Quantitative Biology 2018, 6(2): 175–185
https://doi.org/10.1007/s40484-018-0142-9

METHODOLOGY ARTICLE

Comprehensive simulation of metagenomic sequencing data with non-uniform sampling distribution

Shansong Liu, Kui Hua, Sijie Chen and Xuegong Zhang*

MOE Key Lab of Bioinformatics, Bioinformatics Division, TNLIST and Department of Automation, Tsinghua University, Beijing 100084, China
* Correspondence: zhangxg@tsinghua.edu.cn

Received November 8, 2017; Revised February 7, 2018; Accepted February 27, 2018

***Background*: Metagenomic sequencing is a complex sampling procedure from unknown mixtures of many genomes. Having metagenome data with known genome compositions is essential for both benchmarking bioinformatics software and for investigating influences of various factors on the data. Compared to data from real microbiome samples or from defined microbial mock community, simulated data with proper computational models are better for the purpose as they provide more flexibility for controlling multiple factors.**
***Methods*: We developed a non-uniform metagenomic sequencing simulation system (nuMetaSim) that is capable of mimicking various factors in real metagenomic sequencing to reflect multiple properties of real data with customizable parameter settings.**
***Results*: We generated 9 comprehensive metagenomic datasets with different composition complexity from of 203 bacterial genomes and 2 archaeal genomes related with human intestine system.**
***Conclusion*: The data can serve as benchmarks for comparing performance of different methods at different situations, and the software package allows users to generate simulation data that can better reflect the specific properties in their scenarios.**

Keywords: simulation; metagenomic sequencing data; non-uniform sampling; nuMetaSim

Author summary: nuMetaSim—a non-uniform metagenomic sequencing simulation system is developed as well as a comprehensive simulated metagenomic dataset is designed in this work. The nuMetaSim software provides flexible simulation settings for mimicking different real metagenomic data features under distinct scenarios for researchers with various purposes. The designed simulated dataset can be used as a benchmarking set to verify the performance of metagenomic data processing software utilized by those needed researchers. Python codes of nuMetaSim are released by the following GitHub link: https://github.com/dadinghh2/nuMetaSim

INTRODUCTION

Rapidly growing sequencing data produced by the next-generation sequencing (NGS) technology have revolutionized many biological research fields such as the study of non-culturable microbial communities. One can apply sequencing technology on microbial communities to get metagenomic sequencing data and use bioinformatics processing and analysis to turn the data into information about the microbial composition and functions [1–3]. Metagenomic studies have shown that human microbiomes are related with many human diseases such as diabetes [4], obesity [5] and nutrition disorder [6]. However, the conclusion in all such studies heavily relies on the correctness and accuracy of the adopted bioinformatics software in analyzing the particular data [7]. Due to the complex nature of metagenomic data, well-controlled testing and comparison of bioinformatics tools is crucial for microbiome studies.

In general, there are three types of metagenomic data that may be used to test bioinformatics methods: real data from natural microbial community samples, or data from mock community samples with designed composition, and simulated data generated by computational models. For real data, the true answers to key questions such as the composition and abundances of microbial species are unknown *a priori*. So their utility for testing bioinformatics methods is limited. For mock community data, the members of the microbial community are manually selected from sequenced genomes. Hence the community constituents are under control [8]. But sometimes mock community data could be unreliable because of contamination of other unknown microbes [9]. Furthermore, due to difficulties in experiments, a mock community often contains a small amount of taxa and the degree of variation in their relative abundances is also limited. These limits may cause overfitting of methods and observations on mock data may not be generalizable to more complicated real data [8].

For simulation data, the obvious advantage is the possibility of full control on all aspects of data such as the microbial composition, sequence sampling, sequencing errors, noises like human contamination and unknown microbes that are usually present in a real microbial environment [10–12]. However, these advantages depend on how well the data simulates complexities and unideal properties of real data. For example, in the ideal case, the DNA library building procedure should be a uniform sampling process from all genomes. But this is rarely true on real data as the sequencing read distribution on each genome can be very uneven. Read coverage on different genomes can also have large variations [1]. Unideal conditions like these need to be carefully considered and properly reflected in simulation data.

Simulated metagenomic data have already been used in many software benchmarking studies [7,10,12–16]. Several simulators have been developed to generate simulated metagenomic data [17], such as MetaSim [18], NeSSM [19], BEAR [20], and FASTQSim [21]. However, most existing simulators and simulated datasets did not try to capture all possible features of realistic metagenomic data. The earliest metagenomic simulator MetaSim only considered a few features in real data, such as microbial composition, abundance and sequencing error. Quality score of sequencing data was not included in MetaSim. Both MetaSim and FASTQSim did not grant uneven sampling on the reference genomes. Only NeSSM and BEAR are able to simulate uneven sampling or coverage bias, which is a salient feature existing in real data. Coverage bias is mainly attributed to GC-content when DNA fragments are undergoing PCR amplification [22,23]. NeSSM maps the empirical data to reference genomes with fixed-size bins to obtain a read-mapping profile that represents the coverage bias. However, in some cases, there might be only a fraction of reads sampled from the reference genomes, and therefore the integral distributions over the sampled genomes cannot be acquired. BEAR tries to find the relationship between read length of empirical data and GC-content, then uses this to randomly sample reads from reference genomes.

The nuMetaSim simulator we developed is an effort to consider all major aspects of real metagenomic sequencing data in the simulation model. Users can set their own parameters according to the factors they want to study with the simulation data. Besides the common features such as microbial composition, sequencing error, quality score adopted by existing simulators, we incorporated human contamination and unknown reads generation as a built-in function. For coverage bias, we adopted a disparate strategy with existing ones to calculate the GC-content-based coverage bias profile. We also provided 9 sets of simulation data for different scenarios with regard to the number of component genomes and their relative abundance patterns. These data can be a basic benchmark set for metagenomic data processing methods. Users can also design and generate their own data with special properties using nuMetaSim.

