Computational and Structural Biotechnology Journal 27 (2025) 221–232

Contents lists available at ScienceDirect

Computational and Structural Biotechnology Journal

journal homepage: www.elsevier.com/locate/csbj

Development of a new data management system for the study of the gut microbiome of children who are small for their gestational age

Felix Manske [a,1], Magdalena Durda-Masny [b,1], Norbert Grundmann [a], Jan Mazela [c], Monika Englert-Golon [d], Marta Szymankiewicz-Bręborowicz [c], Joanna Ciomborowska-Basheer [b], Izabela Makałowska [b], Anita Szwed [b], Wojciech Makałowski [a,*]

[a] Institute of Bioinformatics, University of Münster, Münster, Germany
[b] Institute of Human Biology and Evolution, Adam Mickiewicz University, Poznań, Poland
[c] Department of Neonatology, Poznań University of Medical Sciences, Poznań, Poland
[d] Department of Gynecology, Obstetrics and Gynecological Oncology, Poznań University of Medical Sciences, Poznań, Poland

ARTICLE INFO

Keywords:
Data management
Database
Newborn
Gut
Microbiome
Nanopore

ABSTRACT

Microbiome studies aim to answer the following questions: which organisms are in the sample and what is their impact on the patient or the environment? To answer these questions, investigators have to perform comparative analyses on their classified sequences based on the collected metadata, such as treatment, condition of the patient, or the environment. The integrity of sequences, classifications, and metadata is paramount for the success of such studies. Still, the area of data management for the preliminary study results appears to be neglected. Here, we present the development of MetagenomicsDB (http://github.com/IOB-Muenster/MetagenomicsDB; accessed 2024/12/18), a central data management system for the study of the gut microbiome in children who are small for their gestational age (SGA). Our system provided more flexibility to conduct study-specific analyses and to integrate specific external resources than existing and necessarily more generic solutions. It supports short or long read data produced by virtually any sequencing instrument targeting (parts of) popular marker genes, such as the 16S rRNA gene and its variable regions. Classifications of these reads from the MetaG and Kraken 2 software are supported. The main goals of the system are to store the pre-computed study data securely under concurrent load and to make downstream analyses accessible to all researchers, regardless of programming proficiency. Thus, after initial plausibility checks on the input data to reduce human error, data are stored in a relational database and can be continuously updated over the whole life time of the study. We used a modular approach for MetagenomicsDB with comprehensive tests verifying the expected behavior and extensively described the underlying rational which allows users to adapt the system to their needs. We advocate the use of MetagenomicsDB as the backend for a graphical web interface. We showcase the potential of this approach at the example of our study on SGA children (http://www.bioinformatics.uni-muenster.de/tools/metagenomicsDB; accessed 2024/12/02). Without restrictions caused by the level of programming proficiency, our team members could explore the study data and optionally filter them using the graphical interface, before exporting the data in a format directly suitable for external normalization of read counts and statistical analyses. Study results could be conveniently and transparently shared with the public, as demonstrated here. Links to external resources facilitated literature search with regard to the SGA condition and assessments of the potential pathogenicity of taxa. Since different users will have different demands regarding features, data security, and web environments, we provide our implementation of the web interface as a visual example. By providing users with the MetagenomicsDB backend which constitutes the major part of the system, we ensure that custom development can be finished in a reasonable amount of time. We report our endeavors in order to motivate the application of data management systems at the scale of single studies in microbiome research.

* Corresponding author.
E-mail addresses: felix.manske@uni-muenster.de (F. Manske), magdalena.durda@amu.edu.pl (M. Durda-Masny), ngrundma@uni-muenster.de (N. Grundmann), janco@pol-med.com.pl (J. Mazela), mgolon@ump.edu.pl (M. Englert-Golon), szymankiewiczmarta@ump.edu.pl (M. Szymankiewicz-Bręborowicz), joannac@amu.edu.pl (J. Ciomborowska-Basheer), izabel@amu.edu.pl (I. Makałowska), anita.szwed@amu.edu.pl (A. Szwed), wojmak@uni-muenster.de (W. Makałowski).
[1] The authors contributed equally to the study.

https://doi.org/10.1016/j.csbj.2024.12.031
Received 1 July 2024; Received in revised form 22 December 2024; Accepted 26 December 2024
Available online 31 December 2024

1. Introduction

Traditionally, the human gut microbiome was analyzed using culture-based approaches [1,13,24]. Technological advances enabled the identification of bacterial communities by whole- genome sequencing (metagenomics) or the analysis of marker genes [12], which provided a new understanding of microbial habitats: 80 % of bacterial operational taxonomic units (OTUs) in the human gut, which were detected by analysis of the 16S rRNA gene, had not been cultured before [11]. Despite new and improved efforts towards culture-based approaches [17], sequenced-based analyses are still commonly used to investigate diseases [14,21], the impact of external factors such as diet [2,9], and the mode of delivery on the microbiome [2,10].

In an ongoing prospective study, we are investigating the role of the gut microbiome in determining birth weight and weight gain for term children who are small for gestational age (SGA) throughout the first year of life by using full-length 16S rRNA gene sequences. To the best of our knowledge, a similar study has not been conducted so far. Low birth weight has been shown to increase the risk of developing various adverse conditions later in life, such as cardiometabolic disorders ([3]; Ravn [28]), attention-deficit/hyperactivity disorder [4], and learning difficulties [25]. In this study, we define SGA children as low-birth-weight children with a weight-for-age z-score ([33], 302–3) of less than negative two at birth ([36], 1216). In 2022, approximately 0.3 million live births occurred in Poland (3.9 million in the European Union) (https://doi.org/10.2908/TPS00204; accessed 2024/04/17). Given that a z-score of −2 is roughly equal to the 2.3rd percentile, we expect that almost 7000 children in Poland (approximately 90,000 children in the European Union) have been born with SGA in 2022.

During our study, we noticed that a major challenge of microbiome studies for non- programming professionals, such as laboratory scientists, is the data analysis. In our study, we had produced patient and sample metadata, sequencing results, and classifications of the sequences which needed to be stored in an accessible and secure way. Also, our analyses required the use of weight-for-age growth standards from the World Health Organization (WHO). The critical aspect of storing the different data types was not only to ensure the consistency of each, but also to maintain the correct connections between them. Subsequently, statistics needed to be performed on samples or patients of interest. This required us to subset the stored data and to provide them in a specific format to the statistical software which also takes care of normalizing the read counts. All of these operations were supposed to be performed by non-programming professionals. Thus, the use of command line interfaces or the manual execution of a collection of in-house scripts would have been an unnecessary hurdle.

In order to cope with the aforementioned challenges, we designed the central data management system MetagenomicsDB. It features a fully automatic import of patient and sample metadata from spreadsheets produced by popular programs, such as Excel (https://www.microsoft.com/en-us/microsoft-365/excel; accessed 2024/02/29) or LibreOffice Calc (https://www.libreoffice.org/; accessed 2024/02/29), but also from tab- delimited text files. Plausibility checks of the input data reduce human error and the data are stored securely even under concurrent load and can be continuously updated over the whole life time of the study. The variety of input file types enhances use for collaborative projects with medical practitioners, laboratory scientists, and bioinformaticians, who typically prefer different input formats. Sequencing results in the FASTQ format and taxonomic classifications from MetaG [19] or Kraken 2 [35] software can be retrieved from a common directory and are automatically connected to the respective samples in the database using unique identifiers from the spreadsheet. Investigators can export data of interest from the relational database to MicrobiomeAnalyst [18] for read count normalization and subsequent statistical analyses.

We designed the core components of the MetagenomicsDB backend to be as flexible as possible in order to increase the value of our system for future studies. Components were built using a modular concept and were supplemented with tests which verify the current behavior and provide a baseline of expected behavior for future adaptations. Together with the extensive description of the underlying rational, this permits users to customize existing functionalities or to add new ones by modifying existing or adding new components. In line with this, we used a relational database management system to keep the user interface inert to alterations of the physical and logical structure of the stored data ([7], ID/8). Out of the box, MetagenomicsDB supports short or long read data produced by virtually any sequencing instrument targeting (parts of) marker genes, such as the 16S rRNA gene and its variable regions. This is provided basic formatting requirements for the FASTQ files are fulfilled and sequences are independent. For example, paired-end reads should be merged. We advocate that our MetagenomicsDB software should be used as the backend for a graphical web interface. Since user requirements regarding included features, data security, and web environments are different, a graphical interface is not included with our software. Instead, we provide an easy-to-copy visual example by making a reduced subset of our SGA study data publicly accessible through a graphical web interface: https://www.bioinformatics.uni-muenster.de/tools/metagenomicsDB (accessed 2024/12/02). Since the backend constitutes the main part of the total workload, we expect that users can finish their adaptations in a reasonable amount of time. The MetagenomicsDB backend (https://github.com/IOB-Muenster/MetagenomicsDB; accessed 2024/12/18) in concert with our web interface allowed all users, regardless of bioinformatics expertise, to store, explore, filter, and analyze the study data. Links to external resources facilitated the interpretation of the study data. As demonstrated here, the whole interface can be made public for transparent sharing of results. We reported our endeavors in order to motivate the application of databases as central components for managing data produced by ongoing studies.

