Bioinformatics, 2016, 1–8
doi: 10.1093/bioinformatics/btw312
Advance Access Publication Date: 3 June 2016
Original Paper

OXFORD

Sequence analysis

MetaFast: fast reference-free graph-based comparison of shotgun metagenomic data

Vladimir I. Ulyantsev[1,*], Sergey V. Kazakov[1], Veronika B. Dubinkina[2,3], Alexander V. Tyakht[2,3] and Dmitry G. Alexeev[3]

[1]ITMO University, Saint Petersburg, Russian Federation, [2]Federal Research and Clinical Centre of Physical-Chemical Medicine, Moscow, Russian Federation and [3]Moscow Institute of Physics and Technology (State University), Dolgoprudny, Russian Federation

*To whom correspondence should be addressed.

Associate Editor: John Hancock

Received on November 5, 2015; revised on May 14, 2016; accepted on May 16, 2106

Abstract

Motivation: High-throughput metagenomic sequencing has revolutionized our view on the structure and metabolic potential of microbial communities. However, analysis of metagenomic composition is often complicated by the high complexity of the community and the lack of related reference genomic sequences. As a start point for comparative metagenomic analysis, the researchers require efficient means for assessing pairwise similarity of the metagenomes (beta-diversity). A number of approaches were used to address this task, however, most of them have inherent disadvantages that limit their scope of applicability. For instance, the reference-based methods poorly perform on metagenomes from previously unstudied niches, while composition-based methods appear to be too abstract for straightforward interpretation and do not allow to identify the differentially abundant features.

Results: We developed MetaFast, an approach that allows to represent a shotgun metagenome from an arbitrary environment as a modified de Bruijn graph consisting of simplified components. For multiple metagenomes, the resulting representation is used to obtain a pairwise similarity matrix. The dimensional structure of the metagenomic components preserved in our algorithm reflects the inherent subspecies-level diversity of microbiota. The method is computationally efficient and especially promising for an analysis of metagenomes from novel environmental niches.

Availability and Implementation: Source code and binaries are freely available for download at https://github.com/ctlab/metafast. The code is written in Java and is platform independent (tested on Linux and Windows x86_64).

Contact: ulyantsev@rain.ifmo.ru

Supplementary information: Supplementary data are available at *Bioinformatics* online.

1 Introduction

Recently computational life scientists have witnessed an astounding increase in the volume of available shotgun metagenomic datasets. The challenge of reducing dimensionality of the data analysis is of primary demand for statistical analysis of metagenomes. This includes taxonomic and functional profiling, assessing richness and similarity. Technological advances and cost reduction of high-throughput sequencing allow to examine microbiota from previously unexplored ecological niches. The degree of detail has increased in an unprecedented way: the average coverage depth has increased by several orders of magnitude since the first metagenomic studies (Venter *et al.*, 2004). Vast genomic data obtained in studies of microbial isolates served as a cornerstone for developing reference-based approaches. Subsequently, there was a boom of such methods based on more

elaborate techniques than direct alignment of each read with a reference genome, e.g. Kraken (Wood and Salzberg, 2014), CLARK (Ounit *et al.*, 2015), FOCUS (Silva *et al.*, 2014), MetaPhlAn2 (Truong *et al.*, 2015). However, a real challenge for reference-based methods is represented by the communities from novel unexplored niches that contain a large fraction of uncultured bacteria. Accordingly, there is a lack of representative genomes for many clades of microbes and viruses that could serve as a reference. This problem is of significant weight even for the environments that have been thoroughly studied for decades: e.g. human gut microbiota where unknown genomes form a lion's share of the total DNA reads (Nielsen *et al.*, 2014).

One of the approaches for measuring metagenomic similarity that was developed in order to cope with the rapidly accumulating volume of data is based on an adaptive subsampling (Shamsaddini *et al.*, 2014). Another one is an alignment-free approach that appears to be attractive to metagenomic researchers due to the sparseness of available reference genome sets. Among such methods there are abstract composition-based methods (*k*-mer spectrum analysis (Chatterji *et al.*, 2008; Dubinkina *et al.*, 2016; Silva *et al.*, 2014; Wu and Ye, 2011; Wu *et al.*, 2016; Vinga and Almeida, 2003), neural networks (Rasheed and Rangwala, 2012), Markov models (Song *et al.*, 2014)) that are computationally efficient and can be run in parallel sections. However, there are certain limitations of these methods: the differentially abundant features between two or more groups of metagenomes turn out to be concealed within the method or provide little information.

An alternative idea for assessing similarity is a *de novo* assembly of the metagenomes (similar to the process applied to individual genomes) followed by an analysis of the yielded contigs (classification, differential abundance analysis based on coverage depth). Here each individual feature is meaningful; however, the assembly is complicated due to a wide range of typical abundance of bacterial species and significant intra-species genomic variability. Special algorithms intended for metagenomic assembly have been developed that address these issues (Boisvert *et al.*, 2012; Namiki *et al.*, 2012; Peng *et al.*, 2011; Treangen *et al.*, 2013). Particularly, combined assembly of metagenomic reads was proposed for estimating pairwise similarity (crAss) (Dutilh *et al.*, 2012). However, complete assembly from reads to contigs is computationally- and memory-intense, especially due to the rapid increase of publicly available metagenomic data.

We have developed the MetaFast algorithm for compact representation of metagenomes using an adaptive segmentation of metagenomic de Bruijn graph, essentially based on a simplified metagenomic *de novo* assembly. Our method lies between the *k*-mer spectrum analysis and assembly and combines the best of these two alignment-free approaches: the speed of the former with the precision of the latter. Its independence of the reference allows to perform efficiently for both extensively studied and novel microbiota types. The performance of MetaFast was compared with several taxonomic profilers (Kraken, CLARK, FOCUS, MetaPhlAn2), as well as with a cross-assembly-based algorithm crAss on simulated data and real metagenomes of gut microbiota, New-York subway and viruses of lake water. Comparative analysis showed that MetaFast is highly efficient and the results of its work are in agreement with the existing methods.