RESULTS

We developed a package nuMetaSim that can generate simulated metagenome datasets with comprehensive features existing in real sequencing data, such as GC-content coverage bias caused by sequencing machine, human and unknown reads derived from pollution or improper experimental operation. It is meaningful to consider these features when benchmarking bioinformatics software since real metagenome sequencing data contain these features. Then the benchmarking result can be more reliable using the realistic simulated metagenome sequencing data.

Overview of nuMetaSim

As shown in Figure 1, the simulation process in nuMetaSim is composed of four steps. The optional step 2 is designed for simulating the real data features. Users can select some of the features or all of them to produce their specialized simulated dataset. The other steps are required.

STEP 1 – Reference genomes and abundance profile preparation

nuMetaSim is a reference-based metagenomic simulator, developed with Python scripts. Complete microbial

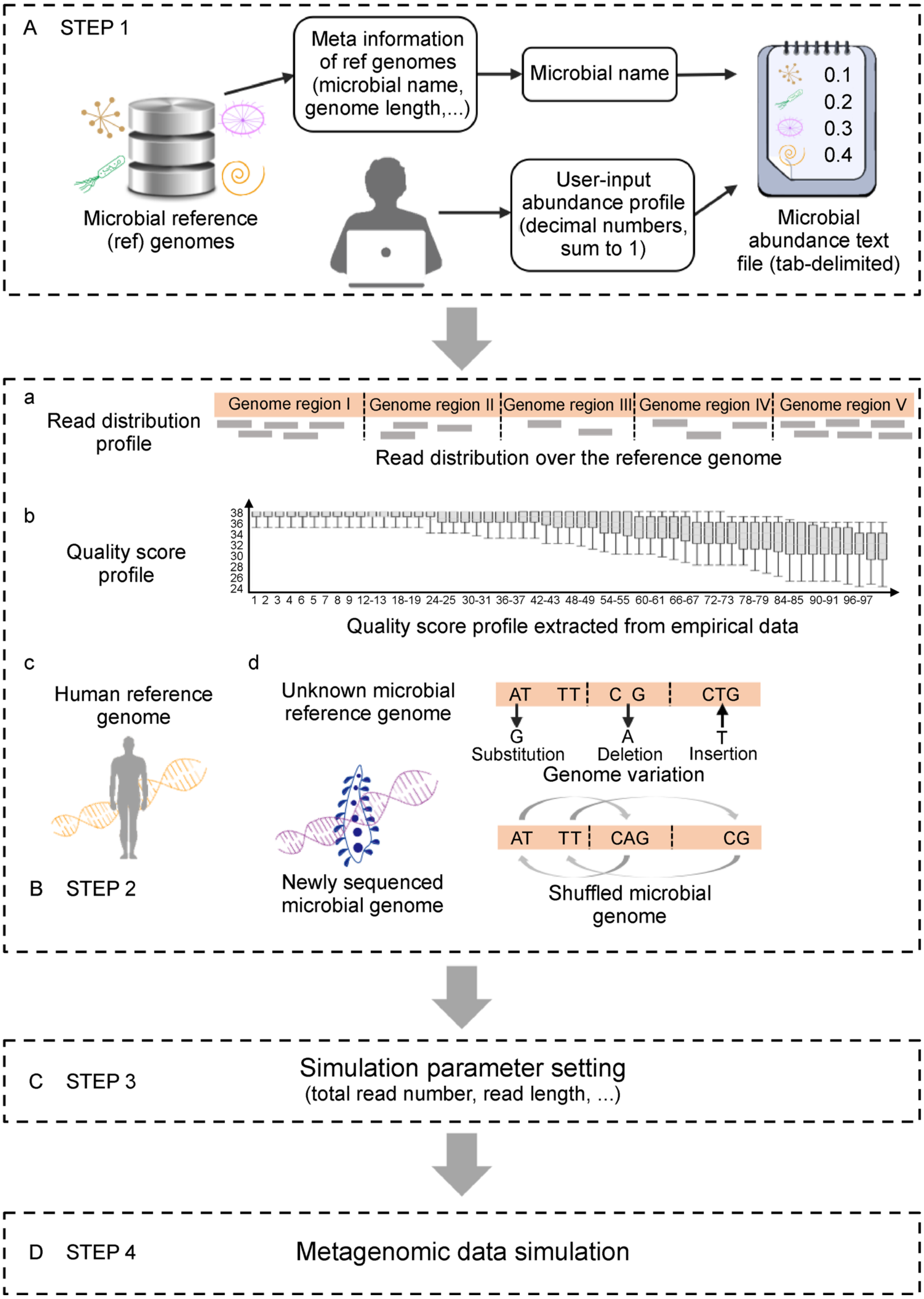
A STEP 1
Microbial reference (ref) genomes
Meta information of ref genomes (microbial name, genome length,...)
Microbial name
User-input abundance profile (decimal numbers, sum to 1)
0.1
0.2
0.3
0.4
Microbial abundance text file (tab-delimited)
a
Read distribution profile
Genome region I
Genome region II
Genome region III
Genome region IV
Genome region V
Read distribution over the reference genome
b
Quality score profile
38 36 34 32 30 28 26 24
1 2 3 4 6 5 7 8 9 12-13 18-19 24-25 30-31 36-37 42-43 48-49 54-55 60-61 66-67 72-73 78-79 84-85 90-91 96-97
Quality score profile extracted from empirical data
c
Human reference genome
d
Unknown microbial reference genome
Newly sequenced microbial genome
AT TT C G CTG
G Substitution
A Deletion
T Insertion
Genome variation
AT TT CAG CG
Shuffled microbial genome
B STEP 2
C STEP 3
Simulation parameter setting
(total read number, read length, ...)
D STEP 4
Metagenomic data simulation

Figure 1. Workflow of nuMetaSim. (A) Step 1: prepare reference genomes and the relative abundance table. (B) Step 2: multiple optional factors that can promote the verisimilitude of the generated data. (C) Step 3: sequencing parameter setting. (D) Step 4: generation of simulated reads.

genomes as components of the simulated metagenome should be provided by the user. The genomes must be in single FASTA format. If a user has a multi-FASTA format genome file, he or she should split it into several single FASTA files, each with a proper prefix and a ".fna" suffix. The script "microbe_index.py" is used to extract meta information like genome length, genome name, etc., from reference genome files. Then the user can define the microbial abundance profile by inputting genome names and their corresponding abundance values with a tab separator. The abundance values represent the relative proportion of each reference genome in the resulting simulated data and their sum should be 1. If not, nuMetaSim will be aborted with error message printing out.