2. Materials and methods

2.1. Study of the gut microbiome in small for gestational age children

We conducted a blinded prospective study on the potential influence of the gut microbiome on birth weight and weight gain in SGA children during the first year of life. The study cohort consisted of otherwise healthy (Supplementary Methods 1) single birth children of Polish descent and was recruited from the Heliodor Święcicki Clinical Gynecology and Obstetrics Hospital of Poznań University of Medical Sciences (Poznań, Poland). No siblings were present in the cohort and all children were born on time (Supplementary Methods 1). Newborns were divided into cohorts of appropriate for gestational age and SGA children (Supplementary Methods 1). 65 children were examined in regular intervals at a maximum of 8 time points (Supplementary Methods 1) from birth (“meconium”) to the age of 1 year (“1 y”) in order to assess the weight gain and to sample the gut microbiome. At birth (“meconium” sample), we recorded factors related to the pregnancy and neonatal development, such as potential illnesses of the mother and her weight gain during pregnancy (Supplementary Table A1; Supplementary Methods 2). Samples were skipped if the children were hospitalized or had diarrhea when the sample was scheduled. During each appointment, children were weighted and stool samples were collected (Supplementary Methods 2 and 3.1). Patient and maternal metadata, measurement results, confounding factors to the gut microbiome (antibiotics, probiotics, and feeding habits), and selected data on the course of the pregnancy, including maternal health, were stored in an Excel spreadsheet together with derived measurements (Supplementary Methods 2; Supplementary Table A1). Derived measurements represented values which were calculated based on other values in the spreadsheet (e.g. “mother’s age at delivery”, Supplementary Table A1).

2.2. Sequencing of stool samples and in silico analyses

In the lab, DNA was extracted from the fecal samples using the DNeasy PowerSoil Pro Kit (QIAGEN GmbH, Hilden, Germany) (Supplementary Methods 3.2). After verifying that the extraction was successful (Supplementary Methods 3.2), we created full-length 16S rRNA gene amplicons using the 16S Barcoding Kit 1–24 (Cat. No.: SQK-16S024, Oxford Nanopore Technologies plc, Oxford, England). 23 of the 24 available barcodes were used for the genuine samples. The remaining barcode (by our definition: 99, Supplementary Methods 5.3.1) was assigned to a sample containing distilled water, which served as an internal negative control for library preparation and sequencing (Supplementary Methods 3.3). The finished library was sequenced for 24 hours using two MinION Mk1B nanopore sequencers (Cat. No.: MIN-101B, Oxford Nanopore Technologies plc, Oxford, England) with Flongle Flow Cells (Cat. No.: FLO-FLG001, Oxford Nanopore Technologies plc, Oxford, England). Real-time basecalling was performed using Dorado v0.3.3 in MinKNOW version 23.07.5 (https://community.nanoporetech.com/downloads; accessed 2024/06/08)

(Supplementary Methods 3.3). After manual quality control (Supplementary Methods 3.3), FASTQ "pass" files were processed with MetaG. We removed residual human contamination using the T2T-CHM13v2.0 (RefSeq [30] accession: GCF_009914755.1) human reference genome in order to improve the accuracy of the taxonomic classifications and help protect the privacy of patients (Supplementary Methods 4.1) before classifying the remainder using RDP [8] version 11.5 (Supplementary Methods 4.2). Both analyses used new MetaG standard parameters, which were generally applicable and not specific to the study of the gut microbiome (Supplementary Methods 4.1.1, 4.1.2, 4.2.1, and 4.2.2). FASTQ "pass" files and MetaG classifications were stored in a common directory with subdirectories named after the run and the barcode of the 16S rRNA gene reads (Supplementary Methods 5.3.1). The run and the barcode from sequencing the stool samples, the classification program (MetaG), and the classification database (RDP) were added to the Excel spreadsheet (Supplementary Table A1). We used the run and the barcode as a unique key to assign the samples from the spreadsheet to the respective sequences and classifications. The program column was used to choose the appropriate parser for the classification data.

2.3. Development of a central data management system

We developed a central data management system called MetagenomicsDB to allow all users, regardless of bioinformatics expertise, to store, explore, and export their preliminary study results (Sections 2.1 and 2.2) in a format suitable for downstream read count normalization and statistical analyses. During development, we aimed to create a modular system which could be adapted for future microbiome studies. MetagenomicsDB consists of three main units (Fig. 1) for import, storage, and export. We built a custom graphical frontend for our use of the system, which is specific to our own web environment and is hosted on an Apache (https://apache.org/; accessed 2024/03/22) 2.4.48 server running under FreeBSD 13 (https://www.freebsd.org/; accessed 2024/03/17). It uses in-house JavaScript and Cascading Style Sheets. The interactions between the web interface and the users are protected by Hypertext Transfer Protocol Secure (HTTPS). The import and export unit, as well as the helper scripts for the web interface, are implemented in Perl 5. Preliminarystudyresultsandweight-for-agegrowthstandardsfromtheWHO (https://cdn.who.int/media/docs/default-source/child-growth/child-growth-standards/ indicators/weight-for-age/expanded-tables/wfa-boys-zscore-expanded-tables.xlsxand https://cdn.who.int/media/docs/default-source/child-growth/child-growth-standards/

indicators/weight-for-age/expanded-tables/wfa-girls-zscore-expanded-tables.xlsx; accessed 2024/01/30) are stored in a relational PostgreSQL (https://www.postgresql.org/; accessed 2024/03/17) 14beta2 database located on a database server with two Intel E5- 2660 v2 CPUs, 384 GB RAM, and three 8 TB hard drives running under FreeBSD 13. Our data management system is described in depth in

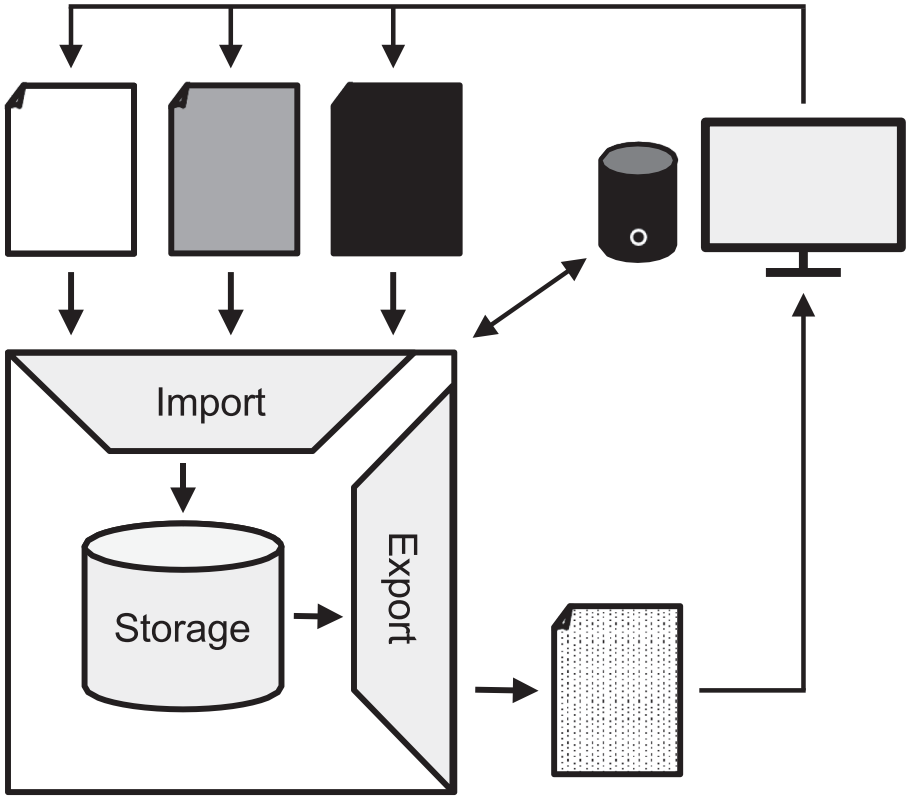

Fig. 1. Components of MetagenomicsDB.

Section 3 and Supplementary Methods 5.

3. MetagenomicsDB: a central data management system

MetagenomicsDB consists of three main components (Fig. 1): an import unit which checks the integrity and plausibility of the input data and synchronizes them with the database (Section 3.1). This way, new data can be added over the whole life time of the study. Preliminary study results and associated metadata are stored using the relational database management system PostgreSQL (Section 3.2). Another unit exports a user selection of the database contents in a format suitable for downstream statistical analyses in MicrobiomeAnalyst (Section 3.3). The source code for the MetagenomicsDB backend is available from our GitHub repository (https://github.com/IOB-Muenster/MetagenomicsDB; accessed 2024/12/18), including detailed installation instructions and a set of simulated data (Supplementary Methods 5.3.3), so users can directly start to explore the system. All components can be operated using a web-based graphical interface (Fig. 1), which allows users, regardless of their bioinformatics expertise, to store, explore, and export the preliminary study results. In the following, we will present our implementation as a visual examples for others (Section 3.4).

Different inputs, illustrated by the three differently colored files, are subject to plausibility checks and synchronized with the storage unit with the help of a general import unit. A user selected subset of data is then exported by the respective component in a format suitable for downstream analyses (dotted file). For our study, we implemented a web-based graphical interface (computer terminal). It allowed us to upload input data for the synchronization with the database, to explore the data contents, and to export the results. This figure was generated with LibreOffice Draw 7.0.4.2 (https://www.libreoffice.org/discover/draw/; accessed 2024/03/22).