2 Algorithm

Briefly, MetaFast consists of partial *de novo* assembly of each metagenome yielding pseudocontigs (or unitigs (Myers *et al.*, 2000)), subsequent merging of these pseudocontigs from all compared metagenomes into a single graph and segmentation of this graph into components (Fig. 1). Coverage depth values of individual components by the reads of each metagenome provide its feature vector and allow to assess pairwise dissimilarity of the metagenomes. The method is less memory- and CPU time-consuming than ordinary *de novo* assembly. Moreover, contrary to the abstract approaches (based on *k*-mer spectrum, etc.), the yielded features are informative and can be 'flattened' to yield a long consensus DNA sequence.

MetaFast is a pipeline accepting multiple metagenomes as input. The pipeline includes six modules:

1. **KmerCounter.** Extracting high-quality *k*-mers for each metagenome.
2. **SequenceBuilder.** De Bruijn graph's constructing for each metagenome, the extracting of nonbranching paths.
3. **ComponentCutter.** The merging of nonbranching paths from all individual metagenomes into a single de Bruijn graph, extraction of the appropriate components from the graph.

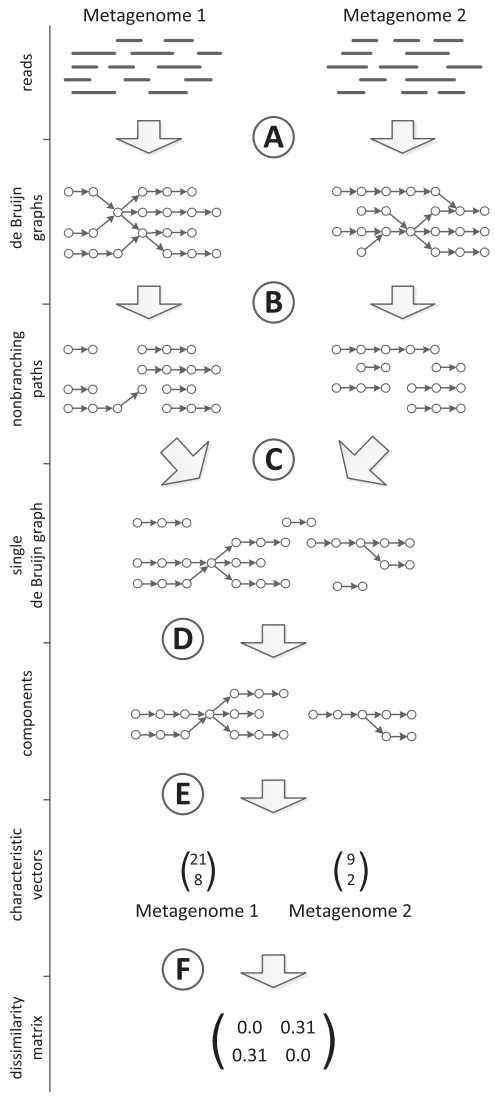

Fig. 1. Workflow of the MetaFast algorithm. The input is a set of metagenomes, the output is a pairwise dissimilarity matrix. The workflow consists of: (**A**) constructing a de Bruijn graph, (**B**) extracting of nonbranching paths, (**C**) Merging into a single de Bruijn graph, (**D**) extracting of the components of appropriate size, (**E**) feature vector computation for each metagenome, (**F**) calculation of the pairwise dissimilarity matrix

4. **FeaturesCalculator.** The calculation of feature vector for each metagenome basing on the coverage depth of the components.
5. **DistanceMatrixCalculator.** The calculation of pairwise dissimilarity matrix between the metagenomes based on the obtained feature vectors.
6. **HeatMapMaker.** Drawing a heatmap between metagenomes based on the calculated dissimilarity matrix, performing clustering analysis and drawing a dendrogram.

2.1 Counting *k*-mers

Preprocessing of metagenomic data is an important step, especially for MetaFast where the *k*-mers originating from erroneous reads will deteriorate the pseudo-assembly. The users are recommended to perform a read quality check, filtering and/or trimming using specialized software described elsewhere (e.g. Pabinger *et al.*, 2014). MetaFast supposes that the input data is already prefiltered; as an internal filtering, it discards reads including N's. However, sometimes sequencing runs produce the majority of reads with N's; therefore, to control the amount of data involved in the analysis it is necessary to check MetaFast read filtering statistics manually to determine if it is necessary to perform initial trimming with a third-party tool.

According to the studies dedicated to *k*-mer binning methods, sequencing errors as well as low-abundance bacterial species lead to the introduction of 'noisy' rare *k*-mers that impair results of the analysis (Wang *et al.*, 2012; Wu and Ye, 2011). For MetaFast, they cause unnecessary 'bubbles' and increase processing time. There is no universal value for *k*-mer frequency threshold, it depends on the properties of specific datasets, e.g. coverage, and can be adjusted at the discretion of the researcher. For high coverage data, this value can be increased if it is desirable to analyze only the *k*-mer features originating from the major microbial species. Under the default threshold, the majority of the *k*-mers caused by the sequencing errors is discarded (because the probability that a sequencer produces more than one read with the same nucleotide error at the same reference position is low), while the *k*-mers originating in the low-abundance bacterial species are maximally preserved.

The module **KmerCounter** performs calculation of *k*-mer spectrum of reads for each metagenome (the default *k*-mer size is 31). The accumulated *k*-mers form vertices of the de Bruijn graph. Computational complexity of the stage is $O(L)$, where L is the total length of all reads.

2.2 Extraction of nonbranching paths

For each metagenome, the **SequenceBuilder** module constructs the de Bruijn graph from the precomputed *k*-mers, afterwards it performs search for the nonbranching paths and preserves only the long paths (longer than 100 bp by default). This step is basically a simplified *de novo* assembly: its goal is to obtain semi-complete contigs, so it takes little computational time. Computational complexity of the stage is $O(N)$, where N is the number of vertices in the de Bruijn graph (equal to the number of the unique *k*-mers). This theoretical estimate is the lowest possible estimate for the processing of the de Bruijn graph containing N vertices; more complex algorithms typically require more operations (e.g. Velvet assembler (Zerbino and Birney, 2008) takes $O(N\log N)$). The preprocessed data obtained using a third-party software (e.g. by khmer (Crusoe *et al.*, 2015)) can be input into this module (an example is described in the Section 4 below).