STEP 2 – Optional simulation settings

(i) **Coverage bias:** nuMetaSim applies a new strategy to calculate the GC-content-based coverage bias. Coverage bias means reads are not uniformly sampled from microbial genomes. The trend that more reads be sampled from regions of high GC-content is a major cause of coverage bias [23]. We attempt to establish a relationship between GC-content and number of mapped reads using samples of single-genome sequencing. For the datasets presented in this paper, we collected 20 distinct single-genome sequencing samples and mapped them to their corresponding reference genomes (Table 1). Each reference genome was segmented into bins of equal length. We calculated the GC-content and number of mapped reads of each bin. All the 20 reference genomes did the same operation. Since the sequencing depth of different samples may vary largely, the number of mapped reads to a reference genome need to be normalized by the total number of mapped reads to that genome. Then we assembled a "GC-content-normalized read count" profile of each reference genome to fit a quadratic curve (Figure 2) to represent the GC-content-based coverage bias. The fitted quadratic function can be leveraged to calculate the read distribution table of reference genomes, which are used for generating simulated data by referring to the GC-content of every bin of each component genome. This fitted quadratic function is provided by nuMetaSim as the default fitted function. Users are also allowed to provide more single-genome sequencing data to fit a function that is more appropriate for the data they want to generate. Additionally, nuMetaSim provides a way similar with NeSSM, but uses a variable-length bin to compute read distribution table. A read distribution table can also be completely homemade as long as its format conforms to the standardized one.

(ii) **Quality score profile:** Quality score profiles are extracted from empirical data, more exactly, the FASTQ format sequencing data with ASCII-encoded characters. In nuMetaSim, a tool script "extract_quality_score_distribution.py" is applied for extracting quality score distribution given a sample in the FASTQ format. It supports Illumina 1.3 + (Phred + 64) and Illumina 1.8 + (Phred + 33) encoding schemes [24] and automatically

Table 1 Single-genome sequencing samples and their corresponding reference genomes used to calculate GC-content-based coverage bias

Organism	# of bases (bp)	NCBI SRR ID	Reference genome NCBI GenBank accession ID
Acetohalobium arabaticum DSM 5501 (firmicutes)	483.8 M	SRR3924031	GCA_000144695.1
Acidaminococcus fermentans DSM 20731 (firmicutes)	485.7 M	SRR4240070	GCA_000025305.1
Acidovorax avenae subsp. *avenae* ATCC 19860 (b-proteobacteria)	304.7 M	SRR3925720	GCA_000176855.2
Actinobacillus succinogenes 130Z (g-proteobacteria)	676.7 M	SRR3923544	GCA_000017245.1
Aequorivita sublithincola DSM 14238 (CFB group bacteria)	506.9 M	SRR402806	GCA_000265385.1
Aerococcus sanguinicola (firmicutes)	444.6 M	SRR3118589	GCA_001543145.1
Aerococcus urinaehominis (firmicutes)	210 M	SRR3139722	GCA_001543245.1
Aerococcus viridans (firmicutes)	561.8 M	SRR3118633	GCA_001543285.1
Alistipes finegoldii DSM 17242 (CFB group bacteria)	467.7 M	SRR3924066	GCA_000265365.1
Aminobacterium colombiense DSM 12261 (bacteria)	504.6 M	SRR3924067	GCA_000025885.1
Aminomonas paucivorans DSM 12260 (bacteria)	460.4 M	SRR3924075	GCA_000165795.1
Anabaena cylindrica PCC 7122 (cyanobacteria)	539.5 M	SRR3926593	GCA_000317695.1
Anaerococcus prevotii DSM 20548 (firmicutes)	622 M	SRR3924070	GCA_000024105.1
Archangium gephyra (d-proteobacteria)	1.2 G	SRR4156095	GCA_001027285.1
Arcobacter nitrofigilis DSM 7299 (e-proteobacteria)	530.5 M	SRR4240290	GCA_000092245.1
Bacillus cellulosilyticus DSM 2522 (firmicutes)	495.8 M	SRR3926454	GCA_000177235.2
Bacteroides coprosuis DSM 18011 (CFB group bacteria)	534.1 M	SRR3926627	GCA_000212915.1
Bacteroides helcogenes P 36-108 (CFB group bacteria)	780.9 M	SRR3926630	GCA_000186225.1
Beijerinckia indica subsp. *indica* ATCC 9039 (a-proteobacteria)	257.9 M	SRR4239896	GCA_000019845.1
Belliella baltica DSM 15883 (CFB group bacteria)	540.8 M	SRR3926633	GCA_000265405.1

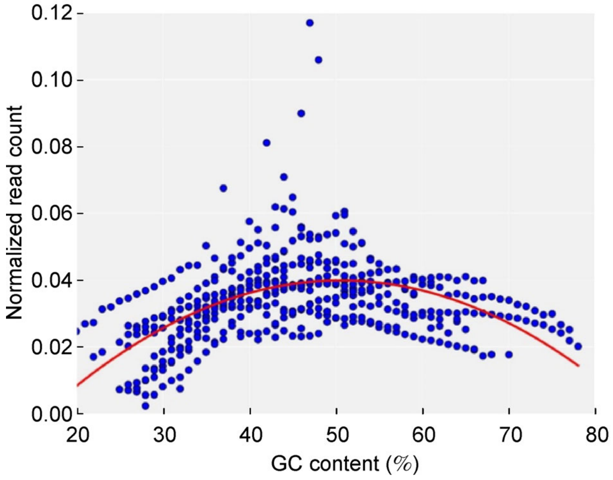

Figure 2. GC-content-normalized read count fitting curve. The *x*-axis is the percentage of GC-content, *y*-axis is the normalized read count, which will be used as the generated read proportion given a specific GC-content in the simulation.

examines the encoding scheme of the input FASTQ file. The script counts the frequency of occurrence of each valid quality score at each base so that every base has a quality score distribution (Figure 1B(b)). The total number of quality score distribution equals to the length of the longest read of the input FASTQ file.