For future microbiome studies, we aimed to design MetagenomicsDB as a non- disposable data management system (Supplementary Methods 5.4). Thus, the import and export units (Fig. 1) were built modular and supplemented with tests verifying the expected behavior. This will allow us and others to reuse universal functions which we expected to be useful across projects, for example functions parsing FASTQ files or spreadsheets. Other more high-level functions, such as definitions of the columns which were supposed to be extracted from the spreadsheets, necessarily had to be project- specific. Due to the modular concept, the extensive tests, and the thorough description of the underlying rational, project-specific functions can be easily replaced with their successors for future projects, considerably reducing the workload for follow-up projects. Modularity also facilitates the extension of the current functionalities to fit changed

demands. For instance, only classifications from the MetaG classifier were processed in this study (Supplementary Methods 4.2). In future projects, the use of other popular classifiers, such as QIIME 2 [5] or Kraken 2 [35], could be of interest. To achieve support for more classifiers, the only major change to the import unit of MetagenomicsDB is the addition of a new function parsing the specific output file formats of the new classifiers and converting them to the common internal data representation of our pipeline. To illustrate this aspect of our pipeline, we recently added support for the Kraken 2 classifier. While the new feature required extensive tests, the implementation of the actual functionality was limited to few lines of code (files “bin/importSGA.pl”, “lib/perl/MetagDB/Sga.pm”,and“lib/perl/MetagDB/Taxa.pm”at https://github.com/IOB-Muenster/MetagenomicsDB/commit/fbfdeea907c5a4b09f067ef512 3f6dc4822e2dd7; accessed 2024/12/04). In line with the modular concept, we designed a flexible database schema (Section 3.2) and used the relational database management system PostgreSQL as the storage component, since relational database management systems keep the user interface inert to alterations of the physical and logical structure of the stored data ([7], ID/8). In the following sections, the three components of MetagenomicsDB (Fig. 1) will be presented. In a subsequent section, we describe the features of the graphical web-based interface for our SGA study as a visual example for users. An exhaustive description of the implementation and the underlying rational is presented in Supplementary Methods section.

3.1. Import of the study data

A single program called importSGA.pl (commit: 4bfc752; http://github.com/IOB-Muenster/MetagenomicsDB/blob/main/bin/importSGA.pl; accessed: 2024/12/09) was responsible for the import and the plausibility checks of the Excel spreadsheet with the measurements and metadata, the FASTQs, the MetaG classifications of reads (“calc.LIN.txt” files), and the WHO weight-for-age growth standards (Sections 2.1 to 2.3; Supplementary Table A6), which were essential to evaluate the growth of the children. The main duties of this program were reading and, if necessary, extracting the input files, parsing the contents, validating the inputs to reduce human error, and synchronizing the data with the database. In the following, we will briefly present the general algorithm of our program for the handling of different input file types. For the sake of simplicity, we will use the term “measurements” for all contents of the Excel spreadsheet.

The WHO standards were only inserted once at the beginning of the project and read by our program as is (Supplementary Methods 5.3.1). The software processed the read classifications based on the entry in the “program” column of the measurement spreadsheet. Classifications from the MetaG software had to be provided as “calc.LIN.txt” files (Supplementary Methods 5.3.1). The files contained the classifications per read and per rank (domain, phylum, class, subclass, order, suborder, family, genus, species, and strain). For each of these ranks, reads that could not be classified at all and reads that could not be classified from the respective rank onward were assigned to the special taxon “UNMATCHED.” Reads were assigned to the taxon “FILTERED” over all ranks if the read appeared in the FASTQ files but not in the “calc.LIN.txt” files. Since we had removed reads with human contamination prior to the taxonomic classification (Section 2.2), the “FILTERED” taxon represented reads identified as human contamination. The taxon name was null in the case no name was available (Supplementary Methods 5.3.1). Our program is also capable of processing the standard Kraken 2 output format (https://github.com/DerrickWood/kraken2/wiki/Manual#output-formats;accessed 2024/12/04) in an analogous fashion (Supplementary Methods 5.3.1), however this was not required for our SGA project. The Kraken 2 files (“*kraken2 *”) contain the taxonomy ID of the lowest assigned rank. Internally, our program uses TaxonKit [32] v0.19.0 to translate the IDs to lineages. At the time of writing, this version of TaxonKit has not been officially released, but can be accessed from the issue section of the official GitHub page (https://github.com/shenwei356/taxonkit/issues/107; accessed 2024/12/04). After an optional extraction, our program concatenated all FASTQ files (Supplementary Methods 5.3.1) found in a subdirectory. However, we enforced that files had to have distinct names (not considering the file extension) in order to avoid concatenating duplicate data. By comparing the read IDs in both files, importSGA.pl verified that FASTQ files and the respective classification files were matching. Sequences, quality strings, read IDs, and nanopore-specific metadata from the headers were extracted in order to store them in the database (Supplementary Methods 5.3.1). Short or long reads from virtually any sequencing instrument can be processed, as long as they are targeting (parts of) marker genes, such as the 16S rRNA gene or its variable regions. Additionally, basic formatting requirements have to be fulfilled (Supplementary Methods 5.3.1). Our data model assumes sequences to be independent (Fig. 2), so paired-end reads must be merged *a priori*.

Only certain columns were extracted from the Excel spreadsheet containing the measurements (Supplementary Table A7). The vast majority of ignored columns represented measurements whose values could be calculated based on values of other measurements. We named these derived measurements (Supplementary Table A4). Derived measurements were calculated in the database itself (Section 3.2; Supplementary

Methods 5.1.2) in order to avoid update anomalies. For example, the weight-for-age z- score of a patient depended on his or her birthdate, the patient's sex, the samp le collection date, the patient's body mass, and the WHO growth standards (Supplementary Methods 5.1.2). If the z-score was only read from the input, a later update of the patient's birthdate in the database would leave the database in an inconsistent state, as the dependent z- score would not have been automatically updated. Some columns in the spreadsheet contained values which were encoded by numbers. importSGA.pl automatically translated these numbers into the referenced values (Supplementary Methods 5.3.2).

The spreadsheet connected the sequences and classifications to the respective samples by providing a unique key in the form of the run and barcode of the sequences (Section 2.2). These keys only referred to the genuine stool samples and not to the internal controls from each run. In order to link the control to all samples of the respective sequencing run, we internally duplicated each sample. One copy was labeled as control and only connected to the sequences from the control barcode 99. The other copy was labeled as the case sample and was assigned to the

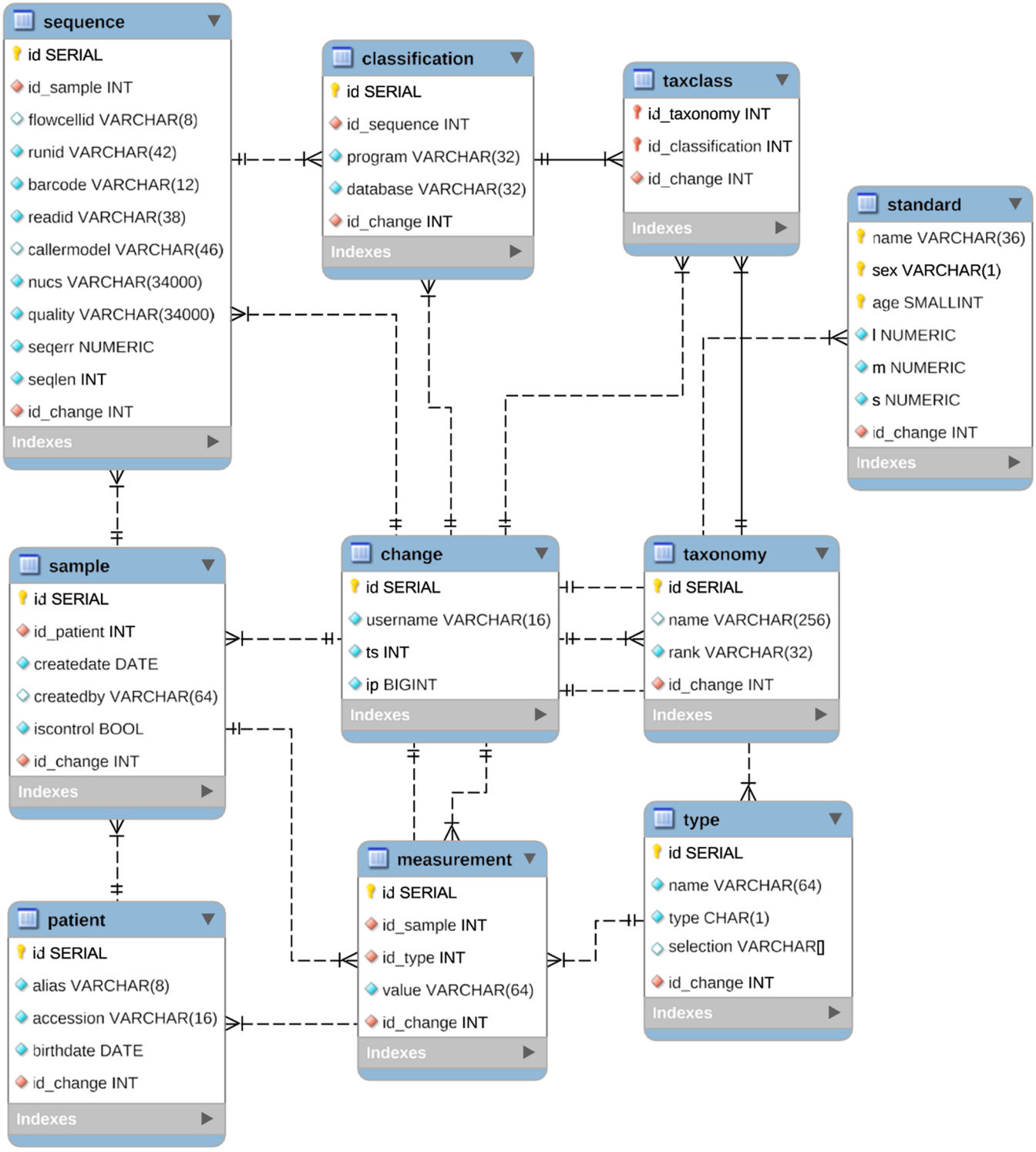

Fig. 2. Relations in MetagenomicsDB.

genuine stool sample sequences and the extracted measurements from the spreadsheet (Supplementary Methods 5.3.2).

