

A peer-reviewed version of this preprint was published in PeerJ on 28 March 2017.

[View the peer-reviewed version](https://doi.org/10.7717/peerj.3138) (peerj.com/articles/3138), which is the preferred citable publication unless you specifically need to cite this preprint.

Dadi TH, Renard BY, Wieler LH, Semmler T, Reinert K. 2017. SLIMM: species level identification of microorganisms from metagenomes. PeerJ 5:e3138 <https://doi.org/10.7717/peerj.3138>

SLIMM: Species Level Identification of Microorganisms from Metagenomes

Temesgen Hailemariam Dadi^{1,2}, Bernhard Y. Renard³, Lothar H. Wieler^{1,3}, Torsten Semmler^{1,3}, and Knut Reinert^{1,2}

¹Freie Universität Berlin

²Max Planck Institute for Molecular Genetics

³Robert Koch Institute

ABSTRACT

Identification and quantification of microorganisms is an important step in studying the alpha and beta diversities within and between microbial communities respectively. Both, identification and quantification of a given microbial community can be carried out using whole genome shotgun sequences with less bias than using 16S-rRNA sequences. However, shared regions of DNA among reference genomes and taxonomic units pose a significant challenge in assigning reads correctly to their true origins. The existing microbial community profiling tools commonly deal with this problem by either preparing signature-based unique references or assigning an ambiguous read to its least common ancestor in a taxonomic tree. The former method is limited to making use of the reads which can be mapped to the curated regions, while the latter suffer from the lack of uniquely-mapped reads at higher (more specific) taxonomic ranks. Moreover, even if the tools exhibited generally good performance in calling the organisms present in a sample, there is room for improvement in calling the correct relative abundance of the organisms. We present a new method Species Level Identification of Microorganisms from Metagenomes (SLIMM) which addresses the above issues by using coverage information of reference genomes to remove unlikely genomes from the analysis and subsequently gain more uniquely-mapped reads to assign at higher ranks of a taxonomic tree. SLIMM is based on a few, seemingly easy steps which lead to a tool that outperforms state-of-the-art tools in run-time and/or memory usage while being on par or better in computing quantitative and qualitative information at the species level.

Keywords: Taxonomic Profiling, Metagenomics, Microbial Communities, Microorganisms, NGS Data, Microbiology

INTRODUCTION

Due to the need to study species diversity of a single microbial community (i.e. alpha diversity) and the degree to which a composition of microbial community changes (i.e. beta diversity) (Whittaker, 1960), identification and quantification of microorganisms from shotgun-metagenomic reads obtained by Next Generation Sequencing (NGS) has become an area of growing interest in the field of microbiology. The publication of numerous taxonomic profiling tools within the last decade only shows how appealing the subject is. Lindgreen et al. (2016) considered 14 different sequence classification tools based on different approaches in a recent review of such methods.

Turning metagenomic raw reads into the relative abundance of multiple groups of microorganisms (clades) residing on the sample from which the environmental DNA was extracted and sequenced is a complicated task for several reasons. To mention a few: 1) shared (homologous) regions of genome sequences across multiple microorganisms make an assignment of reads to their potential origin difficult, 2) the range of variation in the abundance of individual groups of microbes in the sample can be high which makes it difficult to detect the least abundant ones and to differentiate noise from true signal, 3) the high degree of variation in publicly available genome sequence lengths of different microbes makes the quantification non-trivial (Brady and Salzberg, 2009).

In the past benchmarking of taxonomic profiling tools was done at the genus or lower level of the taxonomic tree. This is due to the shortcomings of many earlier tools to report species level taxonomic profile with acceptable accuracy. But a species level resolution of microbial communities is desirable and more modern tools do address this (Piro et al., 2016; Lindner and Renard, 2015;

*Corresponding author

Lindgreen et al., 2016; Francis et al., 2013). For this reason, all the benchmarks in this study are done at the species level.

In general, two distinct approaches have been widely used to tackle the challenge of ambiguous reads that originate from genomic locations shared among multiple groups of organisms. The first approach is to prepare a signature-based database with sequences that are unique to a clade. In this approach, taxonomic clades are uniquely represented by sequences that do not share common regions with other clades of the same taxonomic rank. Even if this approach will make use of the fraction of metagenomic data from the sequencer, it can guarantee to have only a unique assignment of sequencing reads to a clade. Tools like MetaPhlAn2 (Truong et al., 2015), GOTTECHA (Freitas et al., 2015) and mOTUs (Sunagawa et al., 2013) use this approach. The second approach works using the full set of reference sequences available as a database and assigning ambiguous reads to their least common ancestor (LCA) in a taxonomic tree. Kraken (Wood and Salzberg, 2014), a k-mer based read binning method, is an example of such an approach. Both approaches have certain advantages and disadvantages. The former has an advantage in speed and precision, but is limited to utilizing the reads that can be mapped uniquely to the curated regions. The later approach, on the other hand, suffers from the lack of uniquely-mapped reads at higher (more specific) taxonomic ranks since they are assigned to the LCA.