(iii) **Human reference genome preparation:** Reads from the human genome are a main contamination source in metagenomic sequencing data of human microbiome projects [25–27]. In order to make the simulated metagenomic data more real, nuMetaSim incorporates human reads generation as a built-in function to simulate the contamination. We used the multi-FASTA format hg19 reference genome [28] from UCSC genome browser [29] and organized it into a separate folder. This reference genome was divided into 24 single FASTA format chromosomal files, including chromosome 1 to chromosome 22, chromosome X and chromosome Y. Users can also provide human reference genome in single FASTA format by themselves if they want to use a different version.

(iv) **Unknown reference genomes in the simulation:** In most microbiome studies, many reads in the metagenome data are from unknown microbes which cannot be mapped to existing reference genomes [1]. Producing "unknown reads" to mimic those unmapped reads is also included in nuMetaSim as a built-in function. Similar to human reads generation, users can provide reference genomes to mimic unknown genomes in a separate folder. We suggest three ways to prepare unknown reference genomes. The first is to get newly sequenced microbial genomes since newly sequenced microbes usually have not yet been incorporated in the reference databases for the software to be tested. The other way is rearranging the known reference genome to obtain a shuffled genome, similar to the method adopted by Lindgreen *et al.* [7]. Additionally, we can mutate a known reference genome through substitution, insertion and deletion at certain given mutation rate. nuMetaSim offers tool scripts for the latter two methods.

STEP 3 – Simulation sequencing parameter setting

(i) **File path and file name:** Index files produced by "microbe_index.py" and microbial relative abundance table are required. If a user chooses the optional functions described in STEP 2, the read distribution table, quality score profile, human reference genome and unknown reference genomes will be needed. A user should specify the prefix of the file name for the simulated sample. The suffix will be ".fa" if the user chooses to generate FASTA format simulated data and be ".fq" otherwise.

(ii) **Read number and length:** The total number of reads to be generated and the read length is set by the user. nuMetaSim then assigns respective the number of reads for each component genome according to the relative abundance table.

(iii) **Gap length for simulating pair-end data:** nuMetaSim can generate either single-end (SE) sequencing reads or pair-end (PE) sequencing reads according to users' choice. The gap length is expected insertion length between the reads in a pair in the PE sequencing data. The default gap length is set to 200 nt.

(iv) **Bin size:** A bin defined in nuMetaSim is a sliding widow scanning the input reference genomes without overlap. It is the basic local genomic unit used to generate reads. A larger bin size means a lower resolution of read distribution over the reference genome. The bin size will be automatically determined if the user has chosen to produce the read distribution profile with tool scripts provided by nuMetaSim or if the user provides a customized read distribution table.

(v) **Random seed:** Setting a random seed is to make the simulated data repeatable and controllable. Users can use different seeds to generate different simulated samples for each running, or let nuMetaSim use its default seed if there is no need to reproduce the same result in the future.

(vi) **Flags and ratios:** There are several flags and ratios users can set to define the way they want the simulation runs. The variable-random-seed flag (0 or 1) is to control whether nuMetaSim varies the random seed for each new simulation job. The FASTA-or-FASTQ flag controls the output format. Users can also define the sequencing error ratio, human reads ratio and unknown reads ratio. The sequencing error ratio controls the total proportion of substitution and insertion/deletion (indel) of bases in the generated data (note that only FASTQ format output allows sequencing error). If the flag for error ratio is set to 0, no error will be added in the simulation data. If not, there is another flag that controls whether the error will be adjusted according to the quality score in the following way [30]:

$$base\_error\_probablity = 10^{-0.1*quality\_score}$$

If errors are to be generated, nuMetaSim uses an empirical relative proportion of the three types of errors (substitution, insertion or deletion) according to the literature about sequencing error properties of the Illumina sequencing platform [31].

STEP 4 – Simulated data generation

In the data generation procedure, the relative abundance of each input reference genome is adjusted first to acquire the relative proportion of read number contributed by each component genome in the resulting simulated sample. Suppose the total number of component genomes is N, the relative abundance and length of the i-th reference genome is a_i and L_i, respectively, then the relative proportion r_i of read number contributed by the i-th reference genome is:

$$r_i = \frac{a_i * L_i}{\Sigma_{i=1}^{N} a_i * L_i}.$$

Suppose the total number of reads to be generated is *tot*, the read number n_i to be generated by the i-th reference genome is:

$$n_i = tot * r_i = tot * \frac{a_i * L_i}{\Sigma_{i-1}^{N} a_i * L_i}.$$

In practice, the generated read number may have slight discrepancy with the set read number due to rounding off, especially on component genomes with extremely low relative abundances.

Next, nuMetaSim checks whether there exists a user-defined read distribution table. If so, nuMetaSim will load this distribution table and automatically determine the bin size as mentioned before. If not, the user should select a built-in distribution. The default is uniform distribution. Normal and exponential distributions are supported as well. Then the user-defined distribution table or selected built-in distribution is utilized to generate random numbers to calculate sampling points from the reference genome. The sampling points are the origins where the simulated reads are sequenced. If the user has chosen to produce simulated data with sequencing error, errors will be added to the original simulated reads by modifying the bases by substitution and/or indel operations.

Simulated datasets with comprehensive real sequencing data features

Along with the software package, we also generated a few sets of simulation data with different typical settings, based on known genomes and observations from real data.