The import program connected to the database with the service name "metagdb" provided in the PostgreSQL configuration file ".pg_service.conf" on the host machine (Supplementary Methods 5.3.2), which made the application easily portable. We ensured that only one instance of our program was running at the same time by attempting to acquire an exclusive transaction-level advisory lock. If the lock was already held, the new instance of the program was terminated (Supplementary Methods 5.3.2). All database manipulations were performed in the same transaction that was used to acquire the lock, thus all changes could be completely rolled back in the event of an error. Inserts into and updates of the database were conducted in batches for improved performance over the network (data not shown).

After connecting to the database, our script filtered duplicates in the input data and autonomously decided how to synchronize the input files with the database by using the unique constraints of the relations (Supplementary Methods 5.1.1 and 5.3.2). This allowed us to continuously update the study data over the course of the project by simultaneously providing old records as well as records that were supposed to be inserted or updated as input data. Generally, however, it was advantageous to limit input files to new or changed records in order to reduce the calculation time. Nevertheless, even in the case of such a minimal synchronization, old data were mandatory when they were needed to establish the correct connections between the data types (Supplementary Methods 5.3.2). For example, it was not possible to insert FASTQ files by themselves, as the measurement spreadsheet was essential to establish the connection between the sequences and the respective sample. Similar considerations were applied to the classification files which depended on the sequence files and thus also on the measurement spreadsheet (Supplementary Methods 5.3.1 and 5.3.2).

Overall, initializing the database from scratch with the WHO growth standards, 65 patients, 986 samples, and more than 3.8 million sequences (of which over 97 % remained after filtering for human contamination), required roughly 3 hours on a Debian 11 server (2 AMD EPYC 7452 CPUs, 1 TB RAM) with access to the input data and the database over the network. The most time-consuming step was the parsing of FASTQs and the subsequent insertion of the sequence data, which accounted for more than half of the total calculation time.

The key symbol indicates a primary key and the diamond symbolizes a regular column. Foreign keys are depicted by red symbols and turquoise shading indicates "not NULL" constraints. In our schema, all foreign keys also had a "not NULL" constraint (commit: 68b05c6; https://github.com/IOB-Muenster/ MetagenomicsDB/blob/main/www-intern/db/schema.sql; accessed 2024/06/07). Connections between the relations are shown by lines (dashed for non-identifying relationships). This database schema was visualized with MySQL Workbench 6.3.8 build 1228 CE Community (https://dev.mysql.com/downloads/workbench/; accessed 2024/03/15) and edited with Inkscape 1.0.2 (https://inkscape.org/; accessed 2024/03/15).

3.2. Data storage

Data were stored in a PostgreSQL 14beta2 database. The schema consisted of ten relations (Fig. 2) and five materialized views. The database schema is presented in depth in the Supplementary Methods (Supplementary Methods 5.1.1 and 5.1.2). In this section, we will focus on the general properties of the database schema. Names of views and relations will be printed in italics to distinguish them from regular English words.

In our study, each patient in the *patient* relation was sampled at up to eight time points. Thus, individual patients were assigned to a maximum of eight genuine samples and eight control samples in the *sample* relation (Sections 2.1 and 3.1). The controls represented the results from sequencing the internal control sample of each run (Supplementary Methods 3.3) and were identified by the value in the column "isControl" in *sample* (Fig. 2). Controls had no measurements, as opposed to the genuine samples. In order to keep the database schema flexible with regards to the number of stored measurements, we stored single measurements as rows in the *measurement* relation (Fig. 2). This approach supported an arbitrary number of measurements per sample. The values of static measurements were only recorded once and never changed. These were values related to the mother, the pregnancy, and the birth (e.g.: maternal body mass before pregnancy, pregnancy order, and birth mode, respectively). The sex of the patient was also considered static (Supplementary Table A3). In order to reduce redundancy, we assigned static measurements only to the first sample ("meconium" sample) from each patient. Variable measurements, such as the patient's body mass, were recorded during each of the appointments and were assigned to the respective samples (Supplementary Table A3). Derived measurements were only calculated in the views (see below). Measurement names were normalized by the *type* relation. The relation additionally contained the expected input type of the measurement value and, if appropriate, an array of possible values ("type" and "selection", Fig. 2). This information was used in an internal custom administration web interface in order to check the input values and to generate drop-down menus. Due to the storage of single measurements as rows in *measurement*, it was not possible to enforce specific constraints on the data type of values at the level of the database (Fig. 2).

Each sample could be linked to multiple sequences. In the *sequence* relation, we calculate the read length and average sequence error of each entry ("seqlen" and "seqerr", Fig. 2) using stored generated columns. These helped to reduce redundancy, as read length and average sequence error could be directly inferred from the sequence and the quality string of each read, respectively (Supplementary Methods 5.1.1). Both values were later used by the views to calculate maximum, minimum, and average values for the sequence length and the average read quality of sequences supporting a specific taxon (Supplementary Methods 5.1.2). Each read could be classified with different programs and databases. We recorded this information in the *classification* relation (Fig. 2). In our exemplary study, however, all reads were classified by MetaG using the RDP database (Supplementary Methods 4.2). The taxa from the classifications were stored separately by name and rank in the *taxonomy* relation, which was linked by the associative table *taxclass* to *classification* (Fig. 2). The taxa in *taxonomy* were unique considering the combinations of name and rank (Supplementary Methods 5.1.1). However, in the course of the study, we found that RDP contained taxa with the same name and rank that came from different lineages (data not shown). Thus, even though taxa were considered identical by the database, they could sometimes represented distinct biological entities. By default, read counts and supporting metadata in our custom web interface were calculated based on unique taxa as per the database definition. Nevertheless, we also displayed read counts based on unique lineages via a separate menu (Section 3.4). The *change* relation recorded information about the submitter of each record, which we used internally to track changes (Fig. 2). Isolated from the rest but *change* was the *standard* relation (Fig. 2), which contained information from the WHO growth standards (Supplementary Methods 5.3.1) and was needed to calculate the weight-for-age z-scores in the views (see below).

We used materialized views to calculate the derived measurements (such as the aforementioned weight-for-age z-score in *v_measurements*), to aggregate data for the export to MicrobiomeAnalyst (*v_lineages*, *v_samples*, and *v_metadata*), and for the functionalities of our web interface (Section 3.4; Supplementary Methods 5.1.2). Materialization was chosen in order to improve the speed of queries and views were automatically refreshed when necessary by importSGA.pl. Apart from the derived measurements (Supplementary Methods 5.1.2), the views also calculated a time point string, which expressed the sample collection date relative to the birthdate of the respective patient (*v_samples*). This allowed for convenient comparisons of samples from different patients at the same time point after birth. The definitions of time points were fuzzy in order to account for minor variation in the sample collection

date (Supplementary Methods 5.1.2). For example, samples taken between 354 and 376 days were labeled as the 1 year ("1 y") sample. This increased flexibility came at the expense of not being able to enforce uniqueness of sampling time points at the level of our database. Additionally, it was not guaranteed that a time point string was assigned to each sample collection date, as time points were discrete. However, importSGA.pl enforced these two constraints when inserting or updating samples (Supplementary Methods 5.1.2). Other important information that was calculated in the views was the read count per sample (*v_samples*) and the assembly of the taxonomic classifications for a read into a complete lineage (*v_lineages*). Furthermore, minimum, maximum, and average values were calculated for the sequence length and the average read quality of sequences supporting a taxon identified by a specific program and database (*v_taxa*) (Supplementary Methods 5.1.2).

3.3. Export for statistical analyses

MetagenomicsDB provided an export of data to the "Marker Data Profiling" workflow (https://www.microbiomeanalyst.ca/MicrobiomeAnalyst/upload/OtuUploadView.xhtml; accessed 2024/02/01) of MicrobiomeAnalyst in order to facilitate downstream statistical analyses, including read count normalization. MicrobiomeAnalyst is a tool that can use (among others) taxonomic classifications together with sample metadata to perform statistical analyses and functional predictions in a graphical web interface [18]. When provided database IDs for samples of interest, exportSGA.pl (commit: 68b05c6; https://github.com/IOB-Muenster/MetagenomicsDB/blob/main/bin/exportSGA.pl; accessed: 2024/06/07)exportedclassificationsandsamplemetadatatoaZIParchive (Supplementary Methods 5.2; Supplementary Table A5). By default, we excluded the "FILTERED" taxon and skipped any sample associated with the internal controls (Supplementary Methods 5.2). We only exported samples that had classifications and wrote the number of samples which were not exported to a log file in the archive ("microbiomeanalyst_WARNINGS.txt"). The raw counts of supporting reads for each lineage across all exported samples were stored in "microbiomeanalyst_otu.txt". Lineages and samples were identified by unique keys. The keys for lineages contained the term "OTU" followed by a counter. The counter was based on the sorted order of lineages in the respective export. Thus, the counter was stable between multiple exports of the same samples (assuming the database contents remained unchanged), but was not lineage- specific across exports with different samples. Each ID was connected to its lineage (kingdom [here: domain], phylum, class, order, family, genus, and species) in the "microbiomeanalyst_tax.txt" file. Taxa with an empty name were called "NoName" in the file and "UNMATCHED" taxa were set to "NA" (Section 3.1). Sample metadata were provided in the "microbiomeanalyst_meta.txt" file and linked to the lineage counts in the "* _otu.txt" file using the unique sample keys. Exported metadata included values such as antibiotics and probiotics usage, the sampling time point, and the category of the weight- for-age z-scores (Supplementary Table A5). Missing values in the metadata were replaced by "NA."