2.3 Components extraction

The **ComponentCutter** module combines nonbranching paths from all the analyzed metagenomes into the single de Bruijn graph. Further the graph is divided into components of a size within a fixed range ($b_1 \leq$ size $\leq b_2$; default values: $b_1 = 1000$ *k*-mers, $b_2 = 10000$ *k*-mers—roughly encompassing the distribution of bacterial gene sequence length). Each component is subsequently used to generate a single feature, each feature is used to evaluate all samples. The components that are smaller than b_1 are discarded. The components that are larger than b_2 are used to extract their 'core' part by discarding the *k*-mers occurring in a low number of metagenomes. More formally, a threshold f is calculated such that the *k*-mers occurring in no less than in f metagenomes form a connected component of the required size. Upper limit of computational complexity of the stage is $O(M \cdot N)$, where N is the number of vertices in the de Bruijn graph, M—the number of metagenomes (and at the same time the highest possible value for f). In practice, the major part of the large components is divided into components of the appropriate size even for small values of f.

2.4 Construction of metagenomic feature vectors and pairwise dissimilarity matrix

The **FeaturesCalculator** module calculates feature vectors for each metagenome. A value for a single feature (component) of a given metagenome is defined as the total number of times all the *k*-mers belonging to the component occur in this metagenome. Importantly, this step saves the relative abundance information so there is no need to retrieve it at the further steps of the analysis (e.g. via computationally intense remapping of the reads and coverage normalization). The individual features are concatenated to form a feature vector. *k*-mer spectrum computed in the **KmerCounter** module is used; thus it helps to avoid the computationally intense mapping procedure.

Further, the **DistanceMatrixCalculator** module calculates pairwise dissimilarity matrix between the metagenomes using the feature vectors basing on Bray–Curtis dissimilarity measure. The Bray–Curtis dissimilarity $\mathrm{BC}(x, y)$ between metagenomes x and y basing on the normalized abundances of features $x_1..x_N$ and $y_1..y_N$ for the two metagenomes was calculated as:

$$\mathrm{BC}(x, y) = 1 - 2\frac{\sum_{i=1}^{N} \min(x_i, y_i)}{\sum_{i=1}^{N} (x_i + y_i)},$$

where N is the number of features; $\mathrm{BC} = 0$, if the abundance values are equal for each feature in the metagenomes, and $\mathrm{BC} = 1$, if no common features are present in the metagenomes.

2.5 Pipeline output

The MetaFast output includes the following results:

- *k*-mer frequency statistics for each metagenome;
- nonbranching paths for each metagenome;
- extracted components and component statistics file;
- feature vector for each metagenome;
- pairwise dissimilarity matrix;
- dissimilarity heatmap with a dendrogram (Fig. 2A).

2.6 Implementation details

The software is implemented in Java, thus it can be run on any operating system with installed Java 1.6 or higher. It can read compressed and uncompressed FASTA and FASTQ files. A user can set *k*-mer size (up to $k = 31$), minimum and maximum component sizes and other parameters. User can also specify memory and processors

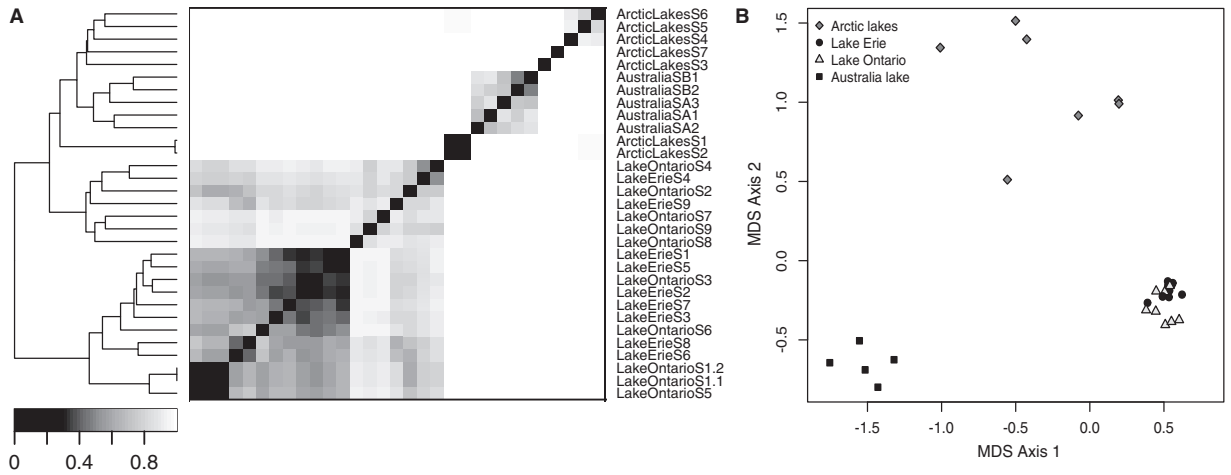

Fig. 2. (**A**) Example of MetaFast graphical output—a dissimilarity heatmap with dendrogram for lake viromes dataset. Colors in heatmap range from black to white, where black corresponds to high similarity between the samples and white—to low. The hierarchical clustering is performed using average linkage and Bray–Curtis metric. Similarity was measured by Spearman correlation. (**B**) Multi-dimensional scaling plot for MetaFast distance matrix for lake viromes dataset

Table 1. Metagenomic datasets used for testing MetaFast algorithm

Dataset	# metagenomes	# reads per metagenome (mean ± SD), mln	Technology (read length, bp)
Simulated human gut microbiota	100	1	N/A (100)
Real human gut microbiota (Qin *et al.*, 2012)	157	47 ±11	Illumina HiSeq (90)
New-York subway microbiota (Afshinnekoo et al., 2015)	29	2.3 ±0.6	Illumina MiSeq (300)
Lake viromes (Emerson et al., 2013; Mohiuddin and Schellhorn, 2015; de Cárcer *et al.*, 2015)	31	5 ±4	Illumina HiSeq (100–150)

usage limitations. The software allows to run the whole process (all modules in the pipeline), as well as any single module separately. The implementation partially uses the itmo-assembler library (http://genome.ifmo.ru/en/assembler).