(i) Source of reference genomes

Reference genomes were selected from a published literature [26]. They are all microbes related with the human intestinal system, including 6 phyla and 205 strain-level genomes (203 bacteria and 2 archaea) (Supplementary Table S1). These microbes all have their complete reference genomes and are well annotated in the supplementary table of Li *et al*'s work [26].

(ii) Design of the simulated dataset

We designed 3 groups of simulated data with 10, 50 and 200 known reference genomes as components, respectively. These datasets mimic microbial communities with distinct compositions [32–35]. Following earlier metagenomic data simulation work [10], we simulated microbial communities with low, medium and high complexity (LC, MC and HC). LC means there is only one dominant microbe with high relative abundance in the community. MC means that two or more microbes are dominant. HC refers to the situation that no microbe is dominant in a HC community [10]. We generated one data set at each level of complexity for each group of simulation. This gave us 9 simulated datasets.

(a) **Relative abundance table:** The relative abundance table was made based on the different levels of complexity and the empirical relative abundance distribution observed in many real data [35–38]. We selected 9 samples of the HMP dataset [34] (Table 2, Figure 3) and used their species abundances as templates of relative abundance tables in our simulated data. For example, the relative abundance of our designed 10-genome LC dataset (10_strain_LC) was derived from SRS064493 of the HMP dataset. The abundance values were ranked in descending order. We picked the top ten abundance values and renormalized them to make the sum equals to one.

(b) **Simulation parameter:** Choosing appropriate parameters is important for improving the verisimilitude of simulated data, especially for factors like sequencing errors caused by the sequencing protocol, and the mixture with human reads and reads from unknown microbes. We designed two levels for each factor to simulate different situations. For sequencing error, we set it as 0.1% and 1% according to observations on error profiles of the Illumina

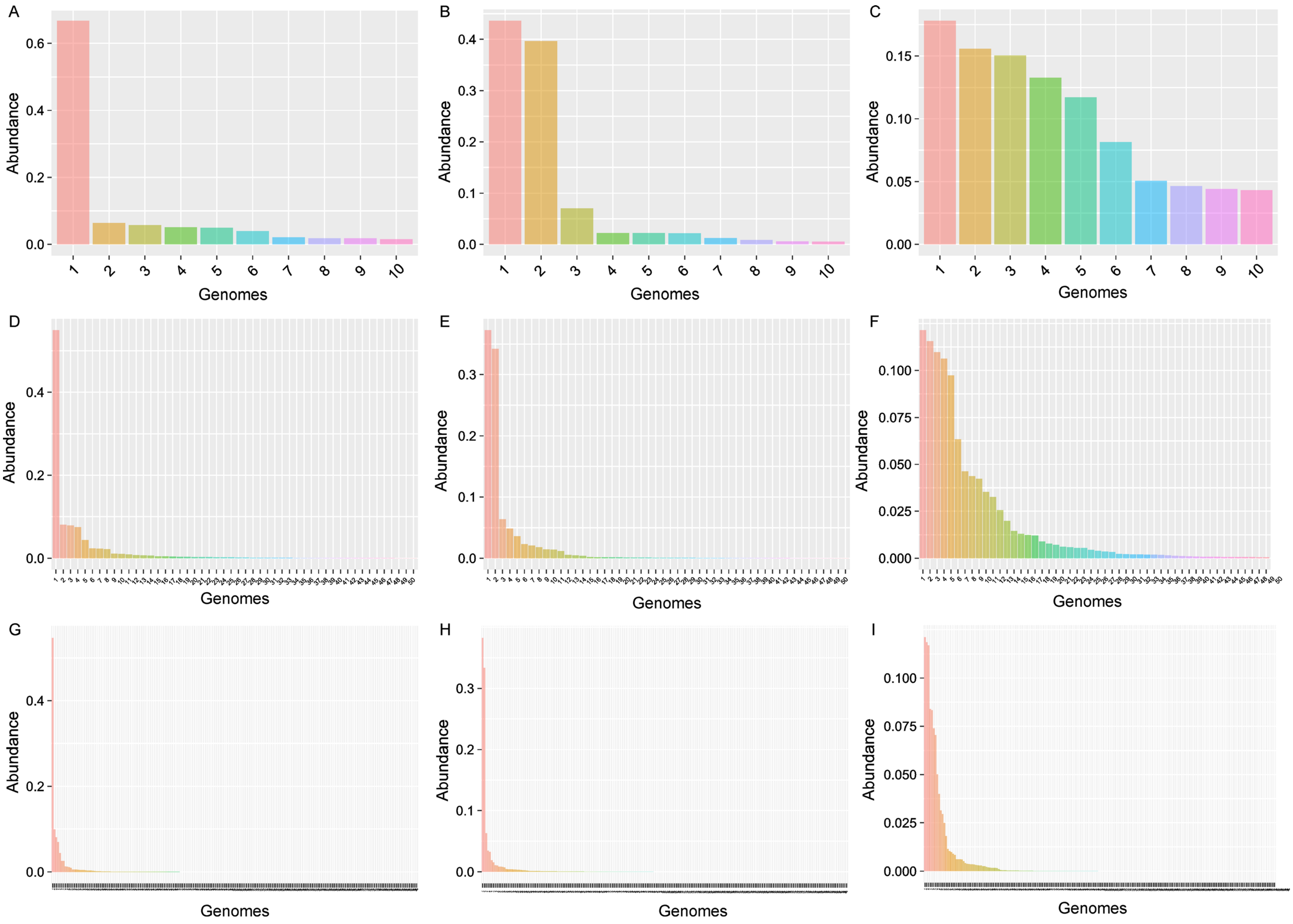

Figure 3. Relative abundance profiles of component genomes of the 9 datasets. The x-axis is the genome name, represented by a number, y-axis is the relative abundance of each component genome. (A) 10_strain_LC. (B) 10_strain_MC. (C) 10_strain_HC. (D) 50_strain_LC. (E) 50_strain_MC. (F) 50_strain_HC. (G) 200_strain_LC. (H) 200_strain_MC. (I) 200_strain_HC.

sequencing platform [31]. We set human reads ratio as 1% and 65%, and unknown reads ratio as 20% and 40%, according to observations on HMP data [24]. We selected some newly sequenced microbial genomes as the unknown reference genomes (Supplementary Table S2). Then the combination of the three parameters resulted in eight samples for each dataset.