Based on the final output of a method there are two categories of metagenomic classification tools, a read binning method and a taxonomic profiling method. A read binning method assigns every single read to a node in a taxonomic tree, whereas a taxonomic profiling method tries to report which organisms or clades are present in the sample with or without having to assign every read to a corresponding taxon. There is an overlap between the two categories such that some read binning methods can be taxonomic profilers too.

GOTTECHA uses a signature-based database specific to a given taxonomic rank and it is highly optimized for low false discovery rate (FDR). Kraken uses instead a database comprising a hash table of k-mers and their corresponding node in a given taxonomic tree. Then it assigns reads based on where the majority of its k-mers are located in the tree and whenever there is no clear vote by the k-mers of the read, it will assign it to its least common ancestor. Kraken is a very fast read binning method, which is also often used to do taxonomic profiling. mOTUs uses single copy universal marker genes to achieve a species level abundance resolution of microbial communities. Even if the tools exhibited generally good performance in calling the organisms present in a sample, there is a room for improvement in calling the correct relative abundance of the called organisms.

In the following, we present a new method we call *Species Level Identification of Microorganisms from Metagenomes* (SLIMM), which addresses the above limitations. At preprocessing stage, we gather as many as possible reference sequences of a group of interest (Archaea, Bacteria, Viruses or any combination of these) and downsize and compile taxonomic information of the gathered sequences. The taxonomic information is stored in the form of SLIMM database (SLIMMDB). Then we use a read mapper to align metagenomic reads against the gathered reference sequences, which we consider as a preprocessing step that often is done for numerous other analyses (we will report on timing with and without preprocessing). SLIMM works on the resulting BAM/SAM alignment file. First, SLIMM uses coverage information both by the reads that mapped on different reference sequences and by reads uniquely-mapped to a reference sequence to remove unlikely genomes from the analysis similar to an approach taken by Lindner et al. (2013). This, in turn, allows us to subsequently gain a larger number of uniquely-mapped reads in relation to the reduced set of genomes which we can assign at higher ranks of a taxonomic tree. We will show that this simple approach has indeed positive effects. The second step is to assign the remaining non-uniquely-mapped reads to the lowest common ancestor. Overall SLIMM is based on a few, seemingly easy steps resulting in a tool that outperforms state-of-the-art tools in run-time and/or memory usage while being on par or better in computing quantitative and qualitative information at the species level which we show in the results section. Following the recommendation in (Piro et al., 2016) with caution, we have carried out digital normalization on the raw reads (Brown et al., 2012) which discards low quality and redundant reads. It works by removing reads belonging to or (would result in) high coverage depth. In our experience, the digital normalization showed a negligible improvement in calling the correct organisms.

METHOD

Nonredundant Reference genomes database

Reference genomes from NCBI GenBank (<ftp://ftp.ncbi.nlm.nih.gov/genomes/genbank/>) and RefSeq (<ftp://ftp.ncbi.nlm.nih.gov/genomes/refseq/>) archives, downloaded on 21.05.2016, are used for

the method we described here. SLIMM is not limited to these public databases provided that there is a proper mapping from sequence identifiers to a taxonomic id and a taxonomic tree that represents all the sequences in the database. For this study, we considered microbes under the super-kingdom of archaea and bacteria. But one can also easily integrate viruses into the database by using the provided SLIMM preprocessing tool. Before downloading all the genomes we checked for redundancy by counting the number of available files for each species of interest. If multiple genomes are present for downloading, then we choose one in the order of 1) RefSeq 2) Complete Genome and 3) Draft Genome. This enabled us to have as many species as possible represented by their best reference genome available so far. After downloading sequences, we checked if every genomic file downloaded contains only a single fasta entry. If not, we take their concatenation separated by a contiguous sequence of ten N's so that reads will not map at the joining point by accident. The final result is a reference genome library of organisms of interest group(s), which contains a single representative sequence per species. In order to cope with dynamically expanding reference genomes library, we implemented a feature as SLIMM preprocessing tool that can seamlessly update the reference genome database. In this way, we get two databases that we call small_DB and large_DB. Small_DB contains 2163 species only with complete genomes while large_DB contains 13192 species including those with only draft genomes available.