3 Datasets

MetaFast was tested on several simulated and real metagenomic datasets listed in Table 1.

The simulated gut metagenomic dataset was generated from 10 genomes of distantly related major bacterial species prevalent in human gut (Supplementary Table S1). Relative abundance values (Supplementary Table S2) were randomly generated under normal distribution with mean and standard deviation values estimated from taxonomic composition of previously published real gut metagenomes (Qin *et al.*, 2012). Basing on the abundance values, a total of one million of 100 bp reads without errors per each metagenome were generated using MetaSim software (Richter *et al.*, 2008) (parameters: `cmd -r <number_of_reads> -f100 -t0 -seed 7`); totally 100 metagenomes with varying taxonomic composition were generated (deposited at http://download.ripcm.com/Ulyantsev_2016_suppl/).

The real gut metagenomic dataset was a large group of gut metagenomes from Chinese population previously assessed by us from both taxonomic and functional perspectives as described previously (Tyakht *et al.*, 2013).

The subway dataset contained all metagenomes from the New-York subway sequencing project (Afshinnekoo *et al.*, 2015) that had read length 300 bp: samples were taken from different surfaces (e.g. turnstile, bench, hand rail, see Supplementary Table S3). Finally, the lake virome dataset included metagenomes of viral fraction of microbiota combined from three projects dedicated to analysis of water in Arctic lakes (see Supplementary Table S4): Lake Linnevatnet, Pond Borgdammane, Lake Tunsjøen, Lake Nordammen, Lake Tenndammen (de Cárcer *et al.*, 2015); hypersaline Lake Tyrrell, Victoria, Australia (Emerson *et al.*, 2013) and Lakes Erie and Ontario (Mohiuddin and Schellhorn, 2015).

The performance of our algorithm on the datasets was compared with the existing tools: Kraken, CLARK, FOCUS, MetaPhlAn2, crAss (initial pre-assembly for crAss was performed using Newbler v2.6, Roche, Switzerland). The tools were run using the default parameters according to the respective user manuals. Databases for Kraken and CLARK were built from NCBI/RefSeq database (downloaded on December 08, 2015).

4 Results

Microbial samples originating in the same type of environment (e.g. gut of the same host species) or from the different time points of the same environment typically contain a significant amount of common bacterial species—and their genes. The abundance of such shared genomic sequences can be analyzed via various algorithms, including those based on the elements of combined assembly of metagenomes, particularly MetaFast. When many prevalent components are common for two samples, it reflects their similarity; while the low number and lack of such components reflect the fact that the communities are quite dissimilar.

In order to perform a comprehensive evaluation of usability of the MetaFast algorithm, we examined the following essential characteristics in dedicated computational experiments: (i) accuracy; (ii) speed; (iii) memory consumption; (iv) dependence on the reference set.

Regarding accuracy, on simple data our method should produce the results that are in agreement with the results obtained on the

Table 2. Detailed comparison of computational time and memory usage for MetaFast and other dissimilarity analysis methods using the NY subway microbiota dataset

Algorithm	MetaFast	FOCUS	Kraken	CLARK	MetaPhlAn2	crAss
Processing time, min	82	33	33	16	89	75 (3190)
RAM usage, Gb	14.0	5.1	76.5	107.4	3.3	18.4 (70.8)
Reference database	N/A	2766	NCBI/RefSeq	NCBI/RefSeq	~17 000	N/A
k-mer size	31	6–8	31	21–31	255	N/A
% reads	89 ± 6	100	61 ± 13	43 ± 15	N/A	N/A

The performance of algorithms was compared on a 48-core compute node (4× Opteron 6176 12 cores 2.3 GHz) with 256 Gb RAM. FOCUS database consists of the complete genomes from the SEED servers. NCBI/RefSeq database includes bacterial, archaeal, viral and human genomes. MetaPhlAn2 uses 1 mln. clade-specific marker genes (~17 000 reference genomes). crAss was used in conjunction with Newbler assembler, its processing time and RAM usage are shown in brackets. '% reads' is the mean fraction of the reads included into the analysis.

same data using a traditional approach (i.e. based on the mapping to the reference set). Therefore, as a basic check of MetaFast validity, it was compared with a reference-based method on the metagenomes with an *a priori* known composition. Accordingly, we used simulated gut metagenomes—to assess accuracy (# 1).

We also tested MetaFast on previously unexamined novel microbiota data by including NY subway metagenomes and performing the comparison of our method with the published relevant algorithms—to compare the dependence on the reference set as well as the processing time and memory consumption (## 2, 3, 4).

To demonstrate the usability of our method in large-scale metagenomic studies, we analyzed a large group of real gut metagenomes and assessed speed, accuracy and memory consumption (## 1, 2, 3). The stability of MetaFast-derived dissimilarity values in regard to the size of the group was shown using subsamples of this dataset.

Finally, to deepen the evaluation of the dependence on the reference, we tested MetaFast and other algorithms on a dataset with an even higher fraction of components with unavailable reference sequences for which a reference-based approach is expected to fail—a metagenome of a viral fraction of lake water microbiota (thus evaluating ## 2, 3, 4).

4.1 Simulated gut metagenomes: evaluation of accuracy

We calculated conventionally true dissimilarity matrix (based on the *a priori* known taxonomic composition from the proportions in which the reads of individual species were taken) using Bray–Curtis measure. Another dissimilarity matrix was computed for the simulated metagenomes using MetaFast. The two matrices were found to be highly correlated (Mantel test: Spearman correlation $r = 0.96$, $P = 0.001$, Supplementary Fig. S1A) confirming the general validity of our method; the effect also held true when *k* value was varied (Supplementary Fig. S1B and C).