As mentioned in the background section, a salient feature observed in real sequencing data is the coverage bias induced by the GC-content. We leveraged the fitted quadratic function calculated from the collected 20 single-genome sequencing samples to compute the GC-content-based coverage bias to be used in the simulation. In addition, we also adopted uniform distribution (no coverage bias) to generate simulated samples as a comparison. Therefore, for each dataset in Table 2, there are 16 simulated samples, resulting in 144 simulated samples in total.

Application example of the simulated dataset

Here we present an example for the application of our simulated data, using the samples in "10_strain" and "50_strain" data sets.

We compared six metagenomic software tools developed for taxonomic profiling on these data. They are FOCUS [39], GOTTCHA [40], MEGAN [41], MetaPhlAn2 [42], MetaPhyler [43] and Taxy [44]. Firstly all the samples were filtered to trim "low quality" bases by VSEARCH [45] (command used: vesearch fastq_filter sample.fq fastqout sample_filtered.fq fastq_minlen 30 fastq_qmax 42 fastq_truncqual 20 fastq_maxns 10) and screened to cleanout human reads via a Perl script [46] (command used: run_contaminant_filter.pl -d hg19_bowtie_index -o output_dir sample_filtered.fq). This step was to mimic the standard procedure of metagenomic data preprocessing. The preprocessed reads are called clean reads. Then we conducted experiments on these clean reads.

We intended to use the same reference database for all the seven software so that the comparison can be equitable. Unfortunately, it is not practical because the reference databases of some of these software are not customizable, such as MetaPhlAn2 and MetaPhyler whose taxonomic profiling strategies are based on marker genes. Users are only allowed to utilize the pre-built maker gene database. Furthermore, establishing a specific database is part of a software's strategy. Therefore, we only used the same customized reference database for FOCUS and MEGAN that allow database customization. This database contained 2,823 bacteria and archaea genomes downloaded from NCBI.

The results of most of these 6 software are the names and relative abundances of the detected microbes, but the output of Megan is count of mapped reads instead of read relative abundances. Using the metrics precision and recall, we found that GOTTCHA performed the best among the six software on both the "10_strain" and "50_strain" datasets. This is only an illustration of one potential use of the simulated data. More systematic experiments with fine-tuning of each software are needed to conduct a benchmarking comparison of the software.

CONLUSION AND DISCUSSION

We presented the detailed simulation process of the software nuMetaSim we developed as well as the design method of a simulated metagenome dataset with comprehensive real data features. For the current stage, there do not exists a standard for comparing the simulated data and real data directly, because the true answer like microbial abundance or gene abundance is unknown. Also, the results obtained by bioinformatics software can only be treated as an approximation of the real data. As a consequence, we do not have a direct approach to certify whether a simulated dataset is close to a real dataset. However, we can make the simulated data realistic by adopting real data features as many as possible. This is the strategy we used for designing the simulated dataset. We provided a set of comprehensive simulated metagenomic

Table 2 Basic information of the simulated data sets

Name of Datasets	Number of dominant microbes	Proportion of dominant microbes in a sample	Referred HMP sample accession number
10_strain_LC	1	66.75%	SRS064493
10_strain_MC	2	83.24%	SRS020386
10_strain_HC	None	N/A	SRS047113
50_strain_LC	1	54.97%	SRS011529
50_strain_MC	2	71.47%	SRS022079
50_strain_HC	None	N/A	SRS064449
200_strain_LC	1	54.69%	SRS023958
200_strain_MC	2	71.62%	SRS017209
200_strain_HC	None	N/A	SRS057539

data with 205 human intestinal microbes. The data represent multiple levels of complexity to reflect different microbiome conditions. Many important features in real human metagenomic samples were considered, such as GC-content-based coverage bias, sequencing errors, human reads and unknown reads. We also provided the nuMetaSim simulator along with the data for free academic use. It can be used to generate customized metagenomic simulation data that have almost all properties of real data but with key factors under the full control of the users.

Very recently, a group of method developers published a consortium work on the critical assessment of metagenome interpretation (CAMI) [16]. It was the summary of a challenge for benchmarking many programs for metagenomic data using a data set generated from about 700 microorganisms and 600 viruses and plasmids. Useful observations and insights have been achieved on the compared metagenomics methods. The focus of that work was on the comparison of methods using the carefully generated benchmark data sets. The work showed the importance of using simulation data for benchmarking methods. Our work presents a software for generating simulated metagenomics data. The software allows users to have more controls on the characteristics of the data they would generate, and thus provides the flexibility for readers to study certain particular aspects of methods. We also generated a few simulation data sets as examples to show how the data mimicking different types of real microbiomes can be simulated. This flexibility enables method developers to design simulation data with special characteristic settings to support the systematical study and comparison of metagenomic data analysis methods from multiple angles.

METHODS

Software availability and usage notes

Code availability

The nuMetaSim is available on GitHub (https://github.com/dadinghh2/nuMetaSim). The main scripts are "nuMetaSim.py" and "nuMetaSim_pe.py". The former is for single-end data simulation, while the latter is for pair-end data simulation. Other tool scripts and their usages are introduced in the "Usage notes" section.

Data records

All the simulated samples generated by nuMetaSim in this project are curated through BIG Data Center, Genome Sequence Archive, Beijing Institute of Genomics (BIGD-GSA). The accession number is GSA PRJCA000415.

Usage notes

(i) **GC-content-based coverage bias**: The GC-content-based coverage bias profile is derived from "cal_genome_distribution_by_GC.py" using two inputs: fitting function and reference genomes. The output are genome names and their corresponding read distributions separated by tabs.

