Supplemental Fig. S10 | Low abundance score calculation using two datasets. The evaluated method returns the species predictions for the datasets D1 and D2. To score the accuracy of the low abundance predictions for the dataset D1, species that are present in low abundance (< 100 reads) in D1 (D1_low, black triangles and diamonds) increase the score when found. However, this is canceled if these species are falsely predicted in D2 (grey triangles). Species that are shared (grey diamonds), unique to D2 (grey circles), and species present in larger abundance in D1 (white triangles) are not considered to calculate this score. To score the accuracy of the low abundance predictions for the dataset D2, the process is reversed (D2_low not illustrated for simplicity).

Supplemental Table S1

Supplemental Table S1 | List of configurations (methods associated with a reference and specific parameters) included in release beta01.20191118 of the LEMMI platform. The configurations that successfully built a reference based on the entire LEMMI/RefSeq repository when provided with 245 GB of RAM are underlined. All others were limited to using one representative per species taxid. The source of corresponding containers can be found on https://gitlab.com/ezlab/lemmi/tree/beta01.20191118/containers

Method	Parameters	Reference	Benchmark category	Note
Kaiju 1.6.0	default	nr_euk 2018-02-23, bundled	TOOLS & REF.	
Centrifuge 1.0.3	default	nt 2018-03-03, bundled	TOOLS & REF.	
Kraken 1.1 + Bracken 2.0	default	Minikraken 8G 2017, bundled	TOOLS & REF.	125-mers db for bracken
Kraken 2.0.7 + Bracken 2.0	k=35	Minikraken 8G 2019, bundled	TOOLS & REF.	150-mers db for bracken
Kraken 2.0.7 + Bracken 2.0	k=35	RefSeq/08.2018/All, built	TOOLS & REF.	150-mers db for bracken
Kraken 2.0.7 + Bracken 2.0	k=35	RefSeq/08.2018/1rep., built	TOOLS & REF.	150-mers db for bracken
Kraken 2.0.7 + Bracken 2.0	k=35	RefSeq/08.2016/All, built	TOOLS & REF.	150-mers db for bracken
Kraken 2.0.7	Protein k=15	RefSeq/08.2018/All, built	TOOLS & REF.	
MetaCache 0.5.0	k=16	RefSeq/08.2018/All, built	TOOLS & REF.	
MetaCache 0.5.0	k=22	RefSeq/08.2018/1rep., built	TOOLS & REF.	
MetaPhlAn 2.7.7	default	Mpa_v20_m200, bundled	TOOLS & REF.	
ganon	k=19	RefSeq/08.2018/1rep., built	TOOLS & REF.	Using only the forward reads
CCMetagen	k=16 +prefix TG	NCBI nt Jan 2018, bundled	TOOLS & REF.	
CLARK-l	--light	RefSeq/08.2018/All, built	TOOLS & REF.	

Kaiju 1.6.0	default	RefSeq/08.2018/1rep. Built with genome exclusion	METHOD ALGO.	
Centrifuge 1.0.3	default	RefSeq/08.2018/1rep. Built with genome exclusion	METHOD ALGO.	Limited to 12 cpus to build the reference
Kraken 2.0.7 + Bracken 2.0	k=35	RefSeq/08.2018/1rep. Built with genome exclusion	METHOD ALGO.	150-mers db for bracken
Kraken 2.0.7	Protein k=15	RefSeq/08.2018/1rep. Built with genome exclusion	TOOLS & REF.	
MetaCache 0.5.0	k=16	RefSeq/08.2018/1rep. Built with genome exclusion	METHOD ALGO.	
MetaCache 0.5.0	k=22	RefSeq/08.2018/1rep. Built with genome exclusion	METHOD ALGO.	
ganon	k=19	RefSeq/08.2018/1rep. Built with genome exclusion	METHOD ALGO.	Using only the forward reads
CCMetagen	k=16 +prefix TG	RefSeq/08.2018/1rep. Built with genome exclusion	METHOD ALGO.	
CLARK-l	--light	RefSeq/08.2018/1rep. Built with genome exclusion	TOOLS & REF.	

Supplemental Table S2

Supplemental Table S2 | Features of the datasets included in the LEMMI initial release. In the case of the LEMMI sets, unknown species and genera represent taxa for which all representatives were selected to generate the reads. Therefore, they are excluded from the reference when using genome exclusion (i.e. for the METHOD ALGORITHMS category). In other datasets, unknown taxa are those not found in the LEMMI/RefSeq repository dated from mid-2018. Datasets having taxa with less than 100 reads are used to compute the low abundance score. The count of non-unique *k*-mers depends on the number, the divergence, and the abundance distribution of organisms in the mock microbial community. It is a measure of the diversity of the reads composing the dataset.