3.4. Web-based interface

MetagenomicsDB is intended to be used as the backend for a graphical web interface. We showcase the potential of this approach by making our custom web interface for the aforementioned SGA study publicly accessible (https://www.bioinformatics.uni- muenster.de/tools/metagenomicsDB; accessed 2024/12/02). The interface enabled our team members, regardless of bioinformatics expertise, to store, explore, and export the preliminary study results. The web interface was conceptually simple, thus we expect that future users can adapt our visual example to their specific web environments. Most of the data for the display are already aggregated in the views and the import and export are performed by the respective components of MetagenomicsDB (Sections 3.1 - 3.3). The only exception is the export of sequences in FASTQ format. The web interface was built based on the concept of tabs. Tabs contain collections of related data and are available for patients, samples, and taxa. The patients tab contains the patient metadata, selected static measurements (Supplementary Tables A3 and A5), and derived measurements related to the mother (Supplementary Tables A4 and A5). Samples and sequence counts per sample, selected variable measurements (Supplementary Table A3), and z-score (sub-)categories (Supplementary Table A4) are shown in the samples tab (Supplementary Table A5). Here, users can also export sequences in FASTQ format for samples of interest. For performance reasons, the download is limited to approximately 800 megabytes. Sequences assigned to the "FILTERED" taxon cannot be exported by external users to help protect the privacy of our patients (Section 3.1).

By selecting samples or patients (in this case, internally the associated samples are used) from the respective tabs, users can export data to MicrobiomeAnalyst. In an overlay, the default settings of the export can be adjusted (Section 3.3). The ZIP archive with the exported data is then downloaded to the user's device. A new tab with the aforementioned "Marker Data Profiling" workflow is automatically opened in the user's browser, so that files can be directly uploaded after the archive has been extracted. The extracted files are uploaded to the "Text table format" tab on the website of the "Marker Data Profiling" workflow (Fig. 3). The file "microbiomeanalyst_otu.txt" is uploaded as "OTU/ASV table" (the associated check boxes are left unchecked), "microbiomeanalyst_meta.txt" as "Metadata file", and "microbiomeanalyst_tax.txt" as "Taxonomy table" (Fig. 3). For classifications based on RDP, "Not Specific/Other" has to be chosen for "Taxonomy labels" (Fig. 3).

This screenshot of the upload interface on MicrobiomeAnalyst [18] website (https://www.microbiomeanalyst.ca/MicrobiomeAnalyst/upload/OtuUploadView.xhtml; accessed 2024/02/01) illustrates how the files exported from MetagenomicsDB should be uploaded.

The taxa tab of MetagenomicsDB contains information about the taxa identified in each sample, the associated analysis program and database, and statistics on the supporting sequences (Section 3.2; Supplementary Table A5). Its display is limited to 10,000 entries in order to ensure good performance for users. Taxa in the taxa tab are grouped by name and rank and not by lineage in order to calculate the counts of supporting reads. In the RDP database that is used to classify the sequences (Supplemental Methods 4.2), taxa with identical names and the same rank can occasionally come from different lineages (Section 3.2). In these cases, read counts supporting the common taxon name can still be summed up. Users can click the question mark button in the "Taxon Name" field to view the read counts per full lineage (*v_lineages*, Supplementary Methods 5.1.2). The question mark button thus reveals if taxa from different lineages were merged (Supplementary Table A5).

Using the name of the taxon, the tab provides several links to public resources which can be useful to obtain further information, also with regard to potential health consequences of the observed taxon (Supplementary Table A5). We created a direct link to PubMed [30] which queries the database for literature on the specific taxon with a focus on birth weight or SGA. The exact search term is given below. The position of the taxon name is indicated by the bold question mark:

(**?**[Title/Abstract]) AND (
) AND (
)
(birth weight[Title/Abstract]) OR (SGA[Title/Abstract]) OR
(small for gestational age[Title/Abstract])
(newborn[Title/Abstract]) OR (infant[Title/Abstract]) OR (child [Title/Abstract]) OR (baby[Title/Abstract])

The interface also creates links to articles for the respective taxon on three other public resources: Wikipedia (https://en.wikipedia.org/; accessed 2024/02/02), the List of Prokaryotic names with Standing in Nomenclature (LPSN) [27], and the Bacterial and Viral Bioinformatics Resource Center (BV-BRC) [26]. The latter is used to obtain information on the potential pathogenicity of the taxon. Links were created without *a priori* information, if the entry existed in the respective web resource.

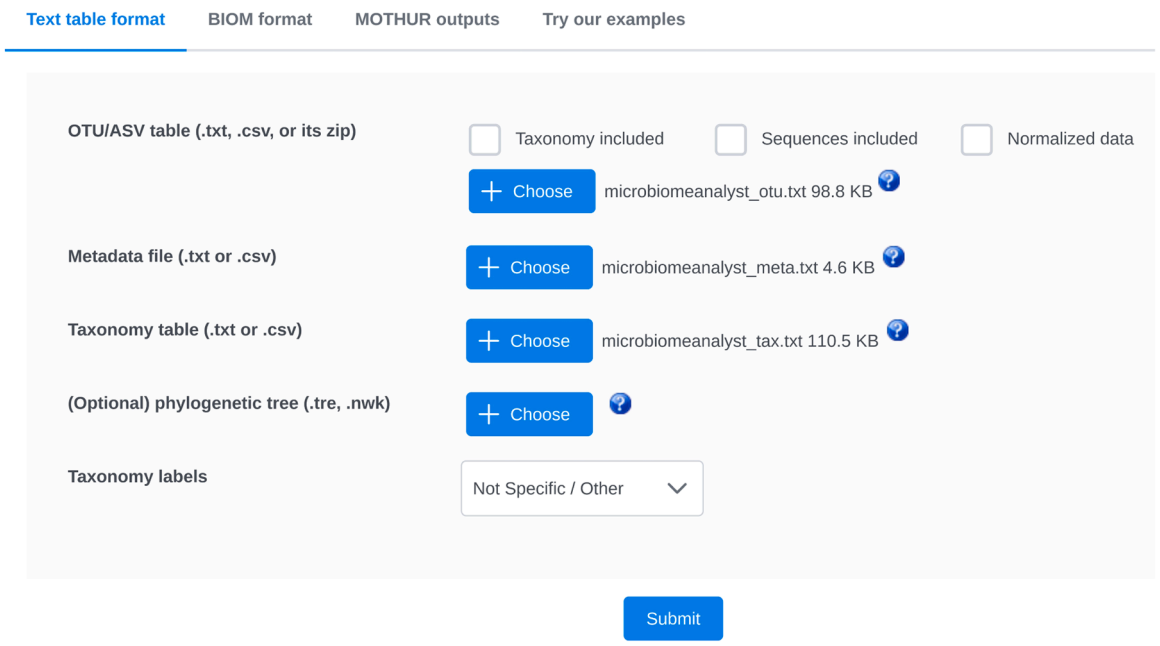

Fig. 3. Upload interface for the "Marker Data Profiling" workflow.

Thus, links might show no search results for certain taxa. However, we exclude special taxa like "UNMATCHED," "FILTERED," and "NA" (used for empty taxon name), since they are not expected to produce meaningful results (Section 3.1).

From the start page of MetagenomicsDB (Fig. 4), tabs can be queried by selecting a tab of interest before entering search terms or making selections for searchable fields. It is not possible to conduct a search based on criteria from different tabs at the same time. Nevertheless, the results can be later refined in order to realize more complex queries (see below). Internally, searches in multiple fields are connected with an "and." A letter in front of the search bars indicates the expected data type of the search term (Fig. 4): "F" for floating point numbers, "I" for integers, and "T" for text. Entries in fields of type "F" or "I" are automatically corrected, if they contain text. Entries in fields of type "T" are interpreted internally as character varying. Thus, they can contain text, as well as integers and special characters. Advanced searches can be conducted by entering symbols together with the search term. The mathematical signs "> =," "< =," "> ," and "< " can be used either individually or in combination (internally: "and") to search integers and floating point numbers. By default, text-based searches have to have a full match and searches are case-sensitive. However, the use of "~" triggers a pattern match. For example, entering "~Escherichia" in the search bar for "Taxon Name," searches for all names containing the term "Escherichia" (not case-sensitive). Multiple search terms can be joined by commas (internally: "or"). Likewise, if multiple checkboxes for a given field are selected, they are internally connected by an "or." Users can also browse through a tab by submitting an empty query.