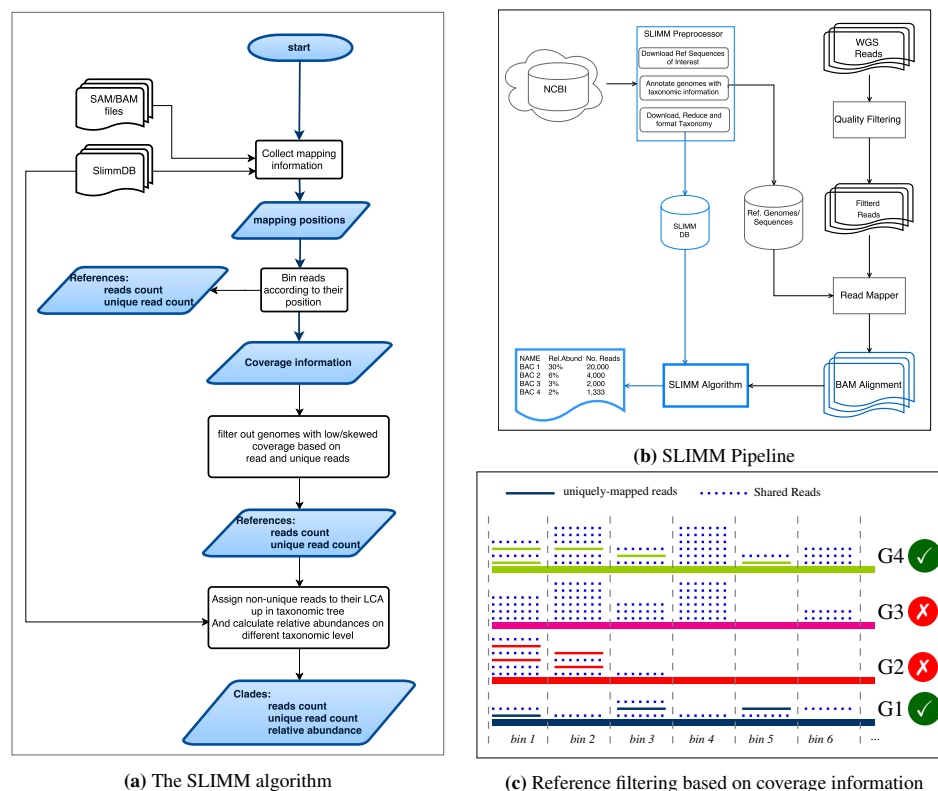


Figure 1. Overview of the SLIMM methodology: (a) : SLIMM takes two inputs i.e the SLIMMDB and an alignment file in either SAM or BAM format and outputs statistics about each reference sequences in the database. Then SLIMM uses coverage information to leave out reference sequences from consideration and recalculate the statistics again. We use this, in turn, to get read counts that are uniquely-mapped to a clade at a given taxonomic rank. (b): The preprocessing module of SLIMM downloads/updates all available genomes of interest group (Archaea, Bacteria, Viruses or any combination of them) and tags the sequences with their corresponding taxonomic information. A read mapper is then used to map the WGS reads to these reference sequences. Then SLIMM algorithm works on the mapping results and produces taxonomic profile reports. (c) an illustration of how SLIMM does reference filtering based on coverage information: G2 and G3 could not pass the filtering steps because they did not have enough coverage by uniquely-mapped reads and all reads respectively.

Read mapping against a database of interest

SLIMM requires an alignment/mapping file in SAM or BAM format as an input. The alignment file can be obtained by aligning the short shotgun metagenome sequencing reads against a library of reference genomes of interest. To do so, one can use a read mapper of choice. Nevertheless the pipeline could benefit from a faster but yet accurate read mapper as this preprocessing step

is relatively time consuming. We make the read mapping program output secondary alignments because 1) it is very likely to have a sequencing read mapped to multiple targets, 2) a read might have multiple best hits and 3) the best hit of a read might not be its true origin. SLIMM uses coverage landscape information as shown in figure (1c) to resolve this. We tried bowtie2 (Langmead and Salzberg, 2012) and Yara Siragusa (2015) in our first experiments, because they are fast read mappers with multi-threading options. Since Yara is several factors faster, does not employ heuristics and its resulting alignments produced better profiles in some of the cases, we used it as default mapper for this study.

Collecting coverage information of each reference genomes

We first identify which reads are mapped to which reference genomes and separate reads, which are uniquely mapped to a single reference sequence (these include reads that are mapped to multiple places in a single reference) from those which are mapped to multiple reference sequences. At this stage, SLIMM collects information like the number of reference genomes with mapping reads, the total number of reads and the average read length, which will later be used for discarding reference genomes. Then we map reads into bins of specific width across each reference genome based on their mapping location. The default bin width is computed as the average length of sequencing reads present in the input mapping file and there is a possibility to set it to a different value. Higher bin width can result in faster runtime, but could also lead to underrepresentation of coverage information depending on the overall coverage depth. We repeat the filtration procedure, this time, using only uniquely-mapped reads. The bin number corresponding to a read mapped to a reference is defined by the center position of its mapping location divided by the width of the bins (integral part only). The bin number of a read mapping to a reference starting from loc_{start} all the way to loc_{end} is given by:

$$binNumber = \left\lfloor \frac{loc_{start} + loc_{end}}{2 \times w} \right\rfloor \quad (1)$$

where w is the width of bins a reference is partitioned into. After binning is done, coverages based on mapping reads and uniquely-mapped reads are calculated based on their corresponding binning. Coverage information of each reference sequence is represented by coverage percentage (%Cov) and coverage depth (CovDepth) as shown in equations 2 and 3 respectively.

$$\%Cov = \frac{|bins'|}{|bins|} \times 100 \quad (2)$$

$$CovDepth = \frac{\sum_{i=1}^{|bins|} \left(\frac{\sum_{j=1}^{N_{bin}} readLength}{w} \right)}{|bins|} \quad (3)$$

where $|bins'|$ is the number of non-zero bins, $|bins|$ is the total number of bins in the reference, N_{bin} is the number of reads in a bin, $readLength$ is the number of bases in a read, and w is the width of a bin.