4.2 New-York subway metagenomes: evaluation of speed, memory consumption and the dependence on reference

NY subway dataset was used to compare MetaFast with several taxonomic composition profiling tools as well as reference-free crAss. The dataset was generated from the samples representing novel previously unexplored microbiota, thus it was used to evaluate MetaFast in comparison with the other algorithms for taxonomic profiling and combined assembly. The results of the comparison are shown in Table 2.

FOCUS was the only tool to surpass MetaFast in terms of both memory economy and running time. MetaPhlAn2 showed time comparable with one of MetaFast and was superior only by the memory economy. Both CLARK and Kraken, declared to be ultrafast tools for taxonomic profiling using *k*-mer spectra, required a much higher memory volume than MetaFast because in order to accelerate the analysis, the tools store both reads and database in the memory.

All the four mentioned third-party tools possess an inherent drawback—the dependence on the reference set. Therefore, only incomplete genomic information is taken into account when the pairwise dissimilarity is calculated. The only approach conceptually similar to MetaFast is crAss that also employs combined assembly for clustering of metagenomes. However, due to the fact that the tool requires the results of standard assembly, crAss analysis uses several times higher amount of time and memory than MetaFast. Moreover, due to the memory requirements only a small number of metagenomes can be processed simultaneously.

In the process of NY subway metagenome analysis, we discovered that the percentage of the reads identified by the reference-based methods was high (Supplementary Table S5). It suggested that, although the environment has not been examined before, its microbiota is to a wide extent dominated by known species (including the typical bacterial inhabitants of human skin).

The taxonomic composition yielded using each of the tools (Supplementary Tables S6–S9) as well as the relative abundance values of contigs from crAss were used to construct pairwise dissimilarity matrices via Bray–Curtis measure. Comparison of the matrices showed that all methods produce generally correlated results (Spearman correlation $r = 0.7 - 0.8$, see Fig. 3). Notably, MetaFast demonstrated higher similarity with taxonomic profiling algorithms than crAss ($r = 0.81 - 0.86$ versus $r = 0.71 - 0.83$). While ground-truth composition of the microbiota is unknown, we regard this observation as an indirect evidence that MetaFast components reflect taxonomic components.

In order to assess the changes in MetaFast performance when the step of pseudo-assembly is replaced with a standard *de novo* assembly, we used assemblers of two different types: SPAdes (based on de Brujin graph) (Bankevich *et al.*, 2012) and Newbler (overlap-layout consensus algorithm) (Roche). The results showed that in both cases, while the time and memory consumption has obviously increased, the correlation of dissimilarity values produced by MetaFast and taxonomic profiling tools slightly decreased (difference of correlation values $dr = 0.01 - 0.04$, see Fig. 3). We speculate that simplified assembly implemented in MetaFast not only reduces the resources consumption but also decreases the negative effect of the metagenomic assembly artefacts (chimeric contigs, etc.) on the results. It could also contribute to the fact that crAss showed lower correlation with the taxonomic methods than MetaFast.

4.3 Gut microbiota metagenome: evaluation of applicability to large datasets

Further, we applied MetaFast to a large metagenomic dataset—using a collection of published human gut metagenomes from

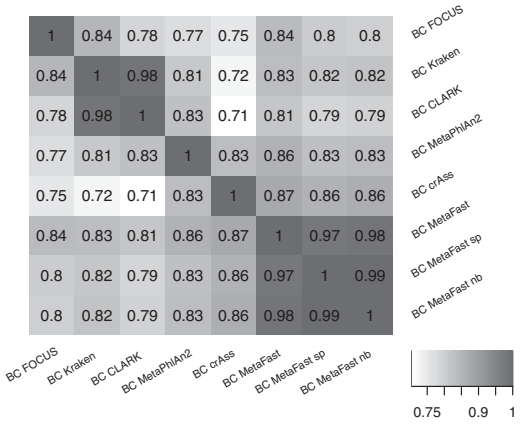

Fig. 3. Spearman correlation between the pairwise dissimilarity matrices obtained using MetaFast and other algorithms. The correlation values were obtained using Mantel test ($P = 0.001$). The label 'MetaFast sp' denotes a modified version of MetaFast analysis when the step of pseudo-assembly is replaced with the assembly using SPAdes, 'MetaFast nb' when it was replaced with the assembly using Newbler

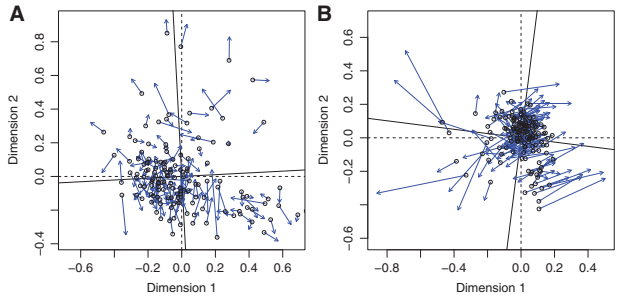

Fig. 4. Procrustes analysis for MDS (multi-dimensional scaling) according to Bray–Curtis index calculated for: (**A**) taxonomic composition versus MetaFast; (**B**) functional composition versus MetaFast. For each metagenome, arrow indicates the direction of coordinate change from MetaFast to reference-based composition

Chinese population. The complete analysis by MetaFast using the default parameters took 34 h on a supercomputer using 90 Gb of memory and 20 CPUs. The two dissimilarity matrices obtained from taxonomic and functional composition using Bray–Curtis index were compared with the MetaFast results using Mantel test. The results across the methods were highly correlated (Spearman correlation values and *P*-value according to Mantel test: $r = 0.91$ for taxonomic mapping, $r = 0.78$ for functional mapping, $P = 0.001$). This fact is illustrated by Procrustes analysis plots (Fig. 4).

The MetaFast dissimilarity between two metagenomes is dependent on the total number of metagenomes involved in the analysis and might significantly fluctuate when it is low. In order to examine this effect, MetaFast was applied to random subsamples of the original group (subsample size: $n = 10$, 20 and 50 metagenomes; 50 times for each size). The results confirmed that the MetaFast dissimilarity converges starting from the number of the analyzed samples around $n = 20$—for the same microbiota type, human gut (Supplementary Fig. S2A).