While the user is filling out the query form, all database IDs matching the query are internally collected. The number of matching entries is displayed on the query website (Fig. 4) and dynamically updated when the user leaves a field of the form. After the query was submitted, all tabs are filled with data based on the collected IDs. Only the chosen tab is displayed, the remaining tabs are stored in the cache. When the user switches tabs, the currently displayed tab is exchanged with the selected tab from the cache. This approach does not require further database requests and thus reduces the loading time between tabs.

The interface shows the search bars and checkboxes for searchable fields in the patients tab (https://www.bioinformatics.uni-muenster.de/tools/metagenomicsDB; accessed 2024/12/02). The letters in front of the search bars indicate the expected data types of the search terms: "T" for text, "I" for integers, and "F" for floating point numbers. The number of matches for the query is indicated in the bottom right corner.

Search results can be further refined by selecting records of interest and clicking the "Apply" button. From hereon we will refer to this action as "applying a selection" in order to distinguish it from a different type of refinement (see below). When a selection is applied, the database IDs stored in the browser are replaced by the IDs of the selected records. This affects the remaining tabs as well. The operation can be undone using the "Undo" button. In contrast, the display of the records within the active tab can also be changed without affecting the internal selection of database IDs and thus, the other tables In the samples tab, the display of samples can be filtered by a minimum and/or maximum number of

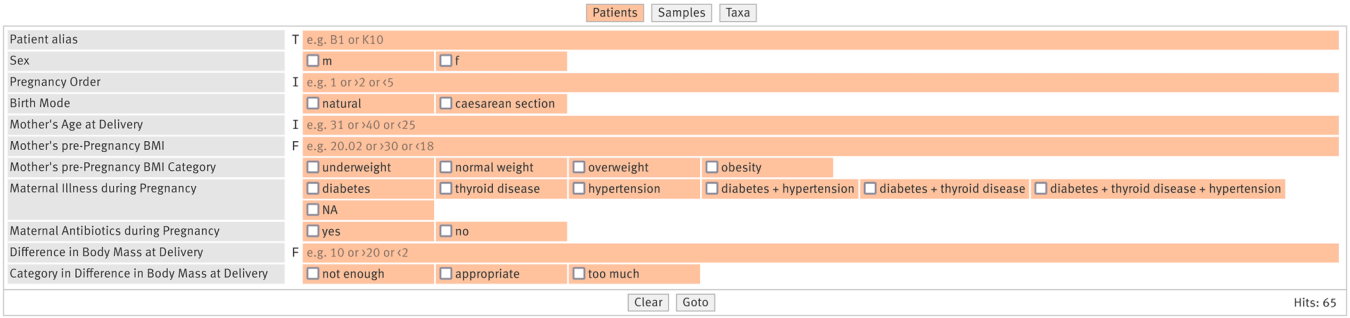

Fig. 4. The search interface of MetagenomicsDB.

sequences ("# Sequences"). The display of records in the taxa tab can be filtered by the rank ("Ranks"), the average read length (minimum and maximum; "Avg. Read Length"), the average read quality (minimum and maximum; "Avg. Read Quality"), the supporting read count (minimum and maximum; "Count"), or any combination thereof. Subsequently, a new selection can be applied to the remainder of displayed records (see above). The combination of filtering the display in the active tab and of applying a selection enables more complex queries. To demonstrate this, we assessed how many patients had at least 5000 reads with a minimum average quality of 10 supporting *Escherichia coli* in their "meconium" samples. We first searched for "Escherichia coli" in the "Taxon Name" field of the taxa tab on the main query page. Before a selection was applied to the remaining records, the display in the taxa tab was filtered by a minimum average read quality of 10 and a minimum read count of 5000. In the samples tab, a selection was applied to the "meconium" samples. After switching to the patients tab, it became apparent that two patients (K19 and K27) matched this complex query.

4. Discussion

As of today, several programs are available which integrate data management and data analyses. MG-RAST [34] and MGnify [29] both offer standardized analyses of sequences with associated metadata, data storage, and comparative analyses in a graphical web interface [23,29,34]. Both services offer to make results accessible to the public in their associated repositories [23,29,34], however MGnify demands to eventually publish the data (Table 1). Lack of private analyses and the requirement to upload the raw study data to the internet can be problematic under some circumstances, for example in the context of clinical research or the pharmaceutical industry where certain data have to remain confidential. The standardized workflows of both repositories allow the comparison of results across studies in meta-analyses [29,34], but come at the expense of not being able to use custom analyses workflows with different tools and databases which may produce higher quality results for the individual study (Table 1). At a smaller scale, MANTA [6], metaXplor [31], and myPhyloDB [20] assist researchers in managing study results for private analyses using graphical interfaces [20,31,6]. We evaluated all software except myPhyloDB to see, if they would be applicable to our project (Table 1). The myPhyloDB program was removed from the comparison, as it relies on Python 2.7 which is obsolete (https://github.com/manterd/myPhyloDB; accessed 2024/06/03). This indicates that future development is unlikely to happen. MANTA provides flexibility with regard to custom classifiers and classification databases (Table 1). However, while all retained programs at least store read classifications and sample metadata (Table 1), MANTA is the only one (Table 1) which does not save sequencing results (https://github.com/chenyian-nibio/manta/blob/master/documents/schema.png; accessed 2024/06/04). Besides, it does not appear to link to external resources or to export the data in standard formats for downstream statistical analyses (Table 1). Sequences should be stored to enable potential reanalysis and thus, MANTA is not applicable for our use case. In contrast, metaXplor also manages sequences, but it forces external classifications to yield NCBI taxonomy IDs or NCBI accessions [30,31], which considerably narrows the choice of applicable classification databases. In our study, we strive to use a more flexible approach than permitted by the workflows of the aforementioned tools. First, we are interested in using the MetaG software for the classification of our 16S rRNA gene nanopore reads, since our benchmarking has indicated that MetaG achieved better performance on nanopore amplicons than other established programs [19]. In our study, MetaG uses the RDP database which does not rely on any NCBI identifiers, conflicting with metaXplor's requirements (see above). Second, our derived measurements contain dependencies on other measurement values and sometimes on external resources (as in the case of the weight-for-age z-scores), thus they are best calculated by a data management system based on stored data in order to maintain data integrity throughout the project (Section 3.1). An additional benefit of this approach is that when the underlying algorithm for the calculation of the derived measurements is altered, data are automatically updated without the need for user interaction. To the best of our knowledge, no other software in our comparison considered these dependencies between measurements (Table 1). Third, while most of the analyzed programs provide links to further more general metadata (Table 1), our custom data management system enables us to link our preliminary study results directly to more specialized web resources (Section 3.4).

Thus, we developed the central data management system MetagenomicsDB to help us store, explore, and analyze the data produced by our ongoing study (Table 1). We argue that a central data management system can alleviate many of the challenges inherent to a microbiome study of a medical condition much better than conventional analyses based on data scattered over separate files. By bundling all preliminary results in a central location, MetagenomicsDB makes accessing the data straightforward, especially during a collaborative project. In our study, clinical, *in vitro*, and *in silico* analyses produced three distinct file types (Sections 2.1 and 2.2). We needed to ensure the integrity of the data of each file type and maintain the correct connections between records in the different files. Using a more conventional approach with plain files, this would have been a tedious and error-prone manual task. Instead, we automated plausibility checks in MetagenomicsDB using the general import unit (Fig. 1). By comparing the read IDs, for instance, it verified that the FASTQ files and the respective classification files were actually matching. It then inserted the validated data into the storage component

Table 1
Comparison of MetagenomicsDB to other available software. Annotations are based on manual inspection of the resources, their official documentations (https://www.mg-rast.org/, https://www.ebi.ac.uk/metagenomics, https://mizuguchilab.org/manta, https://github.com/chenyian-nibio/manta, https://metaxplor.cirad.fr/metaXplor/, and https://github.com/SouthGreenPlatform/ metaXplor; accessed 2024/10/29), and publications [23,29,31,34,6].

Feature	MetagenomicsDB	MG-RAST	MGnify	MANTA	metaXplor
Analyses					
• Private analyses	+	+	-	+	+
• Support for custom classifier	+	-	-	+	+
• Support for custom classification databases	+	-	-	+	a
• Aware of dependencies between measurements	+	-	-	-	-
• Statistical analyses	External	Internal/ external	Internal/ external	Internal	External
Storage					
• Sequences	+	+	+	-	+
• Classifications	+	+	+	+	+
• Measurements	+	+	+	+	+
Export					
• Links to external resources	+	+	+	-	+
• Export in standard formats	+	+	+	-	+
Portable graphical interface	-	NA[b]	NA[b]	+	+

aLimited choice, as NCBI taxonomy ID or accession are required. [b]Public instance available from the internet.

PostgreSQL (Fig. 1). The automation of this quality control helped to reduce human error, as opposed to the manual approach. Since transactions in PostgreSQL are ACID-compliant (https://www.postgresql.org/about/; accessed 2024/03/20), storing data in a PostgreSQL database instead of plain files had additional quality assurance advantages. The acronym ACID stands for "atomicity, consistency, isolation, and durability" ([16], 290) and its principles comprise the following attributes with respect to changes of the stored data. Each transaction is carried out completely or not at all, thus avoiding partial updates (atomicity) [15]. In MetagenomicsDB, any synchronization between the database and the input data is conducted in a single transaction. Thus, even if just a single record is invalid, the synchronization can be completely undone, leaving the database in a consistent state. Changes to the data content must adhere to the constraints of the database management system (consistency) [15]. In the context of our system, this principle enables us to control the data types of the entered data, avoid duplicates over the lifetime of the study, and maintain the connections between our different data types. According to the ACID principles, each transaction is carried out in isolation, despite concurrent activity ([22], 136–38). The benefit for our collaborative project is that multiple investigators can work with the system at the same time to continuously update the study data without running the risk of overwriting each others changes. Once a transaction successfully modifies the data, the modifications are protected by the database management system from accidental loss (durability) [15].