Discarding unlikely genomes based on coverage landscape

We discard reference sequences that have coverage percentage below a threshold. The threshold is calculated based on a given percentile (default 0.001) of all coverage percentages of the genomes. In other words after sorting the reference sequences based on their coverage percentage in descending order we take the top N sequences that cover 99.999 % of the sum of all coverage percentages. This is done both for coverage percentage by reads that mapped on multiple references and uniquely-mapped reads. This will eliminate many genomes even if they have a lot of reads mapping to them as long as they do not have a good enough coverage. This method was also proven to eliminate reference sequences that acquire a stack of reads only in one or two bins across their genomes. This could be explained either by a sequencing artifact from the mock community metagenome dataset or a conserved region in the genome among distant relatives.

Recalculating reads uniqueness after discarding unlikely genomes

After discarding reference sequences, SLIMM calculates the uniqueness of reads again. This can increase the number of uniquely-mapped reads assigned to higher-level clades in a taxonomic tree. The recalculation of uniquely-mapped reads is shown to improve the abundance estimation of a taxon (clade).

Assigning reads to their LCA and calculating abundances at a given rank

After recalculating the uniqueness of reads we assign non-uniquely-mapped reads to their LCA taxon based on the NCBI taxonomic tree downloaded from <ftp://ftp.ncbi.nih.gov/pub/taxonomy>. Instead of using the whole NCBI taxonomic tree we use a reduced subtree produced by the SLIMM preprocessing tool. Since we report only for a given major taxonomic ranks namely superkingdom (domain), phylum, class, order, family, genus and species, the reduced tree contains only these taxonomic ranks. We also discarded the branches of the tree that are out of the interest groups i.e. Archaea and Bacteria for this study. This saves a significant amount of computational time as assigning a read to its LCA is computationally expensive. We also propagate the number of uniquely-mapped reads at a node to any of its ancestors. Then we calculate the relative abundance of each taxonomic unit at a given rank as the uniquely-mapped reads that are assigned to it divided by the total number of uniquely mapped reads at the rank (equation 4). We also report an aggregated coverage depth of each clade defined as in equation 5.

$$RelAb_{clade} = \frac{N_{clade}}{N_{mapped}} \quad (4) \quad CovDepth_{clade} = \frac{\sum_{i=1}^{N_{clade}} readLength}{\sum_{i=1}^{N_{child}} refLength} \quad (5)$$

$RelAb_{clade}$ is the relative abundance of a clade, N_{clade} is the number of reads that are assigned to a clade, N_{mapped} is the total number of reads that are mapped to any clade, $CovDepth_{clade}$ is coverage depth of a clade, $readLength$ is the number of bases in a read, and $\sum_{i=1}^{N_{child}} refLength$ is the sum of reference lengths of children of a clade that contribute at least one read.

RESULTS AND DISCUSSION

Datasets

For this study, we assembled 18 different metagenomic datasets of varying origins and simulation strategies. The datasets contain 1) mock community metagenomes from two different studies that are sequenced using Illumina Genome Analyzer II 2) simulated metagenomes that resemble community profile of an existing metagenome as identified by MetaPhlAn2 (Truong et al., 2015) 3) simulation of randomly created microbial communities with a varying number of organisms and range of relative abundances. We used NeSSM (Jia et al., 2013) to do the simulations. 4) Medium complexity CAMI (The Critical Assessment of Metagenome Interpretation) challenge toy datasets that are publicly available at <https://data.cami-challenge.org/participate>. We believe that this collection of datasets can represent most of metagenomic communities that a taxonomic identifier will have to handle.

We used 3 mock community datasets, 2 from the Human Microbiome Project (HMP) (HMP, 2012) containing genomes of 22 microorganisms and 1 from the study (Shakya et al., 2013) containing genomes of 64 microorganisms. The 2 datasets from HMP are the similar in the species they contain, they only differ in the abundance distribution. One contains an even abundance distribution of the microorganisms whereas the other contains a differing abundance distribution of the 22 microorganisms.

For simulated datasets resembling an existing community we chose: 1) a metagenome obtained from the human gut sample during the HMP (HMP, 2012) 2) a freshwater metagenome dataset from Lake Lanier (Oh et al., 2011). We used MetaPhlAn2 (Truong et al., 2015) - a well known metagenomic profiling tool based on use clade-specific marker genes. Then we used the reported profile as a basis for the simulation.