Name	Species count	Genera count	Unknown species	Unknown genera	< 100 reads species	< 100 reads genera	Non unique 50-mers	Number of reads	Abundance standard deviation	RefSeq assembly states
CAMI_I_LOW	23	22	5	1	0	0	579,605,460	50M	-	-
CAMI_I_HIGH_1	243	194	86	7	0	0	1,496,568,850	50M	-	-
mockrobiota-17	10	18	0	0	0	0	68,345,393	1.2M	-	-
LEMMI_LOWDIV_201805_001	100	72	0	0	10	9	115,211,506	10M	2.75	Complete genome
LEMMI_LOWDIV_201805_002	100	71	0	0	8	4	105,634,790	10M	2.75	Complete genome
LEMMI_MEDDIV_201902_001	600	346	338	30	138	53	383,971,742	50M	3.0	Complete genome
LEMMI_MEDDIV_201902_002	600	339	332	40	98	44	455,319,422	50M	3.0	Complete genome
LEMMI_HIGHDIV_201902_001	600	333	393	34	2	2	1,340,621,833	50M	1.75	All

Supplemental Table S3

Supplemental Table S3 | List of metrics available in the dataset detail pages (some datasets do not produce all of them). Underlined entries correspond to those contributing to the rankings.

Metric	Comment
Taxa detection: precision-and-recall curve	Four data points corresponding to filtering abundance below 1/10/100/1000 reads. It comes with a second plot showing the area under the curve.
Taxa detection: recall	Four data points corresponding to filtering abundance below 1/10/100/1000 reads
Taxa detection: precision	Four data points corresponding to filtering abundance below 1/10/100/1000 reads
Taxa detection: true positive count	Four data points corresponding to filtering abundance below 1/10/100/1000 reads
Taxa detection: false positive count	Four data points corresponding to filtering abundance below 1/10/100/1000 reads
Taxa detection: F1-Score	Four data points corresponding to filtering abundance below 1/10/100/1000 reads
Unweighted UniFrac	The lower the better. Not specific to a taxonomic rank
Proportion of assigned reads	At the evaluated taxonomic rank or lower. No distinction of correct or incorrect assignment here.
Normalized rand index	Clustering accuracy at the evaluated taxonomic rank. Lower assignments are moved up to the evaluated rank for evaluation.
Proportion of reads assigned to a false positive taxa	At the evaluated taxonomic rank. Lower assignments are moved up to the evaluated rank for evaluation. Wrong assignment among true positive taxa are not included
Relative abundance error: L1 distance	
Weighted UniFrac	The lower the better. Not specific to a taxonomic rank
Low abundance score	Only for pairs of LEMMI datasets. See methods and Supplemental Fig. 10
Runtime for analysis	For everything that the task within the container need to do, including cleaning/preprocessing fastq
Runtime for building the reference	For everything that the task within the container need to do, including cleaning/preprocessing fasta
Memory used for analysis	Peak memory in GB
Memory used for building the reference	Peak memory in GB

Supplemental Table S4

Supplemental Table S4 | Transformations applied on each metric to compute the ranking score.

Metric	Transformation	Example
Taxa detection: recall	None	0.85
Taxa detection: precision	None	0.15
Unweighted UniFrac	Divided by an arbitrary value of 25,000 and subtracted from 1.0	1 - (16,000/25,000) = 0.36
Proportion of assigned reads	None	0.7
Normalized rand index	None	0.9
Relative abundance error: L1 distance	Divided by its maximum value of 2.0 and subtracted from 1.0	1 - (1.5/2.0) = 0.25
Weighted UniFrac	Divided by its maximum value of 16.0 and subtracted from 1.0	1 - (4/16) = 0.75
Low abundance score	None	0.7
Runtime for analysis Memory used for analysis	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.	Max = 128GB Value1= 90GB 1 - (90/(2*128)) = 0.64 Value2 = 140GB => 0.0 Max = 60min Value1= 55min 1 - (55/(2*60)) = 0.54 Value2 = 65min => 0.0

References

Nasko D, Koren S, Phillippy A, Treangen T. 2018. RefSeq database growth influences the accuracy of *k*-mer-based lowest common ancestor species identification. *Genome Biology* **19**: 165. doi:10.1186/s13059-018-1554-6.

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