We advocate the use of MetagenomicsDB as a backend for a graphical web interface. In order to highlight the potential of this approach, we have made our custom interface publicly accessible with a limited subset of the SGA study data (Section 3.4). It allows all investigators, regardless of bioinformatics expertise, to synchronize the study data with the database, explore, optionally filter the data using complex queries, and export them for statistical analyses, including read count normalization, in MicrobiomeAnalyst (Section 3.4). Thus, all team members receive live feedback on the progress of the study, which allows them to plan potential extensions, such as reanalyses or follow-up analyses, more efficiently. Furthermore, basic bioinformatic and statistical analyses of the preliminary study results are available to all team members. In order to further streamline the analysis workflow, we linked MetagenomicsDB to several public resources which we considered to be helpful for general microbiome analyses, as well as microbiome analyses in SGA children (Section 3.4). Wikipedia and LPSN provide a quick overview of the identified taxa. BV-BRC helps to assess the potential pathogenicity and thus the potential health impact of the microorganisms. Our customized PubMed search allows investigators to find information on already known links between the taxon of interest and SGA children more quickly. The web interface is not only expected to be useful for our internal purposes. By the time our study will be published, we aim to make the full study data available in MetagenomicsDB to reviewers and to the general public. Openly sharing the study results in an accessible way contributes to more transparent research and allows others to conduct follow-up analyses based on our findings. Nevertheless, in this scenario, data maintenance depends on the research group hosting the specific instance which is not ideal for long-term storage. Thus, the raw data should be additionally stored in public repositories to guarantee long-term access. For example, sequences and measurements could be published in MG-RAST or MGnify. In order to preserve the custom classifications, a public research data repository, such as Zenodo (https://zenodo.org; accessed 2024/12/05), could be used.

Despite the aforementioned advantages of a central data management system, if one was to develop such a system form scratch for every project, the investment in development time can be prohibitive. Thus, we designed MetagenomicsDB as a non- disposable data management system. Many components can be reused for future studies.

The following design decisions contribute to an increased portability: first, the use of a relational database with a flexible schema (Section 3.2) which allows, among other things, to assign any number of measurements to a sample or to classify sequences with multiple programs or databases. Second, we export essential login information for the database to the standardized ".pg_service.conf" configuration file (Supplementary Methods 5.3.2), which helps to migrate our scripts to different computational environments. Third, we use a modular concept supplemented with tests for the import and export unit of MetagenomicsDB. The tests verify the expected behavior of the modules (in new computational environments) and developers can use them later on to check the modules' integrity after custom adaptations. Thus, several general functions (e.g.: extracting files or parsing FASTQs and classification files) can be easily reused for future projects. At the time of writing, these functions are already flexible enough to support spreadsheet formats and classifications from the Kraken 2 software (Supplementary Table A6) which are not currently used in this project. Besides, the system is essentially agnostic to the sequencing instrument producing the reads and to the read length itself, as long as reads are independent (Fig. 2), basic formatting requirements of FASTQs are fulfilled, and (parts of) marker genes are targeted (Supplementary Methods 5.3.1). Currently, MicrobiomeAnalyst, supports reads from the 16S and 18S rRNA genes, as well as from the internal transcribed spacers (ITS) [18]. Certain high level-functions and features of the database schema necessarily are project-specific. Examples include the calculation of the weight-for-age z-scores and the assignment of patients to the weight-for-age z-score (sub-)categories. Our implementation of the web interface is also specific to our web environment. Nevertheless, due to the modular concept, project-specific functions can be replaced or adapted to new projects (Section 3) in a fraction of the time needed for developing a new data management system from scratch.

In conclusion, we have developed a data management system for the study of the gut microbiome in SGA children. We present our custom web interface as a visual example, highlighting the potential of MetagenomicsDB as the backend in a collaborative study effort. We have outlined, how this reduces human error and thus enhances quality assurance of the produced results, how *in silico* analyses in a collaborative project are streamlined, and how more transparent sharing of results at publication time can be achieved. We acknowledge that despite the efforts towards portability, our system is not yet ready to be widely applicable out-of-the-box for any microbiome study. The development of a portable web interface would be a useful improvement, but raised several concerns in terms of data security and selection of a web framework which provides users with a plug-and-play experience, despite largely different web environments. We intend to address this issue in a future version of our system. For the time being, we publish the backend code and encourage users to design their own custom web interfaces on top of our code. As experts of their data and their institution's best security practices, users are in a better position to create web frontends which are highly specialized to their individual use cases. Given that the essential core of the software is provided by us and most of the data needed for the web interface are already aggregated in the views of the database (Section 3.2), we expect that users can finish their adjustments in a reasonable amount of time following our visual example. Customization is supported by the modular concept and the provided tests. Besides, we extensively described the underlying rational of the system and provide example data (Supplementary Methods 5.3.3) and detailed installation instructions in our GitHub repository (https://github.com/IOB-Muenster/MetagenomicsDB; accessed 2024/12/18). Thus, we are reporting our endeavors to motivate the application of data management systems at the scale of single studies in microbiome research.

Ethics

The experimental protocol was established in accordance with the ethical guidelines of the Declaration of Helsinki and was approved by the Bioethics Committee of the Poznań University of Medical Sciences in Poznań, Poland (Resolution No. 248/20). Since the study includes

children, written informed consent was obtained from their parents or legal guardians before entering the study and after the aims and methodologies of the study were explained.

Funding

This study was financially supported by the MINIATURA 4 (2020/04/X/NZ7/00308) grant from the National Science Centre (NCN), Poland. The open access charge was funded by the Open Access Publication Fund of the University of Münster.

Author statement

All authors read and approved the final manuscript.

CRediT authorship contribution statement

Monika Englert-Golon: Resources, Investigation. **Joanna Ciomborowska-Basheer:** Methodology, Investigation. **Marta Szymankiewicz-Bręborowicz:** Resources, Investigation. **Anita Szwed:** Writing – review & editing, Resources, Project administration, Methodology, Investigation, Data curation, Conceptualization. **Izabela Makałowska:** Writing – review & editing, Supervision, Resources, Methodology, Conceptualization. **Wojciech Makałowski:** Writing – review & editing, Supervision, Resources, Conceptualization. **Magdalena Durda-Masny:** Writing – review & editing, Resources, Methodology, Investigation, Funding acquisition, Data curation, Conceptualization. **Felix Manske:** Writing – review & editing, Writing – original draft, Visualization, Methodology, Formal analysis, Data curation, Conceptualization. **Jan Mazela:** Resources, Investigation. **Norbert Grundmann:** Writing – review & editing, Visualization, Software, Data curation.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

We wish to thank Michelle Leyers for her help with the classification of sequences using the MetaG software and Marc-Nicolas Bendisch who contributed to the benchmarking of the MetaG results, which was the basis for the updated parameter training in MetaG. Maciej Makałowski designed the graphical abstract. Parts of the calculations for this publication were performed on the High Performance Computing (HPC) cluster Paralleles Linux-System für Münsteraner Anwender (PALMA) II of the University of Münster, subsidized by the German Research Foundation (INST 211/667–1).

Appendix A. Supporting information

Supplementary data associated with this article can be found in the online version at doi:10.1016/j.csbj.2024.12.031.

References

[1] Aries Vivienne, Crowther JS, Drasar BS, Hill MJ, Williams REO. Bacteria and the aetiology of cancer of the large bowel. Gut 1969;10(5):334–5. https://doi.org/10.1136/gut.10.5.334.

[2] Bäckhed Fredrik, Roswall Josefine, Peng Yangqing, Feng Qiang, Jia Huijue, Kovatcheva-Datchary Petia, Yin Li, et al. Dynamics and stabilization of the human gut microbiome during the first year of life. Cell Host Microbe 2015;17(5):690–703. https://doi.org/10.1016/j.chom.2015.04.004.

[3] Barker DJP, Hales CN, Fall CHD, Osmond C, Phipps K, Clark PMS. Type 2 (non-insulin-dependent) diabetes mellitus, hypertension and hyperlipidaemia (Syndrome X): relation to reduced fetal growth. Diabetologia 1993;36(1):62–7. https://doi.org/10.1007/BF00399095.

[4] Beer Rachael J, Cnattingius Sven, Susser Ezra S, Villamor Eduardo. Associations of preterm birth, small-for-gestational age, preeclampsia and placental abruption with attention-deficit/hyperactivity disorder in the offspring: nationwide cohort and sibling-controlled studies. Acta Paediatr 2022;111(8):1546–55. https://doi.org/10.1111/apa.16375.

[5] Bolyen Evan, Rideout Jai Ram, Dillon Matthew R, Bokulich Nicholas A, Abnet Christian C, Al-Ghalith Gabriel A, Alexander Harriet, et al. Reproducible, interactive, scalable and extensible microbiome data science using QIIME 2. Nat Biotechnol 2019;37(8):852–7. https://doi.org/10.1038/s41587-019-0209-9.