For randomly created microbiomes, we considered three communities with randomly selected member organisms. The number of organisms in these communities is 50, 200, 500. Then we chose three different ranges of relative abundances i.e. even, [1-100] and [1-1000]. This provided us with a total of 9 randomly created metagenomes with varying complexity both in terms of diversity and in abundance differences. The different settings of metagenomic datasets are important to make sure that the tested methods work with a wide range of input datasets. To resemble an actual metagenome and to make the taxonomic profiling more difficult, we contaminated all the simulated datasets with real world metagenomic reads sequenced by Illumina MiSeq, after removing the reads that could be mapped to any of prokaryote genome available. Details of all the datasets used for evaluation can be found in the supplementary material.

Performance Comparison.

We compared the runtime and accuracy of SLIMM with other existing taxonomic profiling tools. We considered GOTTCHA, MOTUs and Kraken as recent and frequently used reference-based shotgun

metagenome classification tools for comparison. For Kraken we created a Kraken database corresponding to both small_DB and large_DB. We use large_DB only for the CAMI datasets as these datasets contain species that have only their draft genomes available. GOTTECHA and mOTUs use their own curated database. Table 1 shows the average runtime and the average peak memory usage of the tools across runs on the 14 different datasets, excluding the CAMI datasets, used in this study. We used a machine with 32 (Intel(R) Xeon(R) CPU 3.30GHz) processors and 378GB of memory. The CAMI datasets are not included in the runtime and memory comparison because we could not run Kraken with large_DB on the same machine because it required 500GB of memory. Instead we run Kraken on a cluster for these particular datasets. Without the time needed for the preprocessing SLIMM is proven to be faster than any of the other tools considered while using an intermediate memory footprint. With the preprocessing, Kraken is faster but uses much more memory. SLIMM is faster than GOTTECHA and mOTUs. The individual runtime per dataset can be found in the supplementary information.

Table 1. Runtime and Memory Comparison of SLIMM against existing methods

	Alignment + SLIMM	Kraken	GOTTECHA	mOTUs
Avg. Runtime (Seconds)	422.1 + 61.0	157.4	1727.1	1526.6
Peak Memory (GB)	5.2	102	4	1.6

We used different accuracy measures namely precision(specificity), recall (sensitivity) and F1-Score to compare the accuracy of each tool with SLIMM. The definition of the accuracy measures is given below.

$$precision = \frac{TP}{TP + FP} \quad (6) \quad recall = \frac{TP}{TP + FN} \quad (7)$$

$$F_1 = 2 \cdot \frac{precision \cdot recall}{precision + recall} \quad (8)$$

Where TP=true positives (species which are in the samples and called by the tools), TN=true negatives (species which are not in the samples and not called by the tools), FP=false positives (species which are not in the samples and yet called by the tools) and FN=false negatives (species which are in the samples but not called by the tools)

Table 2. Comparison of SLIMM against different tools in terms of precision and recall on species levelThe highest values in each row are marked bold for both precision and recall. precision is defined as number of species that are called and are in the sample (true positives) divided by the number of species that are called (true positives + false positives). Whereas recall is defined as number of species that called and are in the sample (true positives) divided by the number of species that were in the sample (true positives + false negatives). * GOTTECHA and mOTUs have unfairly lower recall and F1 values due to their own database which does not contain the complete set of references for the corresponding datasets

Type	Dataset	Precision				Recall				F1			
		SLIMM	Kraken	GOTTECHA	mOTUs	SLIMM	Kraken	GOTTECHA	mOTUs	SLIMM	Kraken	GOTTECHA	mOTUs
Mock	MG01	0.8923	0.6264	0.9808	1.0000	0.9355	0.9194	0.8226	0.8065	0.9134	0.7451	0.8947	0.8929
	MG02	0.9545	0.8400	1.0000	1.0000	1.0000	1.0000	0.9524	0.8571	0.9767	0.9130	0.9756	0.9231
	MG03	0.9524	0.6897	1.0000	1.0000	0.9524	0.9524	0.8571	0.4286	0.9524	0.8000	0.9231	0.6000
Mimic.Sim	MG04	1.0000	0.4250	0.6000	0.9474	1.0000	1.0000	0.6176	0.5294	1.0000	0.5965	0.6087	0.6792
	MG05	1.0000	0.6650	0.8714	0.9630	1.0000	1.0000	0.4656	0.1985	1.0000	0.7988	0.6070	0.3291
	MG06	0.9783	0.4352	0.6897	0.8718	0.9375	0.9792	0.8333	0.7083	0.9574	0.6026	0.7547	0.7816
Rand.Sim	MG07	0.9783	0.4352	0.6964	0.9091	0.9375	0.9792	0.8125	0.6250	0.9574	0.6026	0.7500	0.7407
	MG08	0.9783	0.4299	0.7143	0.8824	0.9375	0.9583	0.8333	0.6250	0.9574	0.5935	0.7692	0.7317
	MG09	0.9929	0.7220	0.8396	0.9286	0.9211	0.9737	0.5855	0.3421	0.9556	0.8291	0.6899	0.5000
CAMI	MG10	0.9930	0.7178	0.7949	0.9574	0.9276	0.9539	0.4079	0.2961	0.9592	0.8192	0.5391	0.4523
	MG11	0.9928	0.7164	0.8058	0.9464	0.9079	0.9474	0.5461	0.3487	0.9485	0.8159	0.6510	0.5096
	MG12	0.9855	0.8284	0.7333	0.9773	0.9315	0.9589	0.0377	0.1473	0.9577	0.8889	0.0717	0.2560
CAMI	MG13	0.9855	0.8237	0.8095	0.9811	0.9315	0.9281	0.0582	0.1781	0.9577	0.8728	0.1086	0.3014
	MG14	0.9851	0.9857	0.8000	0.9811	0.9041	0.9452	0.0548	0.1781	0.9429	0.9650	0.1026	0.3014
	MG15	0.9261	0.7644	0.7397	0.8000	0.8191	0.7990	0.2714*	0.1206*	0.8693	0.7813	0.3971*	0.2096*
CAMI	MG16	0.8377	0.7027	0.6883	0.8462	0.8040	0.7839	0.2663*	0.1106*	0.8205	0.7411	0.3841*	0.1956*
	MG17	0.9302	0.7608	0.4531	0.7368	0.8040	0.7990	0.1457*	0.1407*	0.8625	0.7794	0.2205*	0.2363*
	MG18	0.8223	0.6996	0.4839	0.7778	0.8141	0.7839	0.1508*	0.1407*	0.8182	0.7393	0.2299*	0.2383*