4.4 Virome dataset: evaluation of application to novel microbiota

We tested MetaFast as well as taxonomic profiling algorithms on a set of 31 metagenomes of viral fraction of lake water—a more challenging material than bacterial metagenomes because of the higher diversity and larger fraction of unknown sequences. The fraction of the unidentified reads is typically high for virome samples (up to 60–90%) because this superkingdom is understudied and few reference sequences are available (Mokili *et al.*, 2012); moreover, the mutation rates in viruses are orders of magnitude higher leading to distant unrecognizable homology (Dutilh *et al.*, 2012). Therefore, traditional methods of comparing the metagenomes basing on read mapping against the reference are poorly applicable to such datasets.

While MetaFast included the totality of the reads into the analysis, CLARK and Kraken could identify only around 1% of the reads—despite the fact that these tools use the complete genomes in NCBI RefSeq for the bacterial, archaeal and viral domains as a reference. MetaPhlAn2 was able to detect certain viral as well as bacterial taxa (see Supplementary Table S10), although a large fraction of the taxa were defined as unclassified at the species level and the inherent nature of the algorithm does not directly allows to measure the fraction of the identified reads.

MetaFast analysis of the viral metagenomes took 45 min and 11.4 Gb of RAM. The pairwise dissimilarity analysis showed that the viral metagenomes clustered perfectly by the origin (Fig. 2). Each of the three environments correspond to the respective study, therefore, one cannot exclude the possibility that such clustering is mainly due to batch effect (the combined influence of viral fraction enrichment, sequencing library preparation protocol, etc. unique for each study). However, the sequencing platform was the same in all studies (Illumina HiSeq) so this factor is supposed to contribute little variation. Moreover, the biological meaning of the clustering results is supported by the fact that the time points from the same lake (as well as the technical replicates) tended to cluster together. The samples from lakes Eerie and Ontario appeared to be mixed—however, it is in agreement with the fact that these are neighbouring freshwater lakes in the same climate and the seasonal variation of community structure for the same lake might be higher than between the lakes.

Noteworthy, the MetaFast dissimilarity matrix is highly correlated with one produced using MetaPhlAn2 ($r = 0.78$), suggesting that although MetaPhlAn2 composition vectors containing features with unclassified annotations, the features identified by the both methods are related. Comparison of the clustering results between the methods showed that although generally the matrices are similar, unlike MetaFast, MetaPhlAn2 tends to be more prone to mixing together the samples from distinct locations that are supposed to be quite different—e.g. Arctic, Australia and the Great lakes (Fig. 2B, Supplementary Fig. S3A and B).

The crAss tool was originally suggested by the authors of the method to be especially suitable for processing metagenomes enriched in viral sequences. We expected that due to the reference-free principle of crAss, its application to the mentioned lake viromes would provide the results comparable to MetaFast. Indeed, the yielded dissimilarity matrix was highly correlated with the MetaFast version ($r = 0.94$, see Supplementary Fig. S4). However, the crAss analysis took a much longer time and higher memory volume: the stage of combined *de novo* assembly alone took about 5 days 15 h (146 Gb of RAM, Newbler), followed by 4 h (47 Gb) taken by the final steps of contigs processing by crAss itself.

5 Discussion

Dramatic rates of accumulation of publicly available metagenomic data can be illustrated by a 4–5 times increase in the number of

datasets in some of the largest online metagenomic resources during the last two years; the sequencing depth also grows rapidly, reaching more than a hundred of terabasepairs per sample (Mitchell *et al.*, 2016; Wilke *et al.*, 2016). Such explosion of metagenomic Big Data makes comparative metagenomic studies an even more challenging area demanding new efficient algorithms (Dubinkina *et al.*, 2016) and visualization approaches (Alexeev *et al.*, 2015). Issues of speed, accuracy and memory efficiency become the key factors for novel approaches in metagenomic data processing, especially when the methods are intended to be applied to the broad spectrum of environmental datasets available. Bearing in mind the special nature of metagenomic datasets, we developed a hybrid algorithm that combines the principles of *de novo* pseudo-assembly with the *k*-mer spectrum analysis allowing to perform computationally- and memory-efficient accurate analysis for a large number of metagenomes in a reference-independent way.

The domain of 'shotgun' metagenomic sequences currently contains two large niches. The first one encompasses projects that target moderate coverage for a large number of monotypic samples, e.g. human gut microbiota consortia (Human Microbiome Project Consortium, 2012; Qin *et al.*, 2010; Rampelli *et al.*, 2015; Sankaranarayanan *et al.*, 2015). The second category includes the projects mostly oriented towards *de novo* assembly of novel genomic sequences basing on high-coverage sequencing of a small number of samples—typically coming from various unique environments (e.g. Edwards *et al.*, 2013; Hawley *et al.*, 2014; Howe *et al.*, 2014). Most metagenomic datasets can be placed in between of these two extremes—by containing a fraction of data from unknown organisms and a fraction of genetic material already contained in the public reference databases. A researcher chooses the approach basing on the understanding of the proportions between those two fractions in the current dataset. Interestingly, even in a seemingly well understood metagenomes, the combination of the accumulated data even from multiple projects allows for the discovery of novel species (Dutilh *et al.*, 2014).

The two described groups of datasets imply the choice of considerably different types of algorithms: alignment against a reference base is a fast and easily parallelized algorithm, while assembly algorithms are hardware-demanding. Typical reference-based approaches imply the selection of a similarity criterion to create a non-redundant catalogue—normally selecting a sequence (of a gene or a genome) from a group having a high percent identity over a high percent of length. This step creates further biases in mapping: the more the difference between the analyzed sequence and the reference, the fewer reads are mapped (Degner *et al.*, 2009). The assembly-based approaches for metagenomic datasets are using the same paradigm producing long reference sequences. For estimating the relative abundance of the sequences in a metagenome, the mapping is used again—this time against the sequences assembled from metagenomes that are considered to be more appropriate for mapping. Although one uses several metagenomic samples to increase the coverage, the intrinsic presence of mutations in bacterial genomes leads to selection of only the one variant of a path in the graph out of all available variants. Therefore, the variants are underrepresented in the final obtained sequence—the genomic diversity is thus 'flattened'. The presented algorithm MetaFast is an intermediate solution allowing to work with the speed of mapping and at the same time to gain benefits of *de novo* assembly—particularly, yielding novel genetic sequences. Moreover, one of the ideas implemented in the feature representation allows to avoid the mapping bias and to use 'unflattened' references.