(ii) **Fitting function calculation**: The default fitting function is in the software package. Users are also allowed to obtain a customized fitting function. The samples of single-genome sequencing and their respective corresponding reference genomes should be prepared first. Suppose a sample of single-genome sequencing is called "sample.fq" and its corresponding reference genome is "microbe.fna", then the "sample.fq" should be modified as "microbe.fq". All the samples of single-genome sequencing and reference genomes should do the same operation. Users should use "cal_GC_coverage_relation.py" first to get "GC-content-normalized read count" profile, and then use "polyfit_GC_coverage.py" to compute a fitting function.

(iii) **Generating unknown reference "genomes"**: As mentioned before, nuMetaSim provides tool scripts for obtaining unknown reference genomes apart from using newly sequenced microbes. One of the simpler ways is to use "genome_shuffle.py" to generate shuffled genomes by inputting known reference genomes. Nevertheless, shuffled genomes amount to random sequences that may lose biological meaning. For making more biologically reasonable unknown microbial genomes, another tool script "gen_genome_strain.py" was written for this purpose. It takes a genome to be mutated and a substitution/indel profile produced by "cal_genome_variation.py" as inputs. Mutation rate is set by the user. Please visit GitHub to learn more information.

(iv) **Human and unknown reference sequences preparation**: Users should provide the index files generated by "humanOrUnknown_index.py" and the relative abundance tables for the human and unknown reference sequences. If these files are not given, each human or unknown reference sequence will be assigned an equal relative abundance.

AUTHOR CONTRIBUTIONS

Xuegong Zhang initiated the project and designed the study. Shansong Liu developed the methods and software, and conducted the experiments with help from Kui Hua and Sijie Chen. Shansong Liu and Kui Hua designed the simulation data and Shansong Liu generated the data. Shansong Liu and Xuegong Zhang wrote the manuscript.

ACKNOWLEDGEMENTS

We thank Dr. Hongfei Cui for her comments on the simulation design. This work is partially supported by the National Natural Science Foundation of China (Nos. 61673231 and 61721003).

COMPLIANCE WITH ETHICS GUIDELINES

The authors Shansong Liu, Kui Hua, Sijie Chen and Xuegong Zhang declare that they have no conflict of interests.

This article does not contain any studies with human or animal subjects performed by any of the authors.

REFERENCES

1. Zhang, X., Liu, S., Cui, H. and Chen, T. (2016) Reading the underlying information from massive metagenomic sequencing data. Proc. IEEE, 105, 459–473
2. Raes, J., Foerstner, K. U. and Bork, P. (2007) Get the most out of your metagenome: computational analysis of environmental sequence data. Curr. Opin. Microbiol., 10, 490–498
3. Hamady, M. and Knight, R. (2009) Microbial community profiling for human microbiome projects: tools, techniques, and challenges. Genome Res., 19, 1141–1152
4. Devaraj, S., Hemarajata, P. and Versalovic, J. (2013) The human gut microbiome and body metabolism: implications for obesity and diabetes. Clin. Chem., 59, 617–628
5. Turnbaugh, P. J., Hamady, M., Yatsunenko, T., Cantarel, B. L., Duncan, A., Ley, R. E., Sogin, M. L., Jones, W. J., Roe, B. A., Affourtit, J. P., *et al.* (2009) A core gut microbiome in obese and lean twins. Nature, 457, 480–484
6. Smith, M. I., Yatsunenko, T., Manary, M. J., Trehan, I., Mkakosya, R., Cheng, J., Kau, A. L., Rich, S. S., Concannon, P., Mychaleckyj, J. C., *et al.* (2013) Gut microbiomes of Malawian twin pairs discordant for kwashiorkor. Science, 339, 548–554
7. Lindgreen, S., Adair, K. L. and Gardner, P. P. (2016) An evaluation of the accuracy and speed of metagenome analysis tools. Sci. Rep., 6, 19233
8. Bokulich, N. A., Rideout, J. R., Mercurio, W. G., Shiffer, A., Wolfe, B., Maurice, C. F., Dutton, R. J., Turnbaugh, P. J., Knight, R., Caporaso, J. G. (2016) mockrobiota: a public resource for microbiome bioinformatics benchmarking. mSystems 1, e00062-16
9. Krohn, A., Stevens, B., Robbins-Pianka, A., Belus, M., Allan, G.J., Gehring, C. (2016) Optimization of 16S amplicon analysis using mock communities: implications for estimating community diversity. PeerJ Preprints
10. Mavromatis, K., Ivanova, N., Barry, K., Shapiro, H., Goltsman, E., McHardy, A. C., Rigoutsos, I., Salamov, A., Korzeniewski, F., Land, M., *et al.* (2007) Use of simulated data sets to evaluate the fidelity of metagenomic processing methods. Nat. Methods, 4, 495–500
11. Parks, D. H., Imelfort, M., Skennerton, C. T., Hugenholtz, P. and Tyson, G. W. (2015) CheckM: assessing the quality of microbial genomes recovered from isolates, single cells, and metagenomes. Genome Res., 25, 1043–1055
12. Peabody, M. A., Van Rossum, T., Lo, R. and Brinkman, F. S. (2015) Evaluation of shotgun metagenomics sequence classification methods using *in silico* and *in vitro* simulated communities. BMC Bioinformatics, 16, 362
13. Zhou, Q., Su, X. and Ning, K. (2014) Assessment of quality control approaches for metagenomic data analysis. Sci. Rep., 4, 6957
14. Mende, D. R., Waller, A. S., Sunagawa, S., Järvelin, A. I., Chan, M. M., Arumugam, M., Raes, J. and Bork, P. (2012) Assessment of metagenomic assembly using simulated next generation sequencing data. PLoS One, 7, e31386
15. Randle-Boggis, R. J., Helgason, T., Sapp, M. and Ashton, P. D. (2016) Evaluating techniques for metagenome annotation using simulated sequence data. FEMS Microbiol. Ecol., 92, fiw095
16. 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 computational metagenomics software. Nat. Methods, 14, 1063–1071
17. Escalona, M., Rocha, S. and Posada, D. (2016) A comparison of tools for the simulation of genomic next-generation sequencing data. Nat. Rev. Genet., 17, 459–469
18. Richter, D. C., Ott, F., Auch, A. F., Schmid, R. and Huson, D. H. (2008) MetaSim: a sequencing simulator for genomics and metagenomics. PLoS One, 3, e3373
19. 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, e75448
20. 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
21. Shcherbina, A. (2014) FASTQSim: platform-independent data characterization and *in silico* read generation for NGS datasets. BMC Res. Notes, 7, 533
22. Aird, D., Ross, M. G., Chen, W. S., Danielsson, M., Fennell, T., Russ, C., Jaffe, D. B., Nusbaum, C. and Gnirke, A. (2011) Analyzing and minimizing PCR amplification bias in Illumina sequencing libraries. Genome Biol., 12, R18
23. Dohm, J. C., Lottaz, C., Borodina, T. and Himmelbauer, H. (2008) Substantial biases in ultra-short read data sets from high-throughput DNA sequencing. Nucleic Acids Res., 36, e105
24. Cock, P. J., Fields, C. J., Goto, N., Heuer, M. L. and Rice, P. M. (2010) The Sanger FASTQ file format for sequences with quality scores, and the Solexa/Illumina FASTQ variants. Nucleic Acids Res., 38, 1767–1771
25. Methé, B. A., Nelson, K. E., Pop, M., Creasy, H. H., Giglio, M. G., Huttenhower, C., Gevers, D., Petrosino, J. F., Abubucker, S., Badger, J. H., *et al.* (2012) A framework for human microbiome research. Nature, 486, 215–221
26. Li, J., Jia, H., Cai, X., Zhong, H., Feng, Q., Sunagawa, S., Arumugam, M., Kultima, J. R., Prifti, E., Nielsen, T., *et al.* (2014) An integrated catalog of reference genes in the human gut microbiome. Nat. Biotechnol., 32, 834–841
27. Arumugam, M., Raes, J., Pelletier, E., Le Paslier, D., Yamada, T., Mende, D. R., Fernandes, G. R., Tap, J., Bruls, T., Batto, J. M., *et al.* (2011) Enterotypes of the human gut microbiome. Nature, 473,