Table 2 shows the result of performance comparison among SLIMM and existing metagenomic classifiers using 18 different datasets described above. SLIMM outperforms all of the tools in 13 out of the 18 cases in precision. SLIMM and Kraken showed good results in recall. SLIMM came in second place outperforming Kraken occasionally. But Kraken produced more false positives to attain this recall, hence the lower numbers in precision. GOTTECHA performed well with the HMP datasets while it underperformed in the rest of the datasets in general. mOTUs does not perform well in all of the datasets. F1-Score is also provided in the table as a measure of good balance between

precision and recall. SLIMM outperforms all the other tools both in precision and F1-Score 17 of the 18 cases while kraken is slightly better in recall in the majority of the cases.

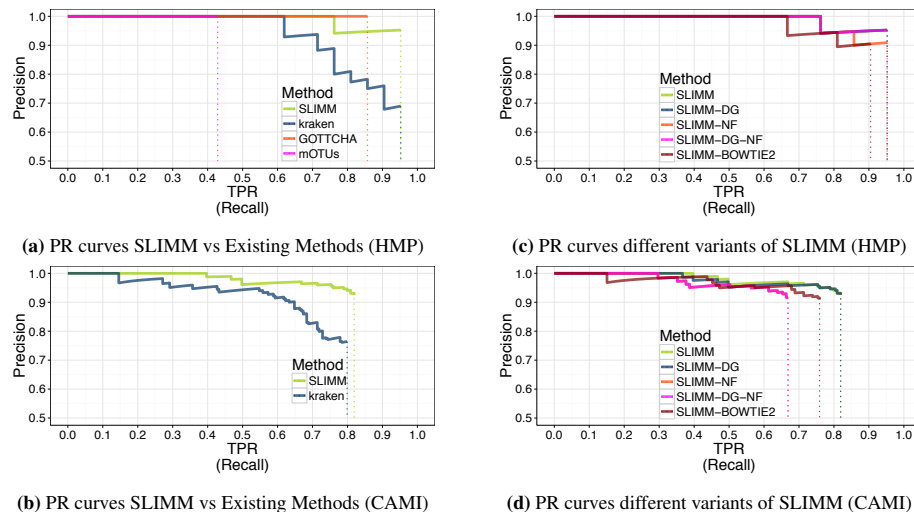


Figure 2. PR Curves: Comparison of SLIMM against existing methods (a and b) PR curves SLIMM vs Existing Methods: True Positive Rate(TPR)/recall down against precision. SLIMM showed the highest performance. GOTTCHA didn't discover any false positives but is low in recall. **(c and d) PR curves different variants of SLIMM:** SLIMM i.e. SLIMM-DG (with digital normalization), SLIMM-NF (without filtration step based on coverage landscape), SLIMM-NF-DG (without filtration but with digital normalization) and SLIMM using alignment produced by the read mapper Bowtie2.

We did a PR curve analysis for the HMP mock community dataset with uneven distribution of relative abundances of member organisms and one of the CAMI challenge datasets. We sorted the predicted species by predicted abundance in decreasing order to draw the PR curves. The PR curves (figure 2a) shows that SLIMM has a better recall rate than the other tools while staying precise.

SLIMM's ability to predict the correct abundances of organisms better than existing methods is shown by scatter-plots in figure 3a and 3b by plotting the real abundance of organisms against their predicted abundance by different tools for one of the CAMI challenge datasets and one of the randomly simulated datasets. From these plots, it can be clearly seen that SLIMM predicts the abundance more accurately. Even though it is not originally developed for abundance estimation, the next best tool is Kraken which slightly overestimates the true abundance. mOTU and GOTTCHA do not perform well at predicting the abundances.