Technically, MetaFast has several differences from the traditional metagenomic assembly approaches. Firstly, MetaFast does not perform a complete *de novo* assembly (e.g. used in the global human microbiome catalog construction (Qin *et al.*, 2010)), but rather an incomplete version of assembly (pseudo-assembly). This greatly improves the performance of the algorithm in terms of speed. To assess whether the precision is affected negatively, we replaced the pseudo-assembly step with a conventional assembler—the results were highly similar. Second, while Qin *et al.* performed the pairwise alignment of the combined pool of sequences and select a single representative per each cluster basing on high percent identity and alignment length, we do not drop the other similar variants but rather connect all of them into a single subgraph. This allows to capture the genomic details of each metagenome individually and then combine the individual results together for the comparison task. Thirdly, we do not perform the final 'flattening' step, instead preserving the branching nature of the graph components (MetaFast features). Thus, information about the variation of the same species between different samples is not lost and can be used for further analysis of gene variations. This also allows to avoid the mapping biases, as the calculation of feature representation in each separate metagenome is performed via *k*-mer counting for a branched Metafast feature. Finally, in the spirit of metagenome-based research, MetaFast allows to identify the features differentiating the groups and concentrate the researcher's attention on them.

Advantages of our approach in terms of speed, accuracy, memory efficiency and independence of reference base were demonstrated using both simulated and real metagenomes; the performance was compared with a variety of tools widely used in metagenomics. Simulations showed high correlation with the results of read mapping confirming basic accuracy of MetaFast. However, it should be kept in mind that the provided simulated datasets are a primitive model of real data; particularly, genomic polymorphisms and gene content variations are ignored in these simulations. A high degree of correlation was shown in experiments with real data, where the results of MetaFast were compared with two versions of mapping—to a reference genome and gene catalog. High correlation with the results obtained via mapping to a genome catalog reflects the correct functioning of the algorithm: it means that the most part of genetic information is mappable to genomes and is also assembled to yield the features.

Noteworthy, the comparison of MetaFast with the combined assembly demonstrated memory efficiency of MetaFast. While in conventional assembly the memory usage increases approximately linearly with the number of the samples, in MetaFast it tends to achieve saturation at a certain number of samples (e.g. around 20 for gut metagenomes, see Supplementary Fig. S2B) and does not increase further. This fact is likely associated with the effect of **ComponentCutter** module (see Methods) that limits the size of the total graph: as soon as the number of metagenomes is sufficiently high to encompass the major diversity of community structures, an addition of new metagenomes does not increase the size of the graph thus does not demand extra memory. Overall, our approach provides a more economic memory usage accompanied with significant speed improvement at the expense of only slight decrease in accuracy.

Applicability of the approach for a wide range of problems was demonstrated for MetaFast. It has shown good correlation with the adopted methods in human gut microbiome datasets and comparative accuracy in case of novel microbiota types and reference based approaches. Interestingly, the only tool comparable to MetaFast in the terms of applicability to a set of metagenomes without well-described reference set is the crAss algorithm designed specifically to assemble species from multiple metagenomes. The results of the comparative viral profiling showed that MetaFast is an adequate

tool for dissimilarity analysis of novel microbiota metagenomes due to the independence of the reference base that might also contain poorly annotated sequences (unclassified at various levels of taxonomy). Our results showed that on viral datasets it outreaches crAss in the terms of both speed and memory consumption, allowing to work with hundreds of metagenomes.

While obviously even in well studied datasets there is a chance to find novel genes, we suggest MetaFast is a very useful tool for exploratory data analysis. The features and their quantification across metagenomes can be obtained rapidly and the technical implementation allows to work with a high number of metagenomes in a robust manner. The settings help to orient the algorithm towards obtaining the features of desired length—comparable to the typical microbial gene length—making it convenient for primary analysis.

Acknowledgements

We thank Alexey Sergushichev for the help with itmo-assembler software and the reviewers for their constructive comments.

Funding

VIU and SVK were financially supported by the Government of Russian Federation (Grant 074-U01) and JetBrains Research. VBD, AVT and DGA were supported by Russian Scientific Foundation (project #15-14-00066).

Conflict of Interest: none declared.

References

Afshinnekoo,E. *et al*. (2015) Geospatial resolution of human and bacterial diversity with city-scale metagenomics. *Cell Syst*., **1**, 72–87.

Alexeev,D. *et al*. (2015) Bacterial rose garden for metagenomic snp-based phylogeny visualization. *BioData Mining*, **8**, 10.

Bankevich,A. *et al*. (2012) Spades: a new genome assembly algorithm and its applications to single-cell sequencing. *J. Comput. Biol.: J. Comput. Mol. Cell Biol*., **19**, 455–477.

Boisvert,S. *et al*. (2012) Ray Meta: scalable de novo metagenome assembly and profiling. *Genome Biol*., **13**, R122.

Chatterji,S. *et al*. (2008) CompostBin: a DNA composition-based algorithm for binning environmental shotgun reads. In: Martin Vingron, Limsoon Wong editors, *Research in Computational Molecular Biology*. Berlin/Heidelberg: Springer, pp. 17–28.

Crusoe,M. *et al*. (2015) The khmer software package: enabling efficient nucleotide sequence analysis [version 1; referees: 2 approved, 1 approved with reservations]. *F1000Research*, **4**, 900.

de Cárcer,D.A. *et al*. (2015) Biodiversity and distribution of polar freshwater DNA viruses. *Sci. Adv*., **1**, e1400127.

Degner,J.F. *et al*. (2009) Effect of read-mapping biases on detecting allele-specific expression from RNA-sequencing data. *Bioinformatics*, **25**, 3207–3212.