174–180

28. Meyer, L. R., Zweig, A. S., Hinrichs, A. S., Karolchik, D., Kuhn, R. M., Wong, M., Sloan, C. A., Rosenbloom, K. R., Roe, G., Rhead, B., *et al.* (2013) The UCSC Genome Browser database: extensions and updates 2013. Nucleic Acids Res., 41, D64–D69
29. Karolchik, D., Baertsch, R., Diekhans, M., Furey, T. S., Hinrichs, A., Lu, Y. T., Roskin, K. M., Schwartz, M., Sugnet, C. W., Thomas, D. J., *et al.* (2003) The UCSC genome browser database. Nucleic Acids Res., 31, 51–54
30. Ewing, B. and Green, P. (1998) Base-calling of automated sequencer traces using *phred.* II. Error probabilities. Genome Res., 8, 186–194
31. Schirmer, M., D'Amore, R., Ijaz, U. Z., Hall, N. and Quince, C. (2016) Illumina error profiles: resolving fine-scale variation in metagenomic sequencing data. BMC Bioinformatics, 17, 125
32. Moyer, C. L., Dobbs, F. C. and Karl, D. M. (1994) Estimation of diversity and community structure through restriction fragment length polymorphism distribution analysis of bacterial 16S rRNA genes from a microbial mat at an active, hydrothermal vent system, Loihi Seamount, Hawaii. Appl. Environ. Microbiol., 60, 871–879
33. Wagner, M. and Loy, A. (2002) Bacterial community composition and function in sewage treatment systems. Curr. Opin. Biotechnol., 13, 218–227
34. Turnbaugh, P. J., Ley, R. E., Hamady, M., Fraser-Liggett, C. M., Knight, R. and Gordon, J. I. (2007) The human microbiome project. Nature, 449, 804–810
35. Ulrich, W. and Ollik, M. (2004) Frequent and occasional species and the shape of relative-abundance distributions. Divers. Distrib., 10, 263–269
36. Hong, S. H., Bunge, J., Jeon, S. O. and Epstein, S. S. (2006) Predicting microbial species richness. Proc. Natl. Acad. Sci. USA, 103, 117–122
37. Unterseher, M., Jumpponen, A., Opik, M., Tedersoo, L., Moora, M., Dormann, C. F. and Schnittler, M. (2011) Species abundance distributions and richness estimations in fungal metagenomics–lessons learned from community ecology. Mol. Ecol., 20, 275–285
38. Yang, Y., Chen, N. and Chen, T. (2017) Inference of environmental factor-microbe and microbe-microbe associations from metagenomic data using a hierarchical Bayesian statistical model. Cell Syst., 4, 129–137
39. 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
40. Freitas, T. A. K., Li, P. E., Scholz, M. B. and Chain, P. S. (2015) Accurate read-based metagenome characterization using a hierarchical suite of unique signatures. Nucleic Acids Res., 43, e69
41. Huson, D. H., Auch, A. F., Qi, J. and Schuster, S. C. (2007) MEGAN analysis of metagenomic data. Genome Res., 17, 377–386
42. 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
43. Liu, B., Gibbons, T., Ghodsi, M. and Pop, M. (2010) MetaPhyler: taxonomic profiling for metagenomic sequences. In Bioinformatics and Biomedicine (BIBM), 2010 IEEE International Conference on IEEE, pp. 95–100
44. Meinicke, P., Asshauer, K. P. and Lingner, T. (2011) Mixture models for analysis of the taxonomic composition of metagenomes. Bioinformatics, 27, 1618–1624
45. Rognes, T., Flouri, T., Nichols, B., Quince, C. and Mahé, F. (2016) VSEARCH: a versatile open source tool for metagenomics. PeerJ, 4, e2584
46. Comeau, A. M., Douglas, G. M., & Langille, M. G. (2017) Microbiome Helper: a custom and streamlined workflow for microbiome research. mSystems 2, e00127–16