Violin plots are similar to box plots, but they also show the density distribution of different data points. The violin plots in figure 3c and 3d show how divergent the abundance predicted by different tools is from the actual true abundance. In the plots we can see that SLIMM has very low divergence from the real abundance. For the randomly simulated dataset, SLIMM has an average absolute difference of 0.00073 and Kraken has an average absolute difference of 0.00116 which is 159% higher compared to SLIMM. For the same dataset, GOTTCHA and mOTUs have average absolute difference of 0.00206 and 0.00273 respectively. SLIMM also got the most correct (closer) abundances with absolute differences of first quantile (Q1)=0.00002 and third quantile (Q3)=0.00016. Kraken is the second best tool in this regard with values Q1=0.00018, Q3=0.00065.

We have also investigated the positive effects of the filtering step in SLIMM. We run SLIMM with the filtration turned off and compare the results with a normal run of SLIMM. Figure 2c shows that the filtration step leads to better results. It is also interesting to note that SLIMM's filtration step effectively reduces the divergence from the true abundance. Figures 3c and 3d show that SLIMM's filtration step produced closer abundances i.e. quantiles of absolute differences between real and predicted abundances are (Q1=0.00002, Q2=0.00004, 0.00016) with filtration compared to (Q1=0.00002, Q2=0.00006, 0.00082) without filtration. More plot for other dataset can be found in the supplement.

In conclusion, we described a method that results in a simple, fast and scalable tool for taxonomic profiling and abundance estimation which utilizes coverage information of individual genomes to filter out those that are unlikely to be in the sample. This is done by discarding genomes with relatively low coverage percentage by uniquely-mapped reads and mapping reads in general. Such simple yet important filtration step makes SLIMM capable of calling organisms with high recall rate

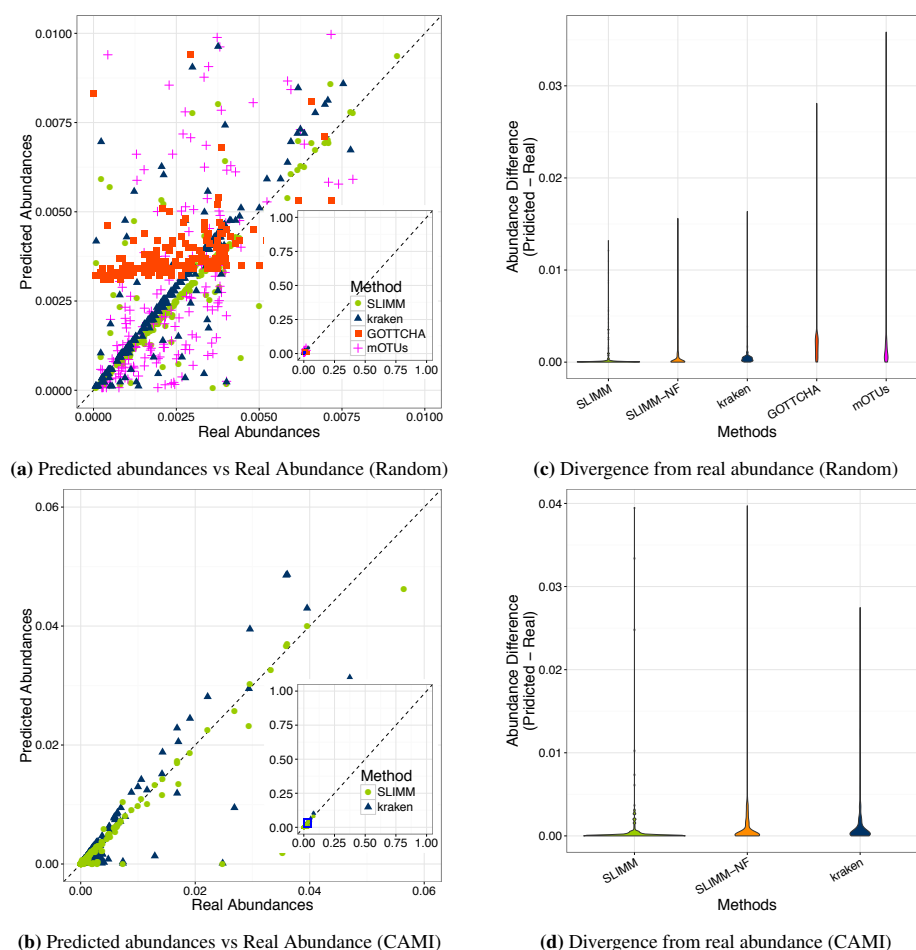


Figure 3. Predicting abundances correctly (a) and (b): Abundances predicted by different tools compared to the real abundance used for simulation. SLIMM predicted the abundances more accurately than the other tools. Kraken overestimates the abundance. GOTTECHA and mOTUs did not perform well in predicting the abundances. **(c) and (d):** Violin plots showing that SLIMM has the lowest divergence from true abundances

while remaining precise. We showed that SLIMM methodology resulted in more accurate taxonomic profiling as well as predicting the individual abundance of member organisms more accurately than the other tools.