[6] Chen Yi-An, Park Jonguk, Natsume-Kitatani Yayoi, Kawashima Hitoshi, Mohsen Attayeb, Hosomi Koji, Tanisawa Kumpei, et al. MANTA, an integrative database and analysis platform that relates microbiome and phenotypic data. PLoS One 2020;15(12):e0243609. https://doi.org/10.1371/journal.pone.0243609.

[7] Codd EF. Is Your DBMS Really Relational? Computerworld 1985. October 14, 1985.

[8] Cole James R, Wang Qiong, Fish Jordan A, Chai Benli, McGarrell Donna M, Sun Yanni, Brown CTitus, Porras-Alfaro Andrea, Kuske Cheryl R, Tiedje James M. Ribosomal database project: data and tools for high throughput rRNA analysis. Nucleic Acids Res 2014;42(D1):D633–42. https://doi.org/10.1093/nar/gkt1244.

[9] David Lawrence A, Maurice Corinne F, Carmody Rachel N, Gootenberg David B, Button Julie E, Wolfe Benjamin E, Ling Alisha V, et al. Diet rapidly and reproducibly alters the human gut microbiome. Nature 2014;505:559–63. https://doi.org/10.1038/nature12820.

[10] Dominguez-Bello Maria G, Costello Elizabeth K, Contreras Monica, Magris Magda, Hidalgo Glida, Fierer Noah, Knight Rob. Delivery mode shapes the acquisition and structure of the initial microbiota across multiple body habitats in newborns. Proc Natl Acad Sci USA 2010;107(26):11971–5. https://doi.org/10.1073/pnas.1002601107.

[11] Eckburg Paul B, Bik Elisabeth M, Bernstein Charles N, Purdom Elizabeth, Dethlefsen Les, Sargent Michael, Gill Steven R, Nelson Karen E, Relman David A. Diversity of the human intestinal microbial flora. Science 2005;308(5728):1635–8. https://doi.org/10.1126/science.1110591.

[12] Escobar-Zepeda Alejandra, Vera-Ponce de León Arturo, Sanchez-Flores Alejandro. The road to metagenomics: from microbiology to DNA sequencing technologies and bioinformatics. Front Genet 2015;6:348. https://doi.org/10.3389/fgene.2015.00348.

[13] Finegold Sydney M, Attebery Howard R, Sutter Vera L. Effect of diet on human fecal flora: comparison of Japanese and American diets. Am J Clin Nutr 1974;27(12):1456–69. https://doi.org/10.1093/ajcn/27.12.1456.

[14] Fujimura Kei E, Sitarik Alexandra R, Havstad Suzanne, Lin Din L, Levan Sophia, Fadrosh Douglas, Panzer Ariane R, et al. Neonatal gut microbiota associates with childhood multisensitized atopy and T cell differentiation. Nat Med 2016;22(10):1187–91. https://doi.org/10.1038/nm.4176.

[15] Gray Jim 1981. The Transaction Concept: Virtues and Limitations (Invited Paper). In Very Large Data Bases, 7th International Conference, September 9-11, 1981, Cannes, France, Proceedings, VLDB '81:144–154. VLDB. IEEE Computer Society.

[16] Haerder Theo, Reuter Andreas. Principles of transaction-oriented database recovery. ACM Comput Surv 1983;15(4):287–317. https://doi.org/10.1145/289.291.

[17] Lagier J-C, Armougom F, Million M, Hugon P, Pagnier I, Robert C, Bittar F, et al. Microbial culturomics: paradigm shift in the human gut microbiome study. Clin Microbiol Infect 2012;18(12):1185–93. https://doi.org/10.1111/1469-0691.12023.

[18] Lu Yao, Zhou Guangyan, Ewald Jessica, Pang Zhiqiang, Shiri Tanisha, Xia Jianguo. Microbiome analyst 2.0: comprehensive statistical, functional and integrative analysis of microbiome data. Nucleic Acids Res 2023;51(W1):W310–8. https://doi.org/10.1093/nar/gkad407.

[19] Manske Felix, Grundmann Norbert, Makałowski Wojciech. MetaGenomic analysis of short and long reads. bioRxiv 2020. https://doi.org/10.1101/2020.03.13.991190.

[20] Manter Daniel K, Korsa Matthew, Tebbe Caleb, Delgado Jorge A. myPhyloDB: a local web server for the storage and analysis of metagenomic data. Database 2016;2016:baw037. https://doi.org/10.1093/database/baw037.

[21] Matson Vyara, Fessler Jessica, Bao Riyue, Chongsuwat Tara, Zha Yuanyuan, Alegre Maria-Luisa, Luke Jason J, Gajewski Thomas F. The commensal microbiome is associated with anti–PD-1 efficacy in metastatic melanoma patients. Science 2018;359(6371):104–8. https://doi.org/10.1126/science.aao3290.

[22] Meier, Andreas, and Michael Kaufmann. 2016. SQL- & NoSQL-Datenbanken. 8th ed. eXamen.press. Berlin, Heidelberg: Springer Vieweg. https://doi.org/10.1007/978-3- 662-47664-2.

[23] Meyer F, Paarmann D, D'Souza M, Olson R, Glass EM, Kubal M, Paczian T, et al. The metagenomics RAST server – a public resource for the automatic phylogenetic and functional analysis of metagenomes. BMC Bioinforma 2008;9:386. https://doi.org/10.1186/1471-2105-9-386.

[24] Moore WEC, Holdeman Lillian V. Human fecal flora: the normal flora of 20 Japanese-Hawaiians. Appl Microbiol 1974;27(5):961–79. https://doi.org/10.1128/am.27.5.961-979.1974.

[25] O'Keeffe Michael J, O'Callaghan Michael, Williams Gail M, Najman Jake M, Bor William. Learning, cognitive, and attentional problems in adolescents born small for gestational age. Pediatrics 2003;112(2):301–7. https://doi.org/10.1542/peds.112.2.301.

[26] Olson Robert D, Assaf Rida, Brettin Thomas, Conrad Neal, Cucinell Clark, Davis James J, Dempsey Donald M, et al. Introducing the bacterial and viral bioinformatics resource center (BV-BRC): a resource combining PATRIC, IRD and ViPR. Nucleic Acids Res 2023;51(D1):D678–89. https://doi.org/10.1093/nar/gkac1003.

[27] Parte Aidan C, Sardà Carbasse Joaquim, Meier-Kolthoff Jan P, Reimer Lorenz C, Göker Markus. List of prokaryotic names with standing in nomenclature (LPSN) moves to the DSMZ. Int J Syst Evolut Microbiol 2020;70(11):5607–12. https://doi.org/10.1099/ijsem.0.004332.

[28] Knop Ravn, Marianne, Geng Ting-Ting, Gorny Alexander Wilhelm, Ding Renyu, Li Changwei, Ley Sylvia H, Huang Tao. Birth weight and risk of type 2 diabetes mellitus, cardiovascular disease, and hypertension in adults: a meta-analysis of 7 646 267 participants from 135 studies. J Am Heart Assoc 2018;7(23):e008870. https://doi.org/10.1161/JAHA.118.008870.

[29] Richardson Lorna, Allen Ben, Baldi Germana, Beracochea Martin, Bileschi Maxwell L, Burdett Tony, Burgin Josephine, et al. MGnify: the microbiome sequence data analysis resource in 2023. Nucleic Acids Res 2023;51(D1):D753–9. https://doi.org/10.1093/nar/gkac1080.

[30] Sayers Eric W, Beck Jeff, Bolton Evan E, Brister JRodney, Chan Jessica, Comeau Donald C, Connor Ryan, et al. Database resources of the national center for biotechnology information. Nucleic Acids Res 2024;52(D1):D33–43. https://doi.org/10.1093/nar/gkad1044.

[31] Sempéré Guilhem, Pétel Adrien, Abbé Magsen, Lefeuvre Pierre, Roumagnac Philippe, Mahé Frédéric, Baurens Gaël, Filloux Denis. metaXplor: an interactive viral and microbial metagenomic data manager. GigaScience 2021;10(2):giab001. https://doi.org/10.1093/gigascience/giab001.

[32] Shen Wei, Ren Hong. TaxonKit: a practical and efficient NCBI taxonomy toolkit. J Genet Genom, Spec Issue Micro 2021;48(9):844–50. https://doi.org/10.1016/j.jgg.2021.03.006.

[33] WHO Multicentre Growth Reference Study Group. WHO Child Growth Standards: Length/Height-for-Age, Weight-for-Age, Weight-for-Length, Weight-for-Height and Body Mass Index-for-Age: Methods and Development. Geneva: World Health Organization; 2006. ⟨https://www.who.int/publications/i/item/924154693X⟩.

[34] Wilke Andreas, Bischof Jared, Gerlach Wolfgang, Glass Elizabeth, Harrison Travis, Keegan Kevin P, Paczian Tobias, et al. The MG-RAST metagenomics database and portal in 2015. Nucleic Acids Res 2016;44(D1):D590–4. https://doi.org/10.1093/nar/gkv1322.

[35] Wood Derrick E, Lu Jennifer, Langmead Ben. Improved metagenomic analysis with kraken 2. Genome Biol 2019;20(257):019–1891-0. https://doi.org/10.1186/s13059-.

[36] World Health Organization. International Classification of Diseases, Eleventh Revision (ICD-11). Geneva: World Health Organization; 2022.