Dubinkina,V.B. *et al*. (2016) Assessment of k-mer spectrum applicability for metagenomic dissimilarity analysis. *BMC Bioinformatics*, **17**, 1–11.

Dutilh,B.E. *et al*. (2012) Reference-independent comparative metagenomics using cross-assembly: crAss. *Bioinformatics*, **28**, 3225–3231.

Dutilh,B.E. *et al*. (2014) A highly abundant bacteriophage discovered in the unknown sequences of human faecal metagenomes. *Nat. Commun*., **5**, 4498.

Edwards,A. *et al*. (2013) A metagenomic snapshot of taxonomic and functional diversity in an alpine glacier cryoconite ecosystem. *Environ. Res. Lett*., **8**, 035003.

Emerson,J.B. *et al*. (2013) New approaches indicate constant viral diversity despite shifts in assemblage structure in an Australian hypersaline lake. *Appl. Environ. Microbiol*., **79**, 6755–6764.

Hawley,E.R. *et al*. (2014) Metagenomes from two microbial consortia associated with Santa Barbara seep oil. *Mar. Genomics*, **18**, 97–99.

Howe,A.C. *et al*. (2014) Tackling soil diversity with the assembly of large, complex metagenomes. *Proc. Natl. Acad. Sci. U. S. A*., **111**, 4904–4909.

Human Microbiome Project Consortium. (2012) Structure, function and diversity of the healthy human microbiome. *Nature*, **486**, 207–214.

Mitchell,A. *et al*. (2016) Ebi metagenomics in 2016 - an expanding and evolving resource for the analysis and archiving of metagenomic data. *Nucleic Acids Res*., **44**, D595–D603.

Mohiuddin,M. and Schellhorn,H.E. (2015) Spatial and temporal dynamics of virus occurrence in two freshwater lakes captured through metagenomic analysis. *Front. Microbiol*., **6**, 960.

Mokili,J.L. *et al*. (2012) Metagenomics and future perspectives in virus discovery. *Curr. Opin. Virol*., **2**, 63–77.

Myers,E.W. *et al*. (2000) A whole-genome assembly of drosophila. *Science*, **287**, 2196–2204.

Namiki,T. *et al*. (2012) MetaVelvet: an extension of Velvet assembler to de novo metagenome assembly from short sequence reads. *Nucleic Acids Res*., **40**, e155.

Nielsen,H.B. *et al*. (2014) Identification and assembly of genomes and genetic elements in complex metagenomic samples without using reference genomes. *Nat. Biotechnol*., **32**, 822–828.

Ounit,R. *et al*. (2015) Clark: fast and accurate classification of metagenomic and genomic sequences using discriminative k-mers. *BMC Genomics*, **16**, 1.

Pabinger,S. *et al*. (2014) A survey of tools for variant analysis of next-generation genome sequencing data. *Brief. Bioinf*., **15**, 256–278.

Peng,Y. *et al*. (2011) Meta-IDBA: a de Novo assembler for metagenomic data. *Bioinformatics*, **27**, i94–i101.

Qin,J. *et al*. (2010) A human gut microbial gene catalogue established by metagenomic sequencing. *Nature*, **464**, 59–65.

Qin,J. *et al*. (2012) A metagenome-wide association study of gut microbiota in type 2 diabetes. *Nature*, **490**, 55–60.

Rampelli,S. *et al*. (2015) Metagenome Sequencing of the Hadza Hunter-Gatherer Gut Microbiota. *Curr. Biol*., **25**, 1682–1693.

Rasheed,Z. and Rangwala,H. (2012) Metagenomic taxonomic classification using extreme learning machines. *J. Bioinf. Comput. Biol*., **10**, 1250015.

Richter,D.C. *et al*. (2008) Metasima sequencing simulator for genomics and metagenomics. *PloS One*, **3**, e3373.

Sankaranarayanan,K. *et al*. (2015) Gut microbiome diversity among Cheyenne and Arapaho individuals from western Oklahoma. *Curr. Biol*., **25**, 3161–3169.

Shamsaddini,A. *et al*. (2014) Census-based rapid and accurate metagenome taxonomic profiling. *BMC Genomics*, **15**, 918.

Silva,G.G. *et al*. (2014) FOCUS: an alignment-free model to identify organisms in metagenomes using non-negative least squares. *PeerJ*, **2**, e425.

Song,K. *et al*. (2014) New developments of alignment-free sequence comparison: measures, statistics and next-generation sequencing. *Brief. Bioinf*., **15**, 343–353.

Treangen,T.J. *et al*. (2013) MetAMOS: a modular and open source metagenomic assembly and analysis pipeline. *Genome Biol*., **14**, R2.

Truong,D.T. *et al*. (2015) Metaphlan2 for enhanced metagenomic taxonomic profiling. *Nat. Methods*, **12**, 902–903.

Tyakht,A.V. *et al*. (2013) Human gut microbiota community structures in urban and rural populations in Russia. *Nat. Commun*., **4**, 2469.

Venter,J.C. *et al*. (2004) Environmental genome shotgun sequencing of the Sargasso Sea. *Science*, **304**, 66–74.

Vinga,S. and Almeida,J. (2003) Alignment-free sequence comparison – a review. *Bioinformatics*, **19**, 513–523.

Wang,Y. *et al*. (2012) Metacluster 5.0: a two-round binning approach for metagenomic data for low-abundance species in a noisy sample. *Bioinformatics*, **28**, i356–i362.

Wilke,A. *et al*. (2016) The mg-rast metagenomics database and portal in 2015. *Nucleic Acids Res*., **44**, D590D594.

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

Wu,Y.W. *et al*. (2016) Maxbin 2.0: an automated binning algorithm to recover genomes from multiple metagenomic datasets. *Bioinformatics*, **32**, 605–607.

Wu,Y.W. and Ye,Y. (2011) A novel abundance-based algorithm for binning metagenomic sequences using l-tuples. *J. Comput. Biol*., **18**, 523–534.

Zerbino,D.R. and Birney,E. (2008) Velvet: algorithms for de novo short read assembly using de Bruijn graphs. *Genome Res*., **18**, 821–829.