ACKNOWLEDGMENTS.

This work is supported by the International Max Planck Research School for Computational Biology and Scientific Computing and by the InfectControl 2020 Project (TFP-TV4). All authors worked on the preparation of the manuscript. The implementation and testing are performed by T. Dadi. We would like to thank Martin Lindner for his helpful discussions and ideas on the subject matter. We also thank the anonymous reviewers for their constructive comments.

Supplements.

Supplements of this paper are available at <http://ftp.mi.fu-berlin.de/dadi/slimm/>.

Availability.

SLIMM is developed in C++ with SeqAn Library (Döring et al., 2008) and freely available at <https://github.com/temehi/slimm>.

REFERENCES

Brady, A. and Salzberg, S. L. (2009). Phymm and PhymmBL: metagenomic phylogenetic classification with interpolated Markov models. *Nature methods*, 6(9):673–6.

- Brown, C. T., Howe, A., Zhang, Q., Pyrkosz, A. B., and Brom, T. H. (2012). A Reference-Free Algorithm for Computational Normalization of Shotgun Sequencing Data. *arXiv*, 1203.4802(v2):1–18.
- Döring, A., Weese, D., Rausch, T., and Reinert, K. (2008). SeqAn an efficient, generic C++ library for sequence analysis. *BMC bioinformatics*, 9:11.
- Francis, O. E., Bendall, M., Manimaran, S., Hong, C., Clement, N. L., Castro-Nallar, E., Snell, Q., Schaalje, G. B., Clement, M. J., Crandall, K. A., and Johnson, W. E. (2013). Pathoscope: Species identification and strain attribution with unassembled sequencing data. *Genome Research*, 23(10):1721–1729.
- Freitas, T. A. K., Li, P.-E., Scholz, M. B., and Chain, P. S. G. (2015). Accurate read-based metagenome characterization using a hierarchical suite of unique signatures. *Nucleic Acids Research*, 43(10):gkv180.
- HMP, C. (2012). A framework for human microbiome research. *Nature*, 486(7402):215–221.
- Jia, B., Xuan, L., Cai, K., Hu, Z., Ma, L., and Wei, C. (2013). NeSSM: A Next-Generation Sequencing Simulator for Metagenomics. *PLoS ONE*, 8(10).
- Langmead, B. and Salzberg, S. L. (2012). Fast gapped-read alignment with Bowtie 2. *Nat Methods*, 9(4):357–359.
- Lindgreen, S., Adair, K. L., and Gardner, P. P. (2016). An evaluation of the accuracy and speed of metagenome analysis tools. *Scientific Reports*, 6:19233.
- Lindner, M. S., Kollock, M., Zickmann, F., and Renard, B. Y. (2013). Analyzing genome coverage profiles with applications to quality control in metagenomics. *Bioinformatics*, 29(10):1260–1267.
- Lindner, M. S. and Renard, B. Y. (2015). Metagenomic profiling of known unknown microbes with MicrobeGPS. *PLoS ONE*, 10(2):e0117711.
- Oh, S., Caro-Quintero, A., Tsementzi, D., DeLeon-Rodriguez, N., Luo, C., Poretsky, R., and Konstantinidis, K. T. (2011). Metagenomic insights into the evolution, function, and complexity of the planktonic microbial community of Lake Lanier, a temperate freshwater ecosystem. *Applied and Environmental Microbiology*, 77(17):6000–6011.
- Piro, V. C., Lindner, M. S., and Renard, B. Y. (2016). DUDes: a top-down taxonomic profiler for metagenomics. *Bioinformatics*, page btw150.
- Shakya, M., Quince, C., Campbell, J. H., Yang, Z. K., Schadt, C. W., and Podar, M. (2013). Comparative metagenomic and rRNA microbial diversity characterization using archaeal and bacterial synthetic communities. *Environmental Microbiology*, 15(6):1882–1899.
- Siragusa, E. (2015). *Approximate string matching for high-throughput sequencing*. PhD thesis, Freie Universität Berlin.
- Sunagawa, S., Mende, D. R., Zeller, G., Izquierdo-Carrasco, F., Berger, S. a., Kultima, J. R., Coelho, L. P., Arumugam, M., Tap, J., Nielsen, H. B., Rasmussen, S., Brunak, S., Pedersen, O., Guarnier, F., de Vos, W. M., Wang, J., Li, J., Dore, J., Ehrlich, S. D., Stamatakis, a., and Bork, P. (2013). Metagenomic species profiling using universal phylogenetic marker genes. *Nat Methods*, 10(12):1196–1199.
- 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. *Nature Methods*, 12(10):902–903.
- Whittaker, R. H. (1960). Vegetation of the siskiyou mountains, oregon and california. *Ecological Monographs*, 30(3):279–338.
- Wood, D. E. and Salzberg, S. L. (2014). Kraken: ultrafast metagenomic sequence classification using exact alignments. *Genome biology*, 15(3